

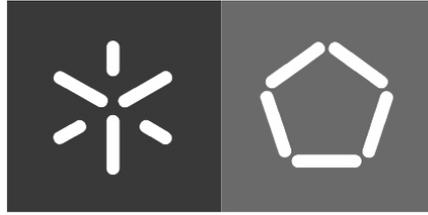
Universidade do Minho

Escola de Engenharia

Pedro Miguel Pereira Borges

Configuração do RUP com Vista à
Simplificação dos Elencos Processuais em
PMEs de Desenvolvimento de Software

Novembro de 2007



Universidade do Minho

Escola de Engenharia

Pedro Miguel Pereira Borges

**Configuração do RUP com Vista à
Simplificação dos Elencos Processuais em
PMEs de Desenvolvimento de Software**

Tese de Mestrado em Sistemas de Informação

Trabalho efectuado sob a orientação do

Professor Doutor Ricardo J. Machado

Departamento de Sistemas de Informação

Novembro de 2007

DECLARAÇÃO

Nome: PEDRO MIGUEL PEREIRA BORGES

Endereço Electrónico: borges.pedro@gmail.com

Telefone: 223 715 890

Nº. do Bilhete de Identidade: 10812319

Título da Dissertação de Mestrado:

Configuração do RUP com Vista à Simplificação dos Elencos Processuais em PMEs de Desenvolvimento de Software

Orientador: Professor Doutor Ricardo J. Machado

Data de conclusão: Novembro de 2007

Designação do Mestrado: Mestrado em Sistemas de Informação

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO, APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, ____ / ____ / _____

Assinatura: _____

Para a Elsa e para a nossa Esperança de um mundo melhor...

Abstract

In the last decades we have been witnessing a significant increase in the complexity inherent to software development projects, due not only to a higher degree of sophistication in the contexts they aim to serve, but also to the natural evolution of the features implemented by the available software systems and applications. On the other hand, the international competition lead by the so-called software factories, frequently based on countries with low labour costs (India, China, Pakistan, etc.), increasingly threatens the portuguese software producers. As their usually high CMM (*Capability Maturity Model*) rankings hint, these companies are well organized and employ knowledgeable human resources.

However, the reduced dimension (micro, small or medium) of the majority of the portuguese corporations operating in this activity sector translates, frequently, in a significant constraint to the group of individuals that might be involved in each project, with obvious consequences to their efficiency and effectiveness. Further more, this situations is aggravated by the absence of known configurations for software development processes (namely for the RUP - *Rational Unified Process*) that lead to a set of roles sufficiently self-contained to adequately address the needs of this type of organizations.

Therefore, this dissertation demonstrates how to accomplish a configuration of the RUP in order to obtain two separate process casts that, without neglecting any critical role of the software development process, may easily be adopted by a small or medium corporation during the project execution period. Additionally, an effort was made to aid the actions of the individuals in charge of each role, by creating job descriptions detailing their responsibilities.

Resumo

Nas últimas décadas tem vindo a assistir-se a um considerável aumento da complexidade inerente aos projectos de desenvolvimento de software, fruto não só de um crescente grau de sofisticação dos contextos que pretendem servir, como também da evolução natural das funcionalidades proporcionadas pelas aplicações e sistemas de base que se encontram disponíveis. Por outro lado, os produtores nacionais de software encontram-se cada vez mais ameaçados pela concorrência internacional, nomeadamente a protagonizada pelas denominadas *fábricas de software*, tipicamente localizadas em países com baixos custos salariais (Índia, China, Paquistão, etc.) e que, como deixam transparecer os elevados níveis na escala CMM (*Capability Maturity Model*) que usualmente apresentam, se encontram bem organizadas e dispõem de quadros de reconhecida competência.

Contudo, a reduzida dimensão (micro, pequena ou média) da maioria das empresas portuguesas a operar neste sector de actividade traduz-se, frequentemente, numa significativa limitação do leque de indivíduos que pode envolver em cada projecto, com consequências óbvias para a eficácia e eficiência do seu funcionamento. Esta situação é, ainda, agravada pelo facto de não serem conhecidas configurações de processos de desenvolvimento de software (nomeadamente do RUP – *Rational Unified Process*) que resultem num conjunto de papéis suficientemente auto-contido para responder adequadamente às necessidades deste tipo de organizações.

Pelo exposto, esta dissertação demonstra como efectuar uma configuração do RUP no sentido de obter dois elencos processuais que, sem negligenciar nenhuma função crítica do processo de desenvolvimento de software, podem ser facilmente adoptados no contexto específico de uma PME durante o período de execução de um projecto. Adicionalmente, procurou-se ainda auxiliar o desempenho de cada um dos papéis pertencentes aos referidos elencos processuais, disponibilizando fichas individuais com a caracterização detalhada das suas responsabilidades.

Agradecimentos

À Elsa, por toda a compreensão e energia positiva que me transmite...

Aos meus pais e restante família, pelo constante incentivo...

Ao meu orientador, Professor Doutor Ricardo J. Machado, pelo empenho que colocou nesse papel e pela inextinguível disponibilidade demonstrada...

À MULTICERT, na pessoa do seu Director-Geral Dr. José Pina Miranda, por ter proporcionado as condições necessárias à aplicação de algumas das ideias que este documento encerra...

Aos meus colegas Fernando Ribeiro, Pina Miranda, Renato Portela, Ricardo Barroso e Tânia Barros, pelos pertinentes comentários e contributos...

Ao Hugo Sousa, por me fazer acreditar que era possível concluir esta dissertação...

À Sandra Ribeiro e ao Gonçalo Hermenegildo por se terem voluntariado como revisores deste documento...

Índice

Abstract	iii
Resumo	iv
Agradecimentos	v
1. Introdução	1
1.1. DESENVOLVIMENTO DE SOFTWARE EM PORTUGAL	1
1.2. DESVANTAGENS COMPETITIVAS DAS PMES	6
1.3. OBJECTIVOS	8
1.4. ESTRUTURAÇÃO DO DOCUMENTO.....	9
2. O <i>Rational Unified Process</i> e suas Aplicações.....	11
2.1. INTRODUÇÃO	11
2.2. O <i>RATIONAL UNIFIED PROCESS</i>	13
2.3. INTERVENIENTES NO RUP	25
2.4. APLICAÇÕES DO RUP	38
2.5. CONCLUSÕES	59
3. Configuração do RUP para Utilização em PMEs.....	61
3.1. INTRODUÇÃO	61
3.2. SIMPLIFICAÇÃO DO ELENCO PROCESSUAL.....	68
3.2.1. <i>Modelo Base</i>	70
3.2.2. <i>Modelo Reduzido</i>	91
3.3. ACUMULAÇÃO DE RESPONSABILIDADES.....	100
3.4. ENQUADRAMENTO ORGANIZACIONAL DO ELENCO PROCESSUAL PROPOSTO.....	104
3.5. CONCLUSÕES	106
4. Caracterização do Elenco Processual Proposto	107
4.1. INTRODUÇÃO	107
4.2. CARACTERIZAÇÃO DO ELENCO PROCESSUAL PROPOSTO.....	112
4.2.1. [ABI] <i>Administrador de Bases de Dados</i>	113
4.2.2. [ANI] <i>Analista</i>	115
4.2.3. [AQI] <i>Auditor de Qualidade</i>	117
4.2.4. [ASI] <i>Arquitecto de Software</i>	119
4.2.5. [CCI] <i>Coordenador de Comunicação e Imagem</i>	122
4.2.6. [CDI] <i>Coordenador de Desenvolvimento</i>	124
4.2.7. [CEI] <i>Chefe de Equipa</i>	128
4.2.8. [CII] <i>Coordenador de Infra-Estrutura</i>	135
4.2.9. [CQI] <i>Coordenador de Qualidade</i>	139
4.2.10. [CSI] <i>Coordenador de Suporte</i>	142
4.2.11. [GPI] <i>Gestor de Projecto</i>	145
4.2.12. [PPI] <i>Patrocinador de Projecto</i>	152
4.2.13. [PRI] <i>Programador</i>	154
4.3. CONCLUSÕES	156
5. Conclusões	157
5.1. EXPERIMENTAÇÃO E RESULTADOS OBTIDOS	157
5.2. TRABALHO FUTURO.....	160

Bibliografia	162
Anexo I Elenco Processual do RUP.....	165
Anexo II Diagramas de Apoio ao Processo	170

Lista de Figuras

Figura 1.1 – Número de Empresas de Serviços e Software (por distrito) segundo [Beira, et al., 2006]	4
Figura 1.2 – Número de Trabalhadores das Empresas de Serviços e Software (por distrito) segundo [Beira, et al., 2006]	5
Figura 1.3 – Volume de Vendas (em milhares de euros) das Empresas de Serviços e Software (por distrito) segundo [Beira, et al., 2006]	5
Figura 1.4 – Volume de Vendas (em milhares de euros) por Empresa de Serviços e Software (por distrito) segundo [Beira, et al., 2006]	6
Figura 1.5 – Número de Trabalhadores por Empresa de Serviços e Software (por distrito) segundo [Beira, et al., 2006]	7
Figura 2.1 – 4Ps do Desenvolvimento de Software	13
Figura 2.2 – Arquitectura Processual do RUP	17
Figura 2.3 - Participação de alguns Papéis nos Fluxos do RUP	26
Figura 2.4 – Modelo Genérico para Organizações Orientadas aos Processos	46
Figura 2.5 – Modelo Genérico para Organizações Produtoras de Software e Orientadas aos Processos	47
Figura 2.6 – Diagrama de Actividades do Sub-Processo de Business Modelling	51
Figura 2.7 – Meta-modelo do OPEN	57
Figura 2.8 – Meta-modelo do RUP	58
Figura 3.1 – Mapeamentos do Modelo Base	88
Figura 3.2 – Mapeamentos do Modelo Reduzido	98
Figura 3.3 – Mapa Comparativo entre o Modelo Base e o Modelo Reduzido	99
Figura 4.1 – Fluxos de Comunicação (Internos)	108
Figura 4.2 – Fluxos de Comunicação (Internos e Externos)	110

Lista de Tabelas

Tabela 2.1 – Papéis definidos em [Rational, 2001]	33
Tabela 2.2 – Diferenças de Designação de Papéis entre [Jacobson, et al., 1999] e [Rational, 2001]	35
Tabela 2.3 – Papéis definidos em [Kruchten, 2004] mas ausentes em [Rational, 2001].....	37
Tabela 2.4 – Omissões do Apêndice A de [Kruchten, 2004].....	37
Tabela 2.5 – Lista de Artefactos utilizados pela Zühlke Engineering AG	42
Tabela 2.6 – Utilização de Artefactos de Business Modelling	48
Tabela 3.1 – Denominações do Elenco Processual Proposto	90
Tabela 3.2 – Restrições à Acumulação de Papéis (no mesmo projecto)	101
Tabela 3.3 – Quadro-Resumo de Restrições à Acumulação de Papéis (no mesmo projecto)	102
Tabela 3.4 – Restrições à Acumulação de Papéis (entre projectos)	103
Tabela 3.5 – Quadro-Resumo de Restrições à Acumulação de Papéis (entre projectos)	104
Tabela 3.6 – Enquadramento Organizacional do Elenco Proposto.....	105

Glossário

ABI	Administrador de Bases de Dados (Interno)
ANI	Analista (Interno)
AQI	Auditor de Qualidade (Interno)
ASI	Arquitecto de Software (Interno)
BRUF	<i>Big Requirements Up Front</i>
CAE	Classificação das Actividades Económicas
CCI	Coordenador de Comunicação e Imagem (Interno)
CDI	Coordenador de Desenvolvimento (Interno)
CEI	Chefe de Equipa (Interno)
CIE	Coordenador de Infra-Estrutura (Externo)
CII	Coordenador de Infra-Estrutura (Interno)
CMM	<i>Capability Maturity Model</i>
CQE	Coordenador de Qualidade (Externo)
CQI	Coordenador de Qualidade (Interno)
CSE	Coordenador de Suporte (Externo)
CSI	Coordenador de Suporte (Interno)
DSI	Departamento de Sistemas de Informação
EOS	<i>Evolutionary, Object-oriented Software Development</i>
GPE	Gestor de Projecto (Externo)
GPI	Gestor de Projecto (Interno)
GUI	<i>Graphical User Interface</i>
IDE	<i>Integrated Development Environment</i>
HTML	<i>HyperText Markup Language</i>
KA	<i>Knowledge Area</i>
OPEN	<i>Object-Oriented Processes, Environment and Notation</i>
PME	Micro, Pequena ou Média Empresa
PPE	Patrocinador de Projecto (Externo)
PPI	Patrocinador de Projecto (Interno)
PRI	Programador (Interno)
UM	Universidade do Minho

UML	<i>Unified Modelling Language</i>
RUP	<i>Rational Unified Process</i>
RUP SE	<i>Rational Unified Process for Systems Engineering</i>
SLA	<i>Service Level Agreement</i>
SPEM	<i>Software Process Engineering Metamodel</i>
SWEBOK	<i>Software Engineering Body of Knowledge</i>
TIC	Tecnologias da Informação e Comunicação

1. Introdução

O capítulo que dá início a esta dissertação começa por apresentar o panorama nacional de desenvolvimento de software, proporcionando alguns dados relevantes para uma melhor compreensão do tecido empresarial que o caracteriza. De seguida, e após analisar as especificidades do contexto PME, é definido o contributo que a tese pretende dar, enunciando os objectivos que se propõe atingir. Por último, procede-se à descrição da estrutura do documento e dos restantes capítulos que o compõem.

1.1. Desenvolvimento de Software em Portugal

Com a sua participação na conferência NATO *Science Committee* de 1969, Fritz Bauer deu origem a uma nova área científica, ao apresentar a primeira definição conhecida de **Engenharia de Software**, caracterizada como “... *a utilização dos princípios básicos da engenharia para obter, de uma forma economicamente viável, software fiável e que corra eficientemente em computadores reais*” [Naur, Randell, 1969]. Como consequência, assistiu-se, a partir desse momento, ao proliferar da investigação científica nessa área de conhecimento, contribuindo para solidificar gradualmente a sua importância e preponderância no meio académico. Cerca de meados da década de 1970, esse crescendo traduziu-se inevitavelmente no surgimento em Portugal¹, e um pouco por todo o globo, da oferta de ensino universitário específico nesse domínio, até então ávida por recursos humanos com formação adequada. Nos anos que se seguiram, as exigências do mercado fizeram com que a procura de profissionais por esse sector de actividade crescesse a um ritmo muito superior à

¹ Em concreto, as pioneiras Licenciatura em Eng^a da Produção (ramo Informática e Sistemas) da Universidade do Minho e Licenciatura Terminal em Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.

capacidade instalada nas universidades para os formar atempadamente. Consequentemente, a partir do final dos anos noventa foi possível assistir a uma competição acérrima entre as organizações para assegurar a contratação do maior número possível de recursos qualificados e, dessa forma, aumentar a sua capacidade de resposta às oportunidades de um mercado (à data) em franca expansão. Convém, contudo, realçar que esse estado de necessidade das organizações lhes impunha um nível de exigência relativamente baixo² nos critérios de recrutamento a aplicar, principalmente quando comparado com o panorama actual³. Assim, no início do século XXI o efeito conjugado de todos estes acontecimentos colocava a Engenharia de Software na ribalta, criando uma aura à volta dos seus profissionais e esperando que os seus desempenhos se transformassem na (por vezes única) base de competitividade das organizações.

Contudo, e focando agora no mercado nacional do desenvolvimento de software, a recessão económica que se fez sentir a partir de 2001 causou uma redução significativa na procura de projectos na área, aumentando o nível de competitividade entre concorrentes e fortalecendo a dependência dos produtores de software para com a administração pública (em virtude de uma redução menos abrupta da procura do sector público). Porém, o decréscimo na capacidade de investimento do sector privado não foi acompanhado por igual diminuição dos seus níveis de exigência, já que procuravam rentabilizar, ao máximo, o valor que podiam dispendir. Como consequência, a implementação de projectos de desenvolvimento de software tornou-se mais complexa, colocando as equipas sob o efeito de mais e maiores pressões, que se viam na necessidade de obter mais resultados com menos recursos. Contudo, no seio dos profissionais que viveram essa época era frequente encontrar quem sentisse alguma falta de preparação perante os desafios com que se confrontava no dia-a-dia, fruto de uma “orfandade” prática (e por vezes até teórica) de alguma oferta universitária em domínios cada vez mais relevantes. Na verdade, apesar de a maior parte dos licenciados deixar as instituições de ensino com elevada formação técnica, era usual verificar a existência de omissões e/ou insuficiências nas suas competências em áreas igualmente importantes, como sejam a gestão de projectos, o trabalho em equipa, as ciências empresariais e as competências sociais. Porém, se anteriormente isso era suficiente e havia oportunidade para dirimir essas

² Sendo usualmente suficiente ter formação, mais ou menos extensa, em engenharia de software e/ou ciências da computação, mesmo que com reduzida (ou até inexistente) experiência profissional, em virtude da “juventude” desta área de conhecimento.

³ Onde, para além da formação base, é consideravelmente valorizada a experiência profissional, bem como outras competências adquiridas através de acções de formação complementares, eventualmente consubstanciadas na obtenção de outros graus académicos ou certificações profissionais (*Sun, Microsoft, Cisco, Oracle, PMBOK*, etc).

lacunas, passaram a viver-se tempos em que estas se transformavam imediatamente em desvantagens competitivas, principalmente perante a recente concorrência internacional protagonizada pelas denominadas *fábricas de software*, bem organizadas e localizadas em países com baixos custos salariais (Índia, China, Paquistão, etc.), como deixam transparecer os elevados níveis na escala CMM (*Capability Maturity Model*) [Paulk, *et al.*, 1995] que usualmente apresentam. Todavia, a adopção do modelo de Bolonha tem originado uma revisão e reformulação da oferta universitária, conduzindo a uma maior focalização dos conteúdos nas competências centrais⁴ de cada curso e a um aumento na diversidade das especializações, o que poderá potenciar a mobilidade⁵ dos alunos e propiciar uma melhor resposta às necessidades do mercado.

No que diz respeito aos adjudicantes de projectos de desenvolvimento de software em Portugal, e não considerando os clássicos⁶ obstáculos ao processo que se verificam um pouco por toda a parte, deve notar-se a sua (ainda) reduzida familiarização com os denominados *métodos ágeis* [Larman, 2003], resultando tipicamente na sua relutância em aceitar a sua adopção e preferindo abordagens mais convencionais (por exemplo, do tipo “*Cascata*”). Porém, e apesar de usualmente se exercer sobre os adjudicatários uma enorme pressão para que apresentem resultados rapidamente, é frequente verificar que o mesmo rigor não é aplicado reflexivamente⁷ pelas entidades adjudicantes, eventualmente devido a factores culturais que as levam a encarar os adjudicatários como meros prestadores de serviço e não como parceiros que compõem a sua cadeia de valor e contribuem decisivamente para a prossecução dos seus objectivos.

Por outro lado, uma análise (ainda que breve) sobre o panorama nacional nesta área não poderia estar completa sem examinar as particularidades mais relevantes dos produtores de software que caracterizam o contexto português sobre o qual, infelizmente, se desconhecem estudos abrangentes. Contudo, e à falta de uma fonte de informação mais precisa e actualizada, em [Beira, *et al.*, 2006] podem encontrar-se dados (relativos ao ano de 2003) sobre as empresas nacionais que desenvolvem a sua actividade no sector das TIC

⁴ Por exemplo, através da redução do tradicionalmente extenso conjunto de cadeiras comuns aos cursos de Engenharia (vulgarmente apelidado de *tronco comum das Engenharias*).

⁵ Expondo-os a diferentes perspectivas e métodos de ensino.

⁶ Dificuldade das organizações cliente em exteriorizar os requisitos, alterações de requisitos durante a(s) fase(s) de implementação, etc.

⁷ Por exemplo, ao não avaliar na altura devida os artefactos que lhe são entregues, ao não se pronunciar atempadamente sobre assuntos de importância para o desenrolar do projecto, etc.

(Tecnologias da Informação e Comunicação) que, entre outras, inclui as que se dedicam aos serviços e software⁸. Começando por abordar a perspectiva geográfica, é possível constatar através da Figura 1.1 que 45,5% das empresas de serviços e software se encontram sediadas no distrito de Lisboa, o que de uma certa forma vem confirmar a noção empírica detida pelos profissionais da área da enorme concentração de oferta existente na capital portuguesa.

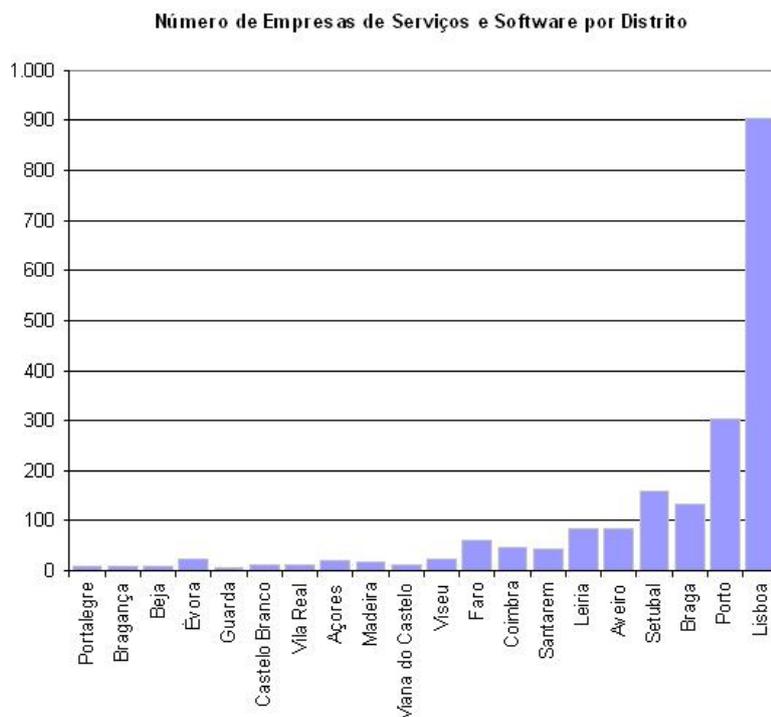


Figura 1.1 – Número de Empresas de Serviços e Software (por distrito) segundo [Beira, et al., 2006]

Porém, se a densidade de empresas de serviços e software no distrito de Lisboa é grande, a Figura 1.2 permite verificar que a convergência de recursos humanos especializados para essa zona é ainda maior, ao representar 69,4% do valor nacional. Note-se que este valor é ainda acrescido de um considerável (e dificilmente mensurável) número de indivíduos que, apesar de formalmente vinculados a organizações sediadas em outras zonas do país (Porto, Braga, Coimbra, etc.), exercem usualmente a sua actividade na zona da Grande Lisboa, no âmbito da prestação de serviços de consultoria e/ou de subcontratação. Por outro lado, e voltando à questão da formação universitária, é fácil prognosticar a enorme disparidade existente entre a procura e oferta de licenciados em Informática nesta região, contribuindo assim para a existência de um fenómeno migratório que encaminha para lá um considerável número de profissionais oriundos de outros locais.

⁸ Que correspondem aos CAE (Classificação das Actividades Económicas) 72100, 72200, 72300, 72400, 72500 e 72600.

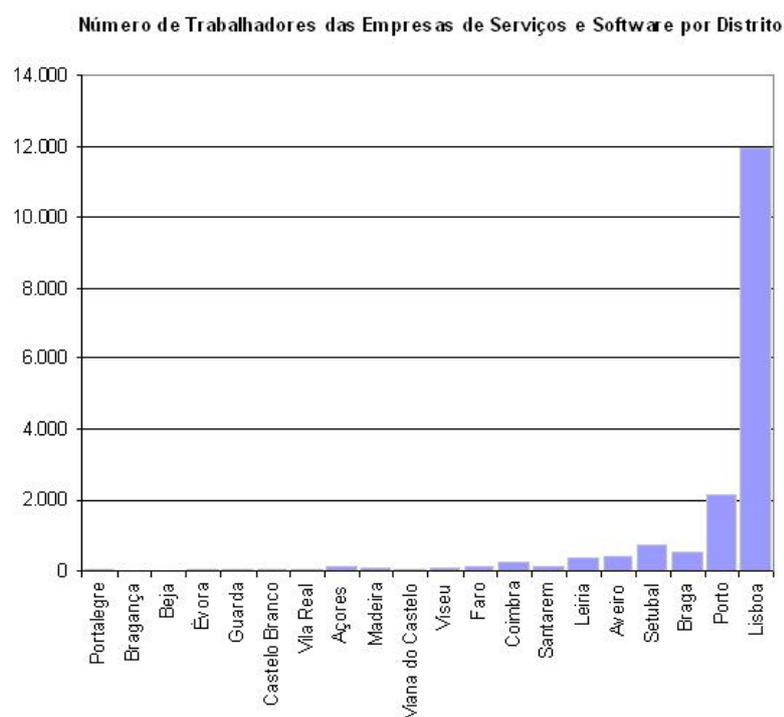


Figura 1.2 – Número de Trabalhadores das Empresas de Serviços e Software (por distrito) segundo [Beira, et al., 2006]

No que concerne ao volume de negócios (ver Figura 1.3), o distrito de Lisboa representa 32,5% do total nacional, enquanto o distrito do Porto atinge os 62,5%.

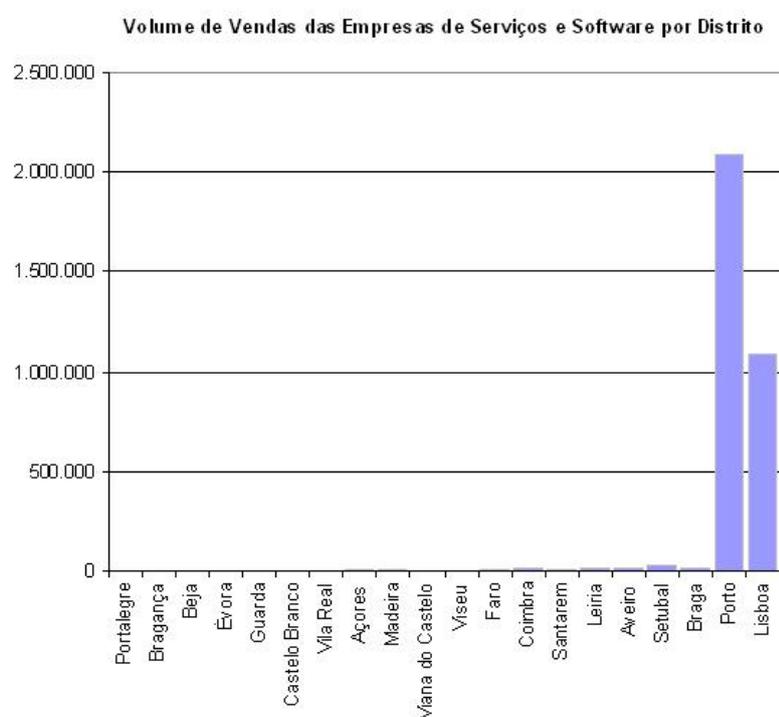


Figura 1.3 – Volume de Vendas (em milhares de euros) das Empresas de Serviços e Software (por distrito) segundo [Beira, et al., 2006]

Curiosamente, as empresas de serviços e software do distrito do Porto apresentam um volume de vendas acumulado 92% superior ao do distrito de Lisboa, não obstante disporem de apenas 18% dos recursos humanos das suas congéneres da capital.

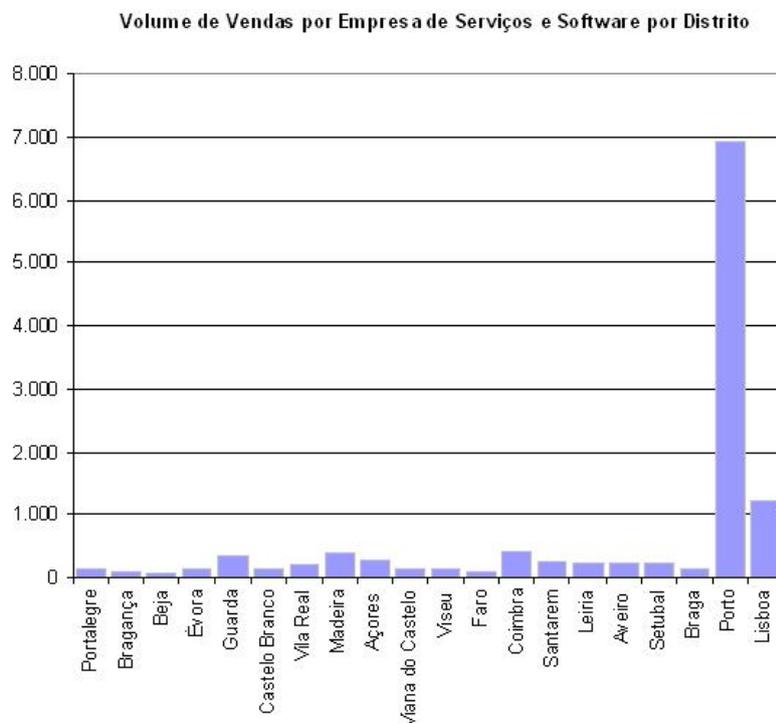


Figura 1.4 – Volume de Vendas (em milhares de euros) por Empresa de Serviços e Software (por distrito) segundo [Beira, et al., 2006]

Consequentemente, e como a Figura 1.4 denota claramente, cada empresa de serviços e software do distrito do Porto factura (em média) mais 474% do que uma empresa similar sediada no distrito de Lisboa o que, mesmo que fruto do contributo de eventuais factores extraordinários⁹, não deixa de ser algo surpreendente. Um outro interessante prisma de análise é o factor escala, a explicar na secção que se segue.

1.2. Desvantagens Competitivas das PMEs

De acordo com [IAPMEI, 2007], uma organização é considerada uma PME¹⁰ (Micro, Pequena ou Média Empresa) quando possui menos de 250 colaboradores e apresenta um volume de negócios igual ou inferior a 50 milhões de euros (ou, em alternativa, um balanço igual ou inferior a 43 milhões de euros). Ainda de acordo com a mesma fonte, em 2004

⁹ Como, por exemplo, a concretização simultânea de várias oportunidades de negócio de grande dimensão.

¹⁰ De acordo com a Recomendação nº 2003/361/CE de 6 de Maio de 2003.

99,6% de todas as empresas nacionais eram PME, empregando 75,1% da população activa e representando 56,8% do volume de negócios agregado. Em termos unitários, os indicadores disponíveis apontavam para a existência (em média) de maiores volumes de negócios nas empresas de serviços e software (1.685,59 milhares de euros, segundo [Beira, *et al.*, 2006]) do que na generalidade das empresas PME (558,1 milhares de euros). Por outro lado, e no mesmo período, cada PME nacional apresentava uma dimensão média de 7,1 trabalhadores, ligeiramente inferior quando comparada com a dimensão média das empresas de serviços e software (em 2003) que atingiam os 8,69 trabalhadores. Porém, se analisarmos a dimensão média deste tipo de empresas em termos geográficos (ver Figura 1.5), podemos ainda verificar que apenas no distrito de Lisboa esta é superior à média nacional (13,3 trabalhadores por empresa).

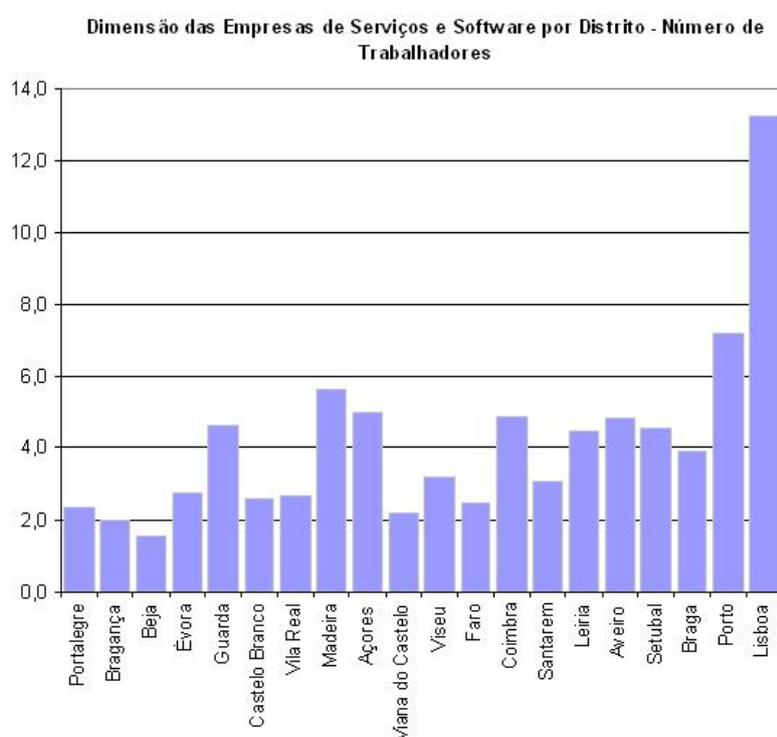


Figura 1.5 – Número de Trabalhadores por Empresa de Serviços e Software (por distrito) segundo [Beira, et al., 2006]

Pelo exposto, o tecido empresarial formado pelos produtores nacionais de software é maioritariamente caracterizado por empresas de pequena e média dimensão que, por definição, dispõem de um conjunto de recursos humanos limitado. Contudo, encontram-se frequentemente em competição directa com outras de maior dimensão, nacionais ou estrangeiras, quer baseadas em recursos próprios, quer suportadas em elaborados esquemas de subcontratação. Por isso, essa disparidade de capacidade instalada impõe um maior rigor no aproveitamento de recursos de modo a minimizar os desperdícios. Adicionalmente, os

recursos financeiros ao seu dispor costumam ser modestos, o que limita a possibilidade de beneficiar de acções de formação e de ferramentas de desenvolvimento e trabalho cooperativo mais poderosas e integradas. Logo, torna-se importante implementar processos de trabalho que rentabilizem os conhecimentos individuais e promovam uma eficaz disseminação do conhecimento entre os vários membros da equipa.

Por outro lado, este tipo de organizações apresentam vulgarmente uma estrutura orgânica (por oposição a mecânica), destinada a ser eficaz (em detrimento de eficiente). Por isso, é comum um baixo nível de normalização de processos, que tem como consequência uma elevada dependência em relação a determinados recursos chave e uma grande variância na forma como um mesmo tipo de tarefas é executado por colaboradores diferentes em momentos distintos, faltando uniformidade e propiciando alguma descoordenação. Finalmente, nem sempre uma PME possui a “massa crítica” necessária para se poderem organizar de forma funcional ou por projecto, pelo menos em moldes rígidos. Como consequência, alguns recursos humanos podem pertencer simultâneamente a várias áreas funcionais ou equipas de projecto, o que frequentemente cria problemas de incompatibilidade entre os prazos e prioridades das tarefas que lhes são solicitadas por diferentes responsáveis.

Poder-se-á dizer que nenhum dos problemas que acabam de ser descritos é específico desta área, sendo igualmente aplicável a muitas PME de outros tantos sectores de actividade. Contudo, é necessário ter presente que deles resulta uma enorme exigência em relação aos envolvidos, a quem se pede capacidade para gerir adequadamente as pressões (de várias índoles) a que são sujeito e sem deixar que afectem a sua produtividade, eficácia na concretização de objectivos, eficiência na gestão dos (usualmente poucos) recursos de que dispõem e desempenhos capazes de dirimir ou anular as insuficiências da organização e do seu contexto envolvente.

1.3. Objectivos

Tal como se viu nas secções anteriores, são muitos os factores indutores de turbulência no contexto envolvente de uma PME de desenvolvimento de software que, como grãos de areia na engrenagem, contribuem para deteriorar a eficácia e eficiência do seu processo de desenvolvimento (seja este mais ou menos formal). Assim, no sentido de minimizar (ou até mesmo eliminar) o impacto negativo desses factores, este tipo de organizações vê-se na necessidade de encontrar metodologias de potencial utilidade, entre as quais se inclui o RUP (*Rational Unified Process*), em virtude do seu abrangente leque de intervenientes. Contudo, apesar da sua (supostamente) fácil configurabilidade, não são do domínio público

configurações do RUP apropriadas para empresas de micro e pequena dimensão, como as que caracterizam a esmagadora maioria do tecido empresarial nacional. Pelo exposto, e dado não ser possível endereçar todos os problemas de uma só vez, esta dissertação tem como objectivo central demonstrar uma ou mais formas de configurar o elenco processual do RUP de modo a que, sem negligenciar nenhuma função crítica do processo de desenvolvimento de software, este possa ser facilmente adoptado no contexto específico de uma organização PME durante o período de execução¹¹ de um projecto. Nesse sentido, proceder-se-á a uma rigorosa selecção dos principais intervenientes no RUP e à colmatação de eventuais lacunas detectadas, para assim chegar à composição dos modelos a propor e recomendar o respectivo enquadramento organizacional.

Adicionalmente, assume-se também como objectivo taxionomizar as incumbências dos participantes nos elencos processuais propostos através da criação de fichas individuais que contribuam para que cada indivíduo saiba exactamente o que dele se espera e de que forma deve operacionalizar o seu contributo. Com isso, pretende-se: (1) evitar a sobreposição e/ou indefinição de responsabilidades; (2) minimizar a ocorrência de erros causados por esquecimento ou falta de consciência das mesmas; (3) tornar mais simples para uma organização a adopção dos modelos apresentados, independentemente de se encontrar ou não familiarizada com o RUP.

Finalmente, é conveniente clarificar que não é objectivo desta dissertação avaliar em contextos reais a aplicação dos resultados dos dois objectivos enunciados anteriormente, já que isso iria inevitavelmente implicar um consumo de tempo significativo e considerado essencial para prosseguir os objectivos assumidos anteriormente. Em vez disso, pretende-se apenas recolher alguns indícios sobre a sua adequabilidade a um contexto PME específico.

1.4. Estruturação do Documento

Para além do actual, esta dissertação desenvolve-se ao longo de quatro outros capítulos, sendo que o **Capítulo 2** apresenta o *Rational Unified Process*, desde a sua génese até à inerente arquitectura processual (decomposta em fases e disciplinas), passando pelas principais ideias-chave em que se alicerça e pelos conceitos essenciais à sua compreensão (actividades, artefactos, papéis e fluxos). Adicionalmente, procede-se a uma detalhada caracterização e reconstituição do elenco processual necessário à operacionalização do

¹¹ Não abrangendo o período a montante (que leva à concretização da oportunidade de o realizar) nem a juzante (ao longo do qual têm lugar as actividades de exploração e manutenção).

processo, identificando incongruências e omissões existentes entre várias fontes de informação que versam sobre o mesmo. Finalmente, são ainda analisados alguns exemplos de aplicação do *Rational Unified Process* que, pela sua natureza, se consideram relevantes para organizações do tipo PME.

Tomando como base o elenco processual previamente apresentado, o **Capítulo 3** descreve um processo de simplificação levado a cabo com o objectivo de obter um modelo (denominado de base) passível de ser efectivamente aplicado em contextos PME, consubstanciado num conjunto de treze papéis distintos. Contudo, e antecipando o caso de este modelo ainda poder ser considerado como demasiadamente exigente para ser adoptado por organizações de reduzida dimensão, é também proposto um modelo alternativo (denominado de reduzido) que, fruto de uma condensação e reorganização de funções, reduz para oito a dimensão do lote de intervenientes processuais. Adicionalmente, e para evitar a ocorrência de situações que possam lesar a eficácia dos modelos propostos, são ainda recomendadas algumas restrições à acumulação de determinados papéis, quer dentro do mesmo projecto quer entre projectos distintos. Por outro lado, e no sentido de auxiliar as organizações adoptantes a enquadrar devidamente o modelo por si seleccionado, é apresentado um mapeamento entre os vários papéis e as funções usualmente encontradas numa PME ligada ao desenvolvimento de software.

No **Capítulo 4** procede-se a uma minuciosa caracterização dos vários intervenientes nos modelos anteriormente propostos, visando contribuir dessa forma para potenciar a sua eficácia e facilitar a sua operacionalização. Nesse sentido, as relações entre os vários papéis são representadas sob a forma dos fluxos comunicacionais (internos ou externos) que as suportam e que traduzem as principais interacções que têm lugar entre eles. Para que seja possível à organização seleccionar os indivíduos mais apropriados para cada papel, para além de assegurar que cada um deles sabe exactamente as funções que é chamado a assumir, são ainda apresentadas fichas individuais que definem as acções a encetar por cada interveniente em determinados momentos-chave do projecto (início de projecto, início de iteração, durante a iteração, fim de iteração e fim de projecto).

Finalmente, o **Capítulo 5** tem como objectivo transmitir algumas conclusões que resultam da aplicação num contexto PME de um misto dos modelos propostos nesta dissertação, pelo que poderão ser de utilidade para organizações que ponderem a sua adopção. Na sequência dos problemas detectados e das vertentes dos modelos que não foi possível explorar, fruto da exiguidade temporal de uma dissertação deste género, são, ainda, identificadas possíveis formas para dar seguimento ao trabalho desenvolvido.

2. O *Rational Unified Process* e suas Aplicações

Este capítulo apresenta o *Rational Unified Process*, desde a sua génese até à inerente arquitectura processual (decomposta em fases e disciplinas), passando pelas principais ideias-chave em que se alicerça e pelos conceitos essenciais à sua compreensão (actividades, artefactos, papéis e fluxos). Adicionalmente, procede-se a uma detalhada caracterização e reconstituição do elenco processual necessário à operacionalização do processo, identificando incongruências e omissões existentes entre várias fontes de informação que versam sobre o mesmo. Finalmente, são ainda analisados alguns exemplos de aplicação do *Rational Unified Process* que, pela sua natureza, se consideram relevantes para organizações do tipo PME.

2.1. Introdução

A exigência dos contextos competitivos actuais tem vindo a traduzir-se, de forma inevitável, numa necessidade e requisitos crescentes por parte dos intervenientes no negócio em relação às funcionalidades disponibilizadas pelos sistemas de informação que pretendem adoptar nas suas organizações. Como consequência destas solicitações, os projectos desenvolvidos são cada vez mais abrangentes, empregam tecnologias mais complexas¹², necessitam de suportar cargas de utilização maiores, destinam-se a utilizadores mais exigentes¹³ e necessitam de reagir a pedidos de alteração mais frequentes e/ou significativos, que carecem de uma gestão adequada e eficaz. Pelo exposto, as organizações produtoras de software vêem-se na necessidade de encontrar um processo de desenvolvimento que lhes permitam impor alguma ordem no caos de geração espontânea em que exercem a sua actividade, na sua senda para atingir níveis qualitativos elevados, eficiência na gestão de

¹² Em virtude da cada vez maior potencialidade destas.

¹³ Que já não ficam deslumbrados por apenas terem acesso ao sistema.

recursos, redução do risco e aumento da previsibilidade do seu funcionamento. Segundo [Jacobson, *et al.*, 1999], um **processo de desenvolvimento de software** pode ser definido como “*um conjunto completo das actividades necessárias para transformar requisitos num conjunto consistente de artefactos que representem um produto de software e, posteriormente, para transformar alterações a esses requisitos num novo conjunto de artefactos*”. Ou seja, um processo corresponde a um conjunto de actividades e não à sua execução, com benefícios óbvios para as equipas de trabalho, já que:

- clarifica as responsabilidades dos seu diferentes elementos;
- proporciona orientação para a priorização das actividades da equipa;
- direcciona as tarefas de cada elemento da equipa e desta como um todo;
- especifica os artefactos que devem ser produzidos e quando;
- oferece critérios para monitorizar e medir o progresso, produtos e actividades do projecto.

Contudo, o desenvolvimento de software já é suficientemente arriscado sem a utilização de metodologias sem provas dadas em contextos de complexidade equivalente, pelo que é compreensível que as organizações se rodeiem de cuidados na selecção do processo que pretendem adoptar. O *Rational Unified Process* [Kruchten, 2004] é um conhecido processo de desenvolvimento de software, que estende¹⁴ o *Unified Process* [Jacobson, *et al.*, 1999], que por sua vez resultou da integração e evolução de processos mais antigos como sejam o *Rational Approach* [Booch, *et al.*, 2007] e o *Objectory Process* [Ivar Jacobson, *et al.*, 1992]. Como o próprio nome indica, a génese do RUP deu-se no interior da *Rational Software Corporation*, que em 2003 passou a fazer parte do *IBM Software Group*. Desde então, o RUP faz parte do leque de produtos da *Big Blue* que, dados os seus mercados-alvo, o tem levado à prática em organizações de grande dimensão, estimando-se [Kroll, Kruchten, 2003] que existam mais de 10.000 organizações a utilizar o produto, em projectos de pequena e grande envergadura. Na verdade, os seus autores realçam como ponto forte da metodologia o seu interesse em aprender com a experiência das organizações que a aplicam, publicando regularmente versões actualizadas da mesma.

¹⁴ Adicionando-lhe detalhe em áreas como a modelação de negócio, a gestão de projectos e a gestão de configurações.

Assim, o RUP constitui uma boa alternativa para produtores de software que enfrentem desafios exigentes, pelo que na secção que se segue se irá proceder a uma análise das suas principais características.

2.2. O Rational Unified Process

Segundo o RUP, um processo de desenvolvimento guia os esforços das pessoas envolvidas num projecto, ao proporcionar-lhes um modelo dos passos a seguir no sentido de criar um produto de software, auxiliados por um conjunto de ferramentas. A Figura 2.1 ilustra esta relação.

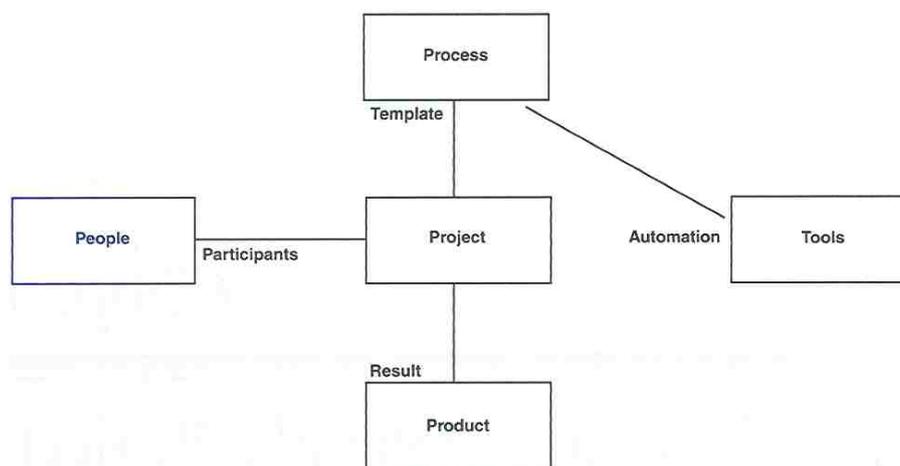


Figura 2.1 – 4Ps do Desenvolvimento de Software

Assim, o RUP apresenta-se como uma abordagem disciplinada para a atribuição de tarefas e responsabilidades no interior de uma organização, com o objectivo de assegurar a produção de software de alta qualidade, que satisfaça as necessidades dos seus utilizadores e no estrito cumprimento de um orçamento e calendário previsível. Para isso, os seus autores decidiram alicerçá-lo em três ideias-chave:

1. **Use-Case Driven (Orientado aos Casos de Uso):** O sucesso de um projecto de desenvolvimento de software depende de forma insofismável da sua adequação às expectativas e reais necessidades dos indivíduos que irão interagir com o(s) sistema(s) a implementar. Contudo, para que tal possa¹⁵ ocorrer é essencial que se consiga chegar tão cedo quanto possível a uma visão partilhada entre cliente e fornecedor sobre quais os requisitos funcionais que o projecto visa satisfazer. Assim, e dada a ligação umbilical entre este processo e a notação UML (*Unified*

¹⁵ É condição necessária mas não suficiente.

Modelling Language) [Booch, Rumbaugh, 1999], o RUP elege os diagramas de casos de uso (*use case*) como o mecanismo ideal para a captura deste tipo de requisitos¹⁶, em detrimento de meras listagens de funcionalidades a assegurar, alegando duas principais ordens de razões:

- Colocam o foco nos utilizadores/actores e não directamente nas funcionalidades do(s) sistema(s). Dessa forma, considera-se o que é relevante para cada um deles ao mesmo tempo que se evitam investimentos supérfluos em funcionalidades não valorizadas nem necessárias a nenhum deles;
- A sua facilidade de interpretação, permitindo uma fácil validação por parte de todos os intervenientes (internos e externos) e minimizando falhas de entendimento de parte a parte.

Para além disso, o modelo de casos de uso¹⁷ serve de base de trabalho a vários intervenientes que o usam como fio condutor das suas actividades, sejam estas de arquitectura, concepção, implementação ou teste, motivo pelo qual se afirma que o processo é conduzido pelos casos de uso. Todavia, [Henderson-Sellers, *et al.*, 2001] considera que a mera focalização nos casos de uso não é apropriada, por não endereçar alguns aspectos (entre os quais os importantes requisitos não funcionais) que não são passíveis desse tipo de representação. Em contrapartida, este artigo apresenta o OPEN (*Object-Oriented Processes, Environment and Notation*) como uma alternativa que, para além de apresentar uma maior flexibilidade na representação e gestão da miríade de requisitos que caracterizam um projecto de desenvolvimento de software, possibilita a utilização de uma qualquer notação de modelação (e não forçosamente da UML);

2. *Architecture-Centric (Centrado na Arquitectura):* Após delinear a primeira versão do modelo de casos de uso, é essencial gizar uma arquitectura de software que permita a concretização dos requisitos funcionais que lá se encontram retratados. Porém, esse processo conceptual carece de vários outros ingredientes que não constam desse modelo: requisitos não funcionais¹⁸, plataformas a

¹⁶ Obviamente complementados, quando necessário, com outros documentos que detalhem a abrangência de cada um deles.

¹⁷ Que é ciclicamente revisto e refinado.

¹⁸ Relacionados com tempos máximos de resposta, número de pedidos em simultâneo, capacidade de tolerância a falhas, etc.

utilizar/suportar¹⁹, protocolos/normas²⁰ a utilizar, restrições de implementação/integração²¹ e componentes passíveis de reutilização. Depois de ponderar tudo isto, e após criar o esboço inicial da arquitectura de software, é necessário detalhar os principais casos de uso em termos de interacção entre sub-sistemas, componentes e classes. Esse processo ajudará o arquitecto a focar-se, procurando uma arquitectura de simples compreensão, reutilizável e facilmente adaptável às futuras e inevitáveis alterações. Para além disso, ao fazê-lo dar-se-á início a um ciclo virtuoso de refinamento alternado do modelo de casos de uso e da arquitectura, até se obterem versões suficientemente estáveis de ambos. No final, a arquitectura resultante servirá como perspectiva enquadradora para todos os intervenientes, evidenciando o(s) sistema(s) que sustenta(m) o modelo de casos de uso. Contudo, apesar de ser uma pretensão da metodologia assegurar esta centralidade da arquitectura, [Hesse, 2003] sustenta (como será analisado em maior detalhe na secção 2.4) que a importância desta acaba por ser secundarizada pela subdivisão do RUP nas quatro fases identificadas na Figura 2.2, sugerindo a adopção de outro tipo de abordagem (concretizada no modelo EOS - *Evolutionary, Object-oriented Software Development*);

3. ***Iterative and Incremental (Iterativo e Incremental)***: À semelhança dos denominados *métodos ágeis*, também o RUP é contrário à perspectiva mais tradicionalista do desenvolvimento em cascata, caracterizada por uma entrega *big bang* no final, após a qual são tipicamente detectadas as (por vezes já insanáveis) disparidades entre as expectativas de produtor e consumidor do software. Em vez disso, propõe que cada projecto se desenrole como uma sequência de pequenas etapas (chamadas de *iterações*) que, de uma forma incremental, vão evoluindo no sentido de concretizar a totalidade dos objectivos do projecto. As motivações subjacentes são várias:

- Redução do risco de equívocos sobre as reais necessidades dos *stakeholders* do projecto, potencionando o seu envolvimento tão cedo quanto possível e

¹⁹ Em virtude do que o arquitecto de software considera mais apropriado e de eventuais requisitos para as plataformas (sistemas operativos, motores de bases de dados, servidores aplicativos, etc) em que o sistema deverá funcionar.

²⁰ Para a comunicação, segurança da informação, redundância de armazenamento, etc.

²¹ Resultantes de sistemas *legacy* com que será necessário integrar, políticas corporativas existentes, de barreiras à comunicação, de inexistência de recursos, etc.

proporcionando-lhe vários pontos de controlo para que possa avaliar se cada iteração corresponde a um passo na direcção que pretende;

- Descrença na eficácia de projectos BRUF²² já que, como qualquer profissional da área saberá, são raros os casos em que os requisitos ditados pelo cliente no início do projecto coincidem com os verificados no final. Assim, o processo iterativo proporciona a capacidade de inflexão necessária para gerir adequadamente a mudança de prioridades²³ e/ou do contexto²⁴ do projecto. Note-se que, em virtude dessas possíveis alterações, nem sempre as iterações são incrementais, podendo envolver a alteração ou eliminação de funcionalidades já existentes;
- Não privar o cliente de utilizar qualquer funcionalidade até ao término do projecto, permitindo-lhe usufruir gradualmente destas, consoante vão sendo concluídas as várias iterações e até, se necessário, reduzir o *time-to-market* à custa de um menor conjunto de funcionalidades;
- Minimizar a probabilidade de derrapagem de prazos em projectos de longa duração ao assegurar a constante existência de objectivos a curto prazo para cumprir.

Contudo, e como será descrito na secção 2.4, [Hesse, 2003] questiona também a eficácia da adopção de iterações quando estas se encontram compartimentadas no interior de fases distintas, por considerar que as decisões tomadas em fases anteriores poderão condicionar (de uma forma não desejada) o rumo a seguir nas iterações posteriores.

²² *Big Requirements Up Front*.

²³ Requisitos iniciais que deixam de fazer sentido, novos requisitos que passam a ser considerados essenciais, etc.

²⁴ Alteração legislativa que afecte a área, lançamento de um produto/serviço concorrente, etc.

Em traços mais gerais, a Figura 2.2 representa a arquitectura processual do RUP.

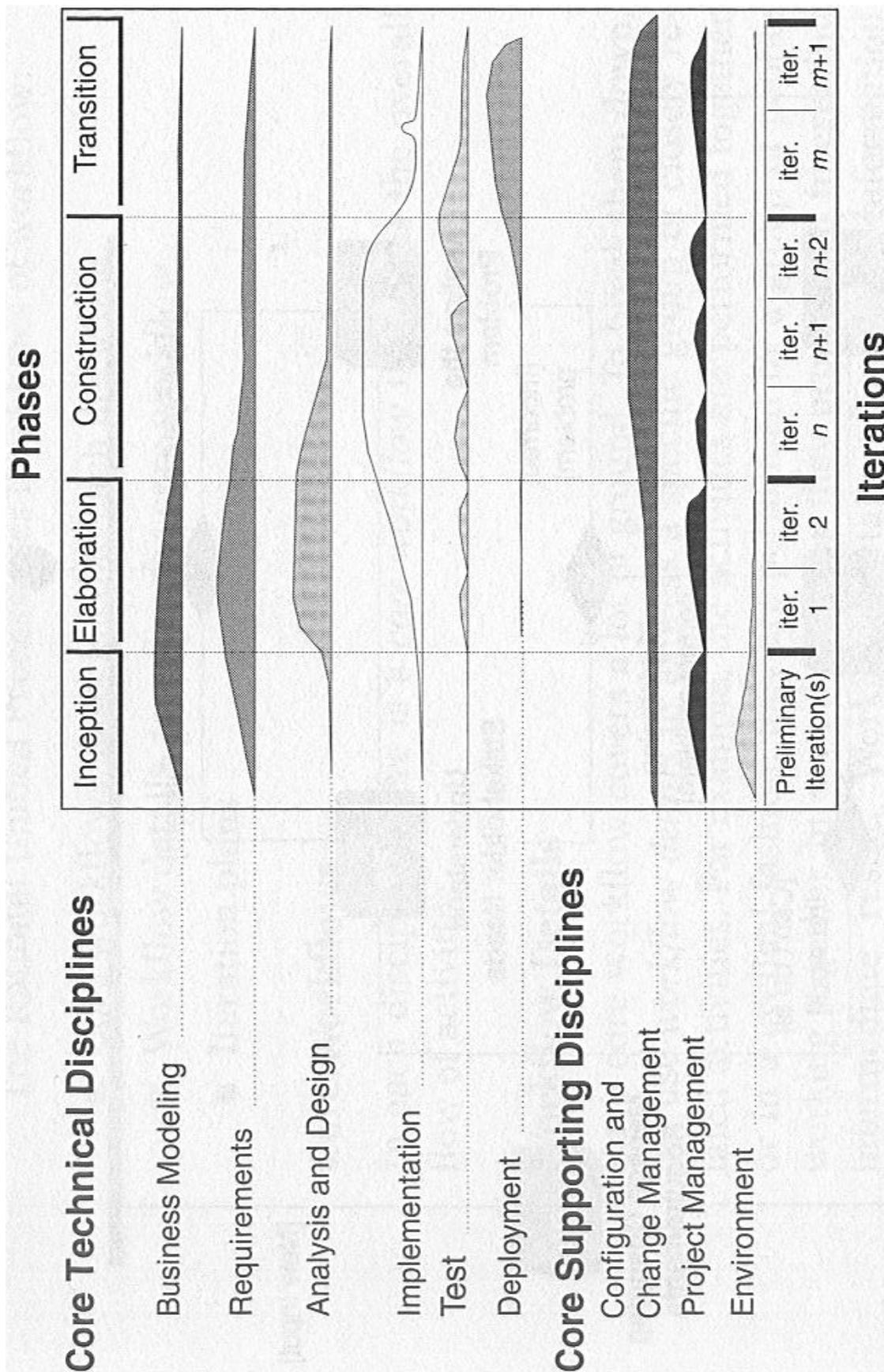


Figura 2.2 – Arquitectura Processual do RUP

Como é dado a observar, esta arquitectura processual encontra-se estruturada em duas dimensões distintas:

- **Tempo (eixo das abcissas):** Representa o aspecto dinâmico do processo, que se traduz no seu ciclo de vida. Encontra-se organizado de acordo com as seguintes fases:
 - *Inception* (ou Esboço): Consiste na definição dos objectivos do projecto, seu âmbito e respectivo modelo de negócio. Culmina com a concretização de um marco²⁵ que a metodologia apelida de LCO (*Lifecycle Objective Milestone*);
 - *Elaboration* (ou Detalhe): Consiste na criação e validação da arquitectura do(s) sistema(s) de software, capturando os requisitos mais importantes do projecto, estimando os recursos necessários à sua implementação e planeando o desenrolar da mesma. Esta fase conclui-se com o marco denominado de LCA (*Lifecycle Architecture Milestone*);
 - *Construction* (ou Construção): Consiste na implementação do(s) sistema(s) de software, de acordo com a arquitectura definida na fase anterior e assegurando a sua evolução até que esteja pronto para ser disponibilizado à comunidade de utilizadores, momento esse que corresponde ao marco IOC (*Initial Operational Capability Milestone*);
 - *Transition* (ou Transição): Consiste no processo de transição do(s) sistema(s) desenvolvidos para a comunidade de utilizadores finais, incluindo a condução da fase final de testes à(s) versão(ões) candidata(s) por parte dos *beta testers*, a formação dos utilizadores e a preparação das actividades de manutenção e suporte de que estes irão necessitar. Assim, o ciclo de desenvolvimento chega ao fim com o marco PR (*Product Release Milestone*).

Cada uma destas fases (que [Hesse, 2003] considerada inadequadas para lidar com a complexidade dos actuais projectos de desenvolvimento de software) pode ainda ser decomposta em uma ou mais iterações, que parte dos resultados da iteração anterior com o objectivo de entregar uma nova versão executável dos artefactos de software, com um conjunto de funcionalidades mais aproximado dos requisitos

²⁵ Denominado pela terminologia inglesa de *milestone*.

definidos. A duração e objectivos de cada iteração devem ser definidos antes do seu início, de modo a permitir um maior controlo da mesma.

- **Disciplinas (eixo das ordenadas):** Representa o aspecto estático do processo, que se traduz no tipo de actividades levadas a cabo num processo de desenvolvimento de software, de acordo com a sua natureza:
 - *Business Modelling* (ou Modelação de Negócio): Actividades de descrição dos processos e estrutura do negócio, de modo a compreender melhor o mesmo e a identificar os requisitos mais importantes para o sistema a ser desenvolvido;
 - *Requirements* (ou Requisitos): Actividades de explicitação, organização e documentação de requisitos;
 - *Analysis and Design* (ou Análise e Desenho): Actividades de criação da arquitectura e concepção do(s) sistema(s) de software;
 - *Implementation* (ou Implementação): Actividades de criação e *debug* de código fonte e testes unitários;
 - *Test* (ou Teste): Actividades de testes de integração, de sistema e de aceitação;
 - *Deployment* (ou Distribuição): Actividades de criação do pacote de instalação, escrita da documentação para utilizadores e outras tarefas similares;
 - *Configuration and Change Management* (ou Gestão de Configurações e da Mudança): Actividades relacionadas com a gestão de versões e de pedidos de alteração;
 - *Project Management* (ou Gestão de Projecto): Actividades de planeamento e monitorização do projecto;
 - *Environment* (ou Ambiente): Actividades de adaptação do processo às necessidades do projecto (ou da organização) e selecção, introdução e suporte de ferramentas de desenvolvimento.

Adicionalmente, cada uma destas **actividades** (ou *activities*) corresponde a uma unidade de trabalho que um interveniente pode ser chamado a executar e que recebem e/ou produzem um ou mais artefactos, dando origem a um resultado relevante no contexto do

projecto. Estes **artefactos** (ou *artifacts*) são pedaços de informação que são produzidos, modificados ou utilizados por um processo, entregues à responsabilidade de um determinado papel (embora possam ser alterados por outros). Ou seja, são os produtos de trabalho (documentos, código fonte, modelo UML, etc.) tangíveis criados ou utilizados pelo projecto, constituindo “o quê ?” do processo. Note-se que o RUP apenas apelida de *deliverables* (termo frequentemente utilizado em projectos de desenvolvimento de software) os artefactos que efectivamente são entregues aos clientes/utilizadores finais. Assim sendo, as actividades são “o como ?” do processo e, dado que cada uma se encontra atribuída a um papel específico, definem o trabalho levado a cabo por cada um deles, embora a dimensão e complexidade inerente a algumas actividades implique a concertação de vários esforços individuais.

Dado que artefactos e actividades apenas definem (respectivamente) “o quê ?” e “o como ?”, a metodologia endereça “o quem ?” tipificando não só o elenco processual existente²⁶, como também os **papéis** (ou *roles*) que cada interveniente assume nos vários fluxos que compõem o processo. Dessa forma, é possível proporcionar a cada *worker*, que pode desempenhar diferentes *roles* em cada fluxo, uma visão clara de quais as suas incumbências ao longo do projecto. Finalmente, a metodologia define ainda um conjunto de **fluxos** (ou *workflows*) destinados a responder ao “quando ?”, ao definir sequências de actividades que produzem um resultado de valor facilmente observável. Assim, para cada uma das disciplinas identificadas anteriormente, existe um fluxo que determina o encadeamento aconselhado para as actividades que a compõem.

Actualmente, a metodologia compreende mais de oitenta artefactos, cento e cinquenta actividades e cerca de quarenta papéis. Para além disso, e não obstante todas as recomendações e boas práticas que o RUP encerra, há um conjunto de princípios-base (que [Kroll, Kruchten, 2003] apelida de *Espírito do RUP*) que, não esgotando o seu conteúdo, o sintetizam e diferenciam de outros métodos ágeis. O primeiro sustenta que um dos erros mais perigosos que se podem cometer na execução de um projecto é descurar a gestão de riscos técnicos conhecidos, ignorando-os ou adiando a sua eliminação/redução. Esta atitude é muitas vezes justificada pelo desejo de mostrar rapidamente resultados e, em contextos de menor concorrência de acessos e necessidade de desempenho, poderá criar uma falsa sensação de confiança no rumo seguido. Porém, com o aproximar do final do projecto poderão começar a evidenciar-se os efeitos de uma arquitectura inadequada ou de um

conjunto não otimizado de requisitos, numa fase em que já foram investidos recursos significativos e em que a capacidade de inflexão se encontra bastante condicionada. Para o evitar, o RUP sugere que no início de cada iteração seja criada/revista²⁷ uma lista dos principais riscos conhecidos, seleccionando-se não mais do que cinco como prioritários para essa iteração e definindo, para cada um deles, qual a estratégia a seguir para o eliminar/mitigar nas mais variadas perspectivas²⁸. Note-se que, dado que alguns riscos têm origem na arquitectura de software a utilizar, a gestão de risco também passa pela sua validação. Ainda que pareça uma *verdade de La Palisse*, outro ponto essencial passa por assegurar a criação de valor para o cliente, já que são frequentes os projectos em que, apesar do fornecedor estar convencido que tudo foi feito da melhor forma, o cliente não fica total/parcialmente satisfeito com os resultados do projecto, situação tão mais provável quanto maior for o alheamento do cliente em relação ao decorrer dos trabalhos. Assim, e como já foi referido anteriormente, deve tirar-se partido do processo iterativo para assegurar o envolvimento do cliente e para ir obtendo as reacções deste ao trabalho desenvolvido até ao momento. Adicionalmente, o RUP propõe a utilização de diagramas de casos de uso e de protótipos de interface de utilizador não só para validar os requisitos junto do cliente, como também para manter a equipa de projecto focada no que realmente é importante para ele.

Para além dos óbvios artefactos executáveis, o RUP propõe a criação de uma vasta lista de artefactos documentais de suporte (casos de uso, planos de iteração, planos de teste, etc.) que podem tornar o software mais fácil de implementar, mais eficaz ou com menor custo de manutenção. Porém, os elementos dessa lista não devem ser considerados como obrigatórios e só deverão ser criados se os benefícios inerentes compensarem²⁹ o seu custo de produção. Nesse sentido, em [Hirsch, 2002], [Fernandes, Duarte, 2005] e [Duarte, *et al.*, 2006] poderão encontrar-se descrições relevantes sobre como proceder a uma selecção de artefactos que, sem criar um alforge pesado demais para uma PME, possibilite uma razoável documentação do processo de desenvolvimento de software levado a cabo na organização. No entanto, a metodologia defende que deve ser dada primazia ao software executável dado que só através desse é que poderá ter uma percepção realista sobre o progresso do projecto, evitando

²⁶ Que o RUP apelida de *workers*.

²⁷ Dado que apesar de alguns riscos irem sendo minimizados/eliminados, o desenrolar do projecto dá origem a novos riscos.

²⁸ Requisitos, concepção, implementação, teste, etc.

²⁹ Como regra base podemos considerar que se temos dúvidas sobre a mais-valia de um determinado artefacto é porque provavelmente não deve ser produzido.

recorrer a outros indicadores bem mais falíveis (como, por exemplo, o número de casos de uso implementados). Dessa forma, é possível desenvolver actividades de teste praticamente desde o início do projecto, permitindo uma verificação gradual da cobertura dos testes e do sucesso dos casos de teste. Uma outra vantagem desta abordagem é evitar que a equipa de projecto prolongue em demasia o seu esforço de análise e de teorização para encontrar a solução óptima, levando-a a pôr rapidamente em prática as suas ideias e contribuindo com isso para uma redução do risco do projecto. Convém também não esquecer que a maior parte dos projectos de desenvolvimento de software implementados actualmente reveste-se de uma complexidade tal que é virtualmente impossível identificar, à partida, todos os requisitos relevantes. Logo, a mudança é uma inevitabilidade, motivo pelo qual deverá ser encarada de uma forma positiva e como um caminho para melhorar a solução e aumentar o valor para o cliente. Contudo, não devemos ignorar que a mudança tem consequências, sejam elas de maior³⁰ ou menor³¹ gravidade, e que podem variar consoante a fase³² em que o projecto se encontra. Por isso, é essencial pôr em prática procedimentos para o tratamento de pedidos de alteração que assegurem, após a avaliação e minimização do impacto da mudança, a aplicação de regras claras para decidir sobre o seu deferimento.

A arquitectura de um projecto compreende a sua divisão nos sub-sistemas/componentes mais importantes e a definição das interfaces que suportam a integração do seu funcionamento, juntamente com os mecanismos arquitecturais escolhidos para endereçar problemas comuns (persistência, balanceamento de carga, *garbage collection*, etc.). Dessa forma, define o esqueleto da solução, estimando-se que, consoante a dimensão do projecto, corresponda a cerca 10% a 20% do código fonte total desenvolvido. Assim, o RUP propõe que a concretização deste “esqueleto” funcional seja prioritária, no sentido de verificar se é viável e se tem potencial para corresponder aos requisitos (nomeadamente não funcionais) definidos. Recorrendo a uma analogia, será como dizer que se deve dar prioridade a levantar as paredes de uma casa, em vez de começar por fazer os armários da cozinha, o que parece fazer todo o sentido. Para além de reduzir o risco técnico associado à execução do projecto, esta abordagem permite também obter estimativas mais precisas, quer sobre os recursos

³⁰ Atraso na finalização do projecto, custos de alteração acrescidos e significativos, redução notória de qualidade (por exemplo, em virtude de uma degradação do desempenho), etc.

³¹ Mudança de configurações das aplicações, falta de tempo para implementar outra(s) funcionalidade(s) de reduzida relevância, etc.

³² As alterações à arquitectura de negócio devem evitar-se após a fase de *Inception*, as alterações à arquitectura tecnológica devem evitar-se após a fase de *Elaboration*, as alterações ao desenho e implementação não devem acontecer após a fase de *Construction*, enquanto uma redução de âmbito pode usualmente ser realizada em qualquer momento.

necessários à sua execução, quer sobre os prazos de implementação do mesmo. Adicionalmente, o facto de se consolidar rapidamente a arquitectura torna mais fácil, se necessária, a introdução de novos elementos menos experientes na equipa, proporcionando-lhes o contexto onde podem/devem desenvolver a sua actividade. Por outro lado, o RUP incentiva o desenvolvimento de software baseado em componentes dado que, entre outras coisas, assegura que os dados e as funcionalidades que sobre eles operam são encapsulados por interfaces num mesmo local/componente, facilitando alterações ao modelo de dados ou à lógica de negócio. Para além disso, dessa forma os sistemas implementados tornam-se mais flexíveis, potencia-se a reutilização, pode beneficiar-se de componentes já existentes e com provas dadas, ao mesmo tempo que se reduzem os custos de manutenção.

Embora o trabalho em equipa seja crucial nos mais variados tipos de actividades, não deixa de ser verdade que é essencial para lidar adequadamente com a complexidade que caracteriza os actuais projectos de desenvolvimento de software. Por isso, ainda que a metodologia seja importante para orientar as actividades dos indivíduos envolvidos, são eles o elemento mais crucial para o sucesso do projecto. Assim, não só é necessário que lhes sejam dadas condições de trabalho propícias³³, como também assegurar que todos os seus elementos³⁴ se encontram preparados para trabalhar de forma iterativa. Para isso, o RUP propõe a criação de equipas multifuncionais, a utilização de ferramentas que facilitem a comunicação³⁵/colaboração e uma boa gestão de recursos humanos por parte do gestor de projectos³⁶. É também recomendável que, quando as equipas atingem dimensões muito elevadas, se crie uma equipa de arquitectos - responsável por definir os sub-sistemas e suas interfaces – e várias equipas multifuncionais, cada uma responsável por um sub-sistema. No que diz respeito à qualidade, é frequente verificar que por vezes é relegada para segundo plano (com potenciais efeitos indesejados³⁷) em contextos onde se verifique uma elevada competitividade entre os produtores de software ou forte restrição orçamental por parte dos respectivos clientes, dada a grande tentação para minimizar os seus custos de produção e, assim, conseguirem ser (pelo menos em preço) mas competitivos. Por isso mesmo, o RUP

³³ Incluindo uma apropriada estrutura organizacional (uma organização funcional pode dificultar a comunicação entre os elementos, uma organização em matriz pode exagerar na reutilização de recursos, etc).

³⁴ Alguns elementos menos técnicos (o gestor de projecto, por exemplo) poderão não estar habituados a fazê-lo.

³⁵ Potenciando sempre que possível a comunicação cara-a-cara ou telefónica/video-conferência, em detrimento do e-mail.

³⁶ De modo a efectuar uma adequada mobilização dos mesmos, assegurar o seu comprometimento para com o projecto e criar um sentimento de co-responsabilização entre todos.

³⁷ Documentação inexistente/incompleta, software com problemas de funcionamento, etc.

defende que o sucesso a longo prazo de um produtor de software deve ser assente na qualidade dos projectos que desenvolve, promovida através do desenvolvimento iterativo³⁸, do ataque inicial aos principais riscos³⁹, da maior automatização possível do processo de testes, de um sistema de revisão de artefactos antes da sua entrega ao exterior e, finalmente, de um processo de aculturação dos envolvidos⁴⁰ que os sensibilize para o facto de a qualidade do todo ser alicerçada na qualidade das partes.

Convém ainda referir que o RUP é proposto pelos seus autores como um *process framework*, cujos conteúdos (artefactos, actividades, papéis, etc.) podem (embora, segundo [Henderson-Sellers, *et al.*, 2001], com reduzido grau de liberdade) e devem ser adaptados à realidade e necessidades da organização adoptante. No sentido de facilitar esse esforço de adequação, a organização do processo baseia-se no meta-modelo orientado aos objectos SPEM (*Software Process Engineering Metamodel*) [OMG, 2001], que o divide em componentes processuais que podem ser combinados entre si da forma mais apropriada ao contexto de aplicação.

Finalmente, para além dos conceitos-chave que acabam de ser analisados, falta relembrar a faceta comercial do RUP que, obviamente, necessita de ser traduzida em mais-valia quando comparada com a vasta obra publicada⁴¹ sobre a metodologia. Assim, enquanto produto, o RUP compreende ainda ferramentas⁴² destinadas a auxiliar os vários *Workers* a fazer o seu trabalho, ajudando-os a saber o que devem fazer em cada etapa do processo, ao mesmo tempo que os isolam da informação que não lhes é relevante. Entre as funcionalidades permitidas por estas ferramentas encontram-se as seguintes: modelação UML; gestão de *bugs* de software; gestão e automatização de testes funcionais de regressão; gestão e automatização de testes de desempenho e escalabilidade; gestão do processo de testes manuais; gestão de requisitos; gestão e controlo de versões; configuração e customização do processo. Adicionalmente, e no sentido de minimizar o trabalho envolvido na produção de uma boa parte dos artefactos propostos pela metodologia, são também

³⁸ Que permite não só iniciar os testes mais cedo como também fazê-los mais frequentemente, incluindo testes de regressão que assegurem que os desenvolvimentos anteriores se mantêm funcionais. Para além disso, o simples facto de se começar cedo a planear o esforço de testes já contribui para uma qualidade acrescida.

³⁹ Fazendo com que no final do projecto os principais factores de risco já se encontrem sob teste há vários meses.

⁴⁰ Incluindo os clientes já que, caso contrário, poderão não valorizar adequadamente a qualidade dos artefactos entregues.

⁴¹ De que são exemplo [Kruchten, 2004], [Kroll, Kruchten, 2003], [Bergström, Råberg, 2003], [Kroll, MacIsaac, 2006] e [Pollice, *et al.*, 2003]

⁴² Integradas com o popular IDE (*Integrated Development Environment*) *Eclipse*.

disponibilizados modelos de vários tipos de documentos, passíveis de fácil adaptação ao contexto do projecto em causa. Por último, o pacote comercial incorpora ainda modelos de configuração do processo apropriados a determinados contextos frequentes (pequenos projectos, projectos formais, etc.) que, através de uma pré-selecção de um determinado subconjunto de componentes processuais, permitem à organização adoptar a metodologia de uma forma mais condizente com a sua realidade.

2.3. Intervenientes no RUP

Agora que o processo como um todo se encontra genericamente apresentado, é altura de proceder a uma análise pormenorizada do elenco processual inerente à aplicação do RUP, no sentido de melhor caracterizar a base de trabalho que proporciona às organizações que o adoptam. Logo, dado que a génese do RUP se encontra no *Unified Process* [Jacobson, *et al.*, 1999], foi este o ponto de partida escolhido, tendo como objectivo encontrar uma definição detalhada e holística de cada um dos papéis envolvidos. Contudo, a narrativa da obra privilegia a apresentação do processo através das suas ideias-chave, fases e fluxos, relegando para segundo plano (não de relevância mas sim de notoriedade) a dimensão humana do mesmo. Assim, em vez de cada papel ser caracterizado em profundidade e de uma só vez, os autores optaram por ir descrevendo a sua participação em cada um dos fluxos apresentados, dispersando a definição das respectivas responsabilidades por vários capítulos. Porém, para além de confusa para o leitor, esta abordagem apenas refere explicitamente a participação de dez intervenientes distintos, fazendo-o de forma pouco formal e ficando muito aquém dos mais de trinta papéis que compõem o elenco processual do RUP. De facto, a Figura 2.3 representa a exposição mais sistematizada que é possível encontrar na obra no que concerne à delimitação das esferas de intervenção dos dez papéis referenciados⁴³.

As secções que se seguem apresentam, pela ordem pela qual são referenciados em [Jacobson, *et al.*, 1999], a descrição (tão fiel quanto possível) que é possível recuperar da obra sobre cada um dos papéis identificados na Figura 2.3.

⁴³ A Figura 2.3 refere (por lapso) o papel *Test Engineer* em vez do correcto *Test Designer*.

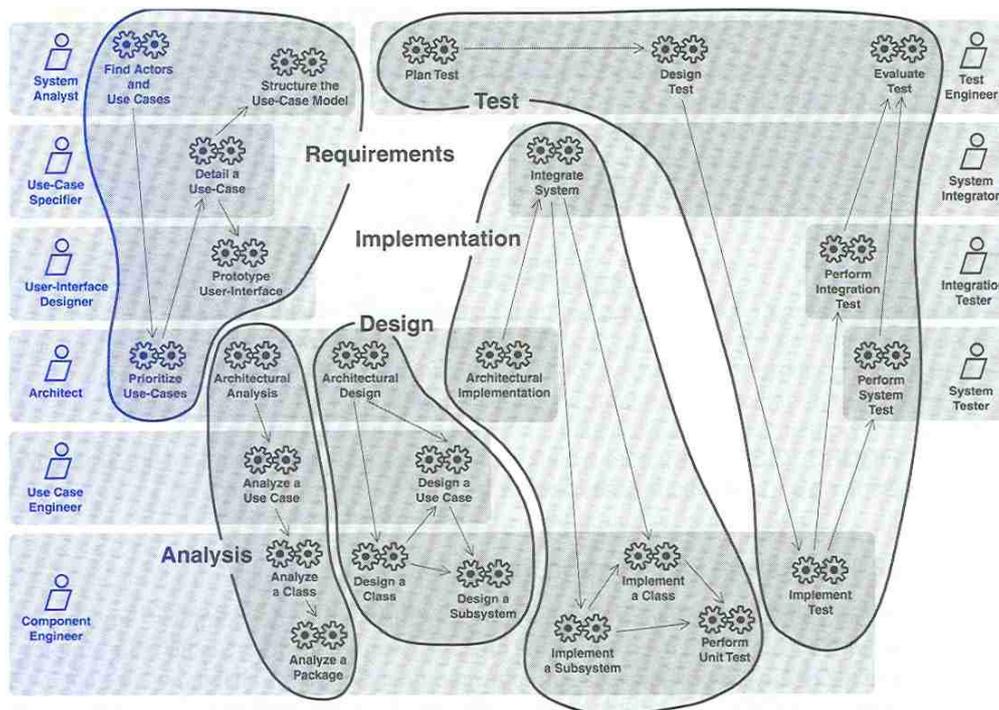


Figura 2.3 - Participação de alguns Papéis nos Fluxos do RUP

System Analyst

É responsável por todo o conjunto de requisitos (funcionais e não funcionais) que são modelados como casos de uso, incluindo todos os requisitos específicos a cada caso de uso. Para além disso, é responsável por delimitar o sistema, encontrar os actores e casos de uso e garantir que o modelo de casos de uso está completo e consistente. Para a consistência, o *System Analyst* pode utilizar um glossário para chegar a um consenso sobre os termos comuns, noções e conceitos quando os requisitos são capturados. Contudo, apesar de ser responsável pelo modelo de casos de uso e actores lá incluídos, não é responsável por cada caso de uso individual, atribuição do *Use Case Specifier*. Adicionalmente, o *System Analyst* é também o líder e coordenador da captura de requisitos. Apesar de apenas existir um *System Analyst* para cada sistema, na prática, este papel é suportado por uma equipa (em *workshops* ou eventos semelhantes) que inclui várias outras pessoas, que também trabalham como analistas.

Use-Case Specifier

Usualmente, as actividades de captura de requisitos não podem ser concluídas por apenas um indivíduo. Em vez disso, estes intervenientes assessoram o *System Analyst*, responsabilizando-se pela descrição detalhada de um ou mais casos de uso. Nesse sentido, estes intervenientes necessitam de trabalhar de perto com os utilizadores finais dos seus casos de uso.

User-Interface Designer

Dá forma visual à interface de utilizador. Contudo, embora não faça a implementação da mesma (que será efectuada por developers durante os fluxos de concepção e implementação), pode ser responsável por desenvolver protótipos para alguns casos de uso, usualmente um por cada actor. Logo, é apropriado deixar o *User-Interface Designer* adaptar a interface a um ou mais actores.

Architect

Durante o fluxos de análise, concepção e implementação, assegura a integridade dos respectivos modelos, garantindo que estão correctos, consistentes e legíveis. Em sistemas de grande dimensão e complexidade estas responsabilidades podem, após algumas iterações, requerer uma maior manutenção e o trabalho envolvido poderá tornar-se rotineiro. Nesses casos, o *Architect* poderá delegar estas funções noutro interveniente, possivelmente num *Component Engineer* de alto nível. Contudo, o *Architect* continuará responsável pelo que for arquitecturalmente significativo – a descrição arquitectural – enquanto o outro interveniente será o principal responsável pelos modelos de análise, concepção e implementação, que necessitam de cumprir a descrição arquitectural. Os modelos estão correctos quando concretizam (apenas) a funcionalidade descrita nos modelos de casos de uso e os (relevantes) requisitos adicionais. Adicionalmente, responsabiliza-se pela arquitectura dos modelos de análise, concepção e implementação, ou seja, pela existência das suas partes arquitecturalmente significativas, de acordo com a visão arquitectural do modelo (recorde-se que esta visão é parte da descrição arquitectural do sistema). Finalmente, um resultado importante da implementação é o mapeamento dos componentes executáveis para os nodos, processo pelo qual o *Architect* é responsável. Contudo, convém referir que o *Architect* não é responsável pelo desenvolvimento contínuo e pela manutenção dos vários artefactos contidos nos modelos de análise e de concepção (a cargo do *Use-Case Engineer*), nem pelo desenvolvimento contínuo e pela manutenção dos vários artefactos contidos no modelo de implementação (da responsabilidade do *Component Engineer*).

Use-Case Engineer

É responsável pela integridade da concepção de uma ou mais concretizações de casos de uso, assegurando que cumprem integralmente os respectivos requisitos. Uma concretização de cada caso de uso deve descrever correctamente (e apenas) o comportamento correspondente ao caso de uso respectivo (análise no modelo de análise e comportamento do

caso de uso respectivo no modelo de caso de uso). Nesse sentido, é necessário assegurar que todas as descrições textuais e diagramas que descrevem a concretização do caso de uso são legíveis e servem o seu propósito. Note-se que o *Use-Case Engineer* não é responsável pela concepção das classes, sub-sistemas, interfaces e relações de análise empregues na concretização do caso de uso, funções a desempenhar pelo *Component Engineer*. Contudo, é responsável pela concepção das concretizações do caso de uso já que, ao tratar da análise e concepção, assegura uma transição natural.

Component Engineer

Define as responsabilidades, atributos, relações e requisitos especiais de uma ou mais classes de análise e as operações, métodos, atributos, relações e requisitos de uma ou mais classes de concepção, assegurando que cada uma delas cumpre os requisitos impostos pelas concretizações de casos de uso em que participa. Para além disso, mantém o código fonte de um ou vários componentes, assegurando que cada componente implementa a funcionalidade correcta (ou seja, a que se encontra especificada na classe de concepção). Compete-lhe também manter a integridade de um ou mais pacotes de análise, garantindo que os seus conteúdos (por exemplo, classes e suas relações) estão correctos e que as suas dependências em relação a outros pacotes de análise são correctas e mínimas. Por outro lado, deve implementar componentes de teste que automatizem alguns dos procedimentos de teste (já que nem todos podem ser automatizados), dado que a criação destes componentes pode requerer substanciais competências de programação (de que o *Component Engineer* também necessitará durante o fluxo de implementação). Usualmente, é apropriado permitir que o *Component Engineer* que é responsável por uma pacote de análise o seja também pelas classes de análise que contém. Para além disso, quando existe um mapeamento directo entre um pacote de análise e os correspondentes sub-sistemas de concepção, o mesmo *Component Engineer* deverá ser também responsável por esses sub-sistemas, de modo a usar o conhecimento adquirido durante a análise efectuada ao desenhar e implementar o pacote de análise. Caso não exista esse mapeamento directo, outros *Component Engineers* poderão ser envolvidos na concepção e implementação no pacote de análise. Adicionalmente, deve zelar por manter a integridade de um ou mais sub-sistemas, o que inclui assegurar que os seus conteúdos (por exemplo, classes e suas relações) estão correctas, que as suas dependências em relação a outros sub-sistemas e/ou interfaces estão correctas e minimizadas, e que concretizam correctamente as interfaces que proporcionam. É frequente permitir que o *Component Engineer* responsável por um sistema também o seja pelos elementos do modelo

que contém. Para além disso, para atingir um desenvolvimento suave, é natural que os artefactos do modelo de concepção (por exemplo, as classes e sub-sistemas) sejam implementados durante do fluxo de desenvolvimento pelo mesmo *Component Engineer*. Por último, compete-lhe também a integridade de um ou mais sub-sistemas de implementação. Dado que entre estes e os sub-sistemas concebidos existe uma correspondência um-para-um, a maior parte das alterações a estes sub-sistemas são geridas durante a fase de concepção. Contudo, o *Component Engineer* necessita de garantir que os conteúdos (componentes, etc.) dos sub-sistemas de implementação estão correctos, que as suas dependências em relação a outros sub-sistemas e/ou interfaces estão correctas e que implementam correctamente as interfaces que proporcionam. É frequente que o *Component Engineer* responsável por um sub-sistema também o seja pelos elementos (componentes, etc.) contidos no seu modelo. Para além disso, para atingir um desenvolvimento suave, é natural que um *Component Engineer* seja responsável por um sub-sistema e respectivo conteúdo ao longo dos fluxos de concepção e implementação. Nesses casos, ele trataria de desenhar e implementar as classes que se encontrassem à sua responsabilidade.

System Integrator

A integração do sistema está para além do âmbito individual de cada *Component Engineer*, cabendo ao *System Integrator*. Tal inclui o planeamento da sequência de versões requeridas em cada iteração e a integração de cada versão assim que as suas partes estejam implementadas. Este planeamento resulta no Plano de Integração de Versões (*Integration Build Plan*).

Test Designer

É responsável pela integridade do modelo de testes, assegurando que este cumpre os seus objectivos. Planeiam os testes, o que significa que decidem quais são os objectivos de cada teste e a sua calendarização. Adicionalmente, seleccionam e descrevem os casos de teste e os respectivos procedimentos de teste necessários, para além de serem responsáveis por avaliar os testes de integração e sistema após estes terem sido executados. Note-se que os *Test Designers* não executam os testes (quem o faz são os *Integration Testers* e os *System Testers*), concentrando-se em vez disso na sua preparação e avaliação.

Integration Tester

São responsáveis por realizar os testes de integração necessários para cada versão produzida no fluxo de implementação. Estes testes são executados de modo a verificar que os

componentes integrados numa versão funcionam correctamente em conjunto. Por isso, os testes de integração são usualmente derivados de casos de teste que especificam como testar as concretizações da concepção dos casos de uso. Os resultados dos testes de integração são defeitos emanados pelo *Integration Tester*. Note-se que um *Integration Tester* testa o resultado (ou seja, uma versão) criada por um *System Integrator* durante o fluxo de implementação. Consequentemente, em alguns projectos, estes papéis são acumulados por um ou mais indivíduos, de modo minimizar a necessidade de replicar conhecimento comum em colaboradores distintos.

System Tester

É responsável por realizar os testes de sistema necessários para uma versão de um resultado (executável) de uma iteração completa. Estes testes são principalmente executados para verificar as interacções entre os actores e o sistema. Por isso, os testes de sistema são usualmente derivados dos casos de teste que especificam como testar os casos de uso, mas outros tipos de testes também são aplicáveis ao sistema como um todo. Os resultados dos testes de sistema são defeitos emanados pelo *System Tester*. Dada a natureza dos testes de sistema, os indivíduos que actuam como *System Testers* não necessitam de saber muito sobre o funcionamento interno do sistema.

Pelo exposto, [Jacobson, *et al.*, 1999] peca por não possuir, nem sequer em anexo, uma clara caracterização de todos os intervenientes, essencial não só para compreender a abrangência e profundidade da sua intervenção, como também para permitir, a cada titular de um desses papéis, saber exactamente o que dele se espera. Porém, e apesar de a Figura 2.3 ser útil como visão sumarizada e enquadradora, fica muito aquém do que seria necessário para operacionalizar eficazmente a metodologia, dada a miríade de tarefas que usualmente recaem sobre cada participante num processo de desenvolvimento de software. Estabelecendo uma analogia com o contexto militar, podemos considerar esta figura como um diagrama que dispõe as tropas no terreno e que, apesar de os orientar em relação ao seu posicionamento, não elimina a necessidade de uma definição pormenorizada sobre a forma como cada companhia deverá agir durante o combate (alvos a atingir, quem deverá avançar primeiro, qual o flanco de ataque, etc.). Todavia, convém não esquecer que, para além da sua promoção pública⁴⁴, o RUP é um produto⁴⁵ comercial da *Rational*, pelo que é bastante provável que esta

⁴⁴ De que a obra referida é exemplo.

omissão seja tudo menos accidental. Por tudo isto, conclui-se que [Jacobson, *et al.*, 1999] não constitui uma boa base de partida para a identificação do elenco processual nem para o estabelecimento de um referencial de acção claro para cada um dos seus elementos. Pelo exposto, impôs-se a necessidade de encontrar uma outra fonte de informação que permitisse clarificar esta questão, tendo a escolha recaído sobre o Tutorial HTML do RUP. Contudo, dado que este tutorial é parte integrante do pacote comercializado pela *Rational*, num primeiro momento apenas foi possível obter uma versão datada de 2001 [Rational, 2001], já sujeita a uma considerável desactualização em virtude das frequentes revisões a que a metodologia é periodicamente sujeita. Não obstante, a consulta desta referência revelou-se bastante mais proveitosa do que [Jacobson, *et al.*, 1999], permitindo o acesso fácil e rápido⁴⁶ a informação sobre o elenco processual e segundo a qual:

- *Um Role é uma definição abstracta de um conjunto de actividades desenvolvidas e de artefactos pelos quais é responsável;*
- *A cada Role está associado um conjunto de actividades coerente e da sua competência. Estas actividades encontram-se tão associadas entre si que serão melhor executadas por um mesmo indivíduo;*
- *Um elemento da equipa de projecto pode desempenhar vários Roles;*
- *Os Roles não são indivíduos; em vez disso, descrevem como os indivíduos se devem comportar e que responsabilidades devem ter.*

in [Jacobson, *et al.*, 1999]

Todavia, esta caracterização parece um pouco contraditória com a descrição do processo apresentada em [Jacobson, *et al.*, 1999] que, como referido na secção anterior, define o elenco processual como um conjunto de *Workers* e não de *Roles*. Aliás, a obra é clara no que concerne à intencionalidade dessa diferença, alegando os seguintes motivos para evitar a utilização do termo *Role*:

- Tem um significado preciso e diferente em UML;
- É necessário utilizar o termo *Role* para identificar os papéis desempenhados por um *Worker*, já que este pode assumir um papel diferente em cada fluxo de trabalho em que participa.

⁴⁵ Composto por documentação, modelos para documentos, ferramentas, etc.

⁴⁶ Seguindo a opção *Roles*.

Assim, surpreende a incongruência terminológica, principalmente por se tratarem ambas de obras oficiais sobre a metodologia, e não da autoria de terceiros que não se encontrem envolvidos na concepção e divulgação.

Terminologia à parte, o tutorial RUP subdivide o elenco processual nos seguintes grupos:

- ***Analyst***: Abrange os *Roles* com maior envolvimento no processo de levantamento de requisitos;
- ***Developer***: Abrange os *Roles* com maior envolvimento na concepção e implementação de software;
- ***Tester***: Abrange os *Roles* com maior envolvimento no processo de teste de software;
- ***Manager***: Abrange os *Roles* com maior envolvimento na gestão e configuração do processo de engenharia de software;
- ***Other Roles***: Abrange todos os restantes *Roles*, nomeadamente os envolvidos no suporte ao processo.

Assim, e de acordo com esta classificação, a Tabela 2.1 apresenta os papéis definidos pelo tutorial.

Tabela 2.1 – Papéis definidos em [Rational, 2001]

(a) Grupos Analyst e Developer

Papel	Descrição
Analyst	
<i>Business Designer</i>	Detalha a especificação de uma parte da organização e especifica o fluxo dos casos de uso de negócio em termos de <i>workers</i> e entidades, para além de definir as responsabilidades, operações, atributos e relações destes.
<i>Business-Model Reviewer</i>	Participa na revisão formal dos modelos de casos de uso de negócio e de objectos de negócio.
<i>Business-Process Analyst</i>	Lidera e coordena a modelação dos casos de uso de negócio, demarcando e delimitando a organização a modelar (por exemplo, estabelecendo quais os actores e casos de uso de negócio existem e como interagem entre si).
<i>Requirements Reviewer</i>	Planeia e conduz a revisão formal do modelo de casos de uso.
<i>Requirements Specifier</i>	Detalha a especificação de uma parte da funcionalidade do sistema, descrevendo os requisitos de um ou mais casos de uso e outros requisitos para o software de suporte. Também pode ser responsável pela manutenção da integridade de um determinado pacote de casos de uso, sendo recomendável nesses casos que também o seja sobre os casos de uso e actores que o compõem.
<i>System Analyst</i>	Lidera e coordena a enumeração dos requisitos e a modelação dos casos de uso, delimitando o sistema e sublinhando a sua funcionalidade (por exemplo, estabelecendo quais os actores e casos de uso existem e como interagem entre si).
<i>User-Interface Designer</i>	Lidera e coordena a prototipagem e concepção das interfaces de utilizador, através da: <ul style="list-style-type: none"> ▪ captura de requisitos para a interface de utilizador, nomeadamente no que concerne à sua usabilidade; ▪ construção de protótipos das interfaces de utilizador; ▪ envolvendo outros <i>stakeholders</i> da interface de utilizador nas revisões de usabilidade e nas sessões de teste; ▪ revisão da implementação final das interfaces de utilizador, reportando as conclusões retiradas.
Developer	
<i>Architecture Reviewer</i>	Planeia e conduz a revisão formal da arquitectura de software inerente ao projecto.
<i>Capsule Designer</i>	Procura assegurar que o sistema pode responder a eventos de forma atempada e de acordo com os requisitos de concorrência.
<i>Code Reviewer</i>	Assegura a qualidade do código fonte e planeia a realização de revisões ao mesmo, para além de se responsabilizar pelo reporte das conclusões resultantes desse processo.
<i>Database Designer</i>	Define as tabelas, índices, <i>views</i> , <i>constraints</i> , <i>triggers</i> , <i>stored procedures</i> e outros parâmetros específicos ao motor de base de dados relacionados com o armazenamento, recuperação e remoção de objectos persistentes.
<i>Designer</i>	Define as responsabilidades, operações, atributos e relações de uma ou mais classes e determina como estas devem ser ajustadas ao ambiente de implementação. Adicionalmente, pode ser responsável por um ou mais pacotes ou sub-sistemas, incluindo quaisquer classes que pertençam aos referidos pacotes ou sub-sistemas.
<i>Design Reviewer</i>	Planeia e conduz a revisão formal do modelo de concepção.
<i>Implementer</i>	É responsável pelo desenvolvimento e teste de componentes, de acordo com as normas adoptadas no projecto para a integração em sub-sistemas de maior dimensão. Quando componentes de teste, como <i>drivers</i> ou <i>stubs</i> , têm de ser criados para suportar os testes, é também responsável pelo seu desenvolvimento e teste (e dos sub-sistemas correspondentes).
<i>Integrator</i>	É responsável por combinar os componentes entregues pelos <i>Implementers</i> no sentido de produzir uma <i>build</i> . Adicionalmente, é também responsável pelo planeamento desse processo de integração, que decorre ao nível do sistema e sub-sistema, cada um com um espaço de integração próprio.
<i>Software Architect</i>	Lidera e coordena as actividades e artefactos técnicos ao longo do projecto. Adicionalmente, estabelece a estrutura geral de cada visão arquitectural: a decomposição da visão, o agrupamento de elementos e as interfaces entre os maiores agrupamentos. Logo, comparativamente com os restantes papéis, a visão do arquitecto deve ser mais abrangente do que profunda.

Tabela 2.1 – Papéis definidos em [Rational, 2001]

(b) Grupos Manager, Tester e Other

Papel	Descrição
Manager	
<i>Change Control Manager</i>	Supervisiona o processo de controlo de alterações. Usualmente, este papel é desempenhado por um Comité de Controlo da Mudança, composto por representantes de todas as partes interessadas, incluindo clientes, <i>developers</i> e utilizadores, etc. Em projectos de pequena dimensão, uma única pessoa, como o <i>Project Manager</i> ou o <i>Software Architect</i> , pode desempenhar este papel.
<i>Configuration Manager</i>	É responsável por disponibilizar a infra-estrutura de gestão de configurações para a equipa de desenvolvimento, que suporta a sua actividade de modo a que os <i>developers</i> e integradores possuam os espaços de trabalho apropriados às suas acções de implementação, teste e disponibilização dos artefactos que produzem. Assegura que o ambiente de gestão de configurações facilita as actividades de revisão, alteração e gestão de defeitos do produto. Adicionalmente, é também responsável por escrever o plano de gestão de configurações e por reportar estatísticas sobre o progresso de pedidos de alteração.
<i>Deployment Manager</i>	É responsável por planear a transição do produto para a comunidade de utilizadores, documentando adequadamente o processo.
<i>Process Engineer</i>	É responsável pelo processo de desenvolvimento de software em si, incluindo a sua configuração antes de se iniciar e a sua melhoria contínua ao longo do esforço de desenvolvimento.
<i>Project Manager</i>	Aloca recursos, define prioridades, coordena interacções com clientes e utilizadores e esforça-se por manter a equipa de projecto focada nos objectivos correctos. Adicionalmente, estabelece um conjunto de práticas no sentido de assegurar a integridade e qualidade dos artefactos gerados pelo projecto.
<i>Project Reviewer</i>	É responsável por analisar os artefactos de planeamento do projecto e de avaliação do desenrolar do mesmo em determinados pontos chave do seu ciclo de vida. Estes são eventos de revisão significativos porque marcam pontos nos quais o projecto pode ser cancelado se o planeamento for inadequado ou se o progresso for inaceitavelmente mau.
Tester	
<i>Test Designer</i>	É o interveniente com maior envolvimento no esforço de testes, sendo responsável pelo seu planeamento, concepção, implementação e avaliação, incluindo a: <ul style="list-style-type: none"> ▪ geração do plano e modelo de testes; ▪ implementação de procedimentos de teste; ▪ avaliação da abrangência dos testes, dos seus resultados e da sua eficácia; ▪ geração de um sumário de avaliação do esforço de testes.
<i>Tester</i>	É responsável por preparar e executar um ou mais testes, por avaliar a sua execução e por recuperar de erros que possam ter lugar durante os mesmos.
Other	
<i>Any Role</i>	Qualquer outro papel pode, com os necessários privilégios de acesso, efectuar o <i>check-in</i> e <i>check-out</i> dos artefactos do projecto para proceder à sua manutenção no sistema de controlo de configurações. Adicionalmente, também pode submeter pedidos de alteração e actualizar os pedidos de alteração da sua responsabilidade.
<i>Course Developer</i>	Desenvolve material de formação (slides, exemplos, tutoriais, etc.) para ensinar os utilizadores a compreender e usar o produto.
<i>Graphic Artist</i>	Desenvolve o trabalho gráfico inerente ao pacote de distribuição do produto.
<i>Stakeholder</i>	Qualquer elemento que seja materialmente afectado pelo resultado do projecto. Este é outro papel genérico que cobre, dependendo do contexto do projecto, cliente, utilizadores finais, adjudicante, accionista, etc.
<i>System Administrator</i>	Mantém o ambiente de desenvolvimento (quer hardware quer software) e é responsável pela administração do sistema, pela realização de <i>backups</i> regulares, etc.
<i>Technical Writer</i>	Produz material de suporte a utilizadores finais, como manuais de utilizador, textos de ajuda, notas da distribuição, etc.
<i>Tool Specialist</i>	É responsável pelas ferramentas de suporte ao projecto, incluindo a selecção e aquisição das mesmas. Adicionalmente, prepara e configura as ferramentas e verifica que estas se encontram funcionais.

Encontra-se assim apresentado, de uma forma sistematizada e integrada, o elenco que, segundo [Rational, 2001], suporta a operacionalização do RUP. Contudo, também esta definição encerra algumas surpresas, não tanto pelo que inclui mas sim pelo que parece faltar, como por exemplo:

- Quatro dos dez papéis referenciados em [Jacobson, *et al.*, 1999] não constam desta lista, nomeadamente: *Component Engineer*, *Integration Tester*, *Use-Case Engineer* e *Use-Case Specifier*;
- Três dos dez papéis referenciados em [Jacobson, *et al.*, 1999] constam de [Rational, 2001] mas com uma designação ligeiramente diferente:

Tabela 2.2 – Diferenças de Designação de Papéis entre [Jacobson, *et al.*, 1999] e [Rational, 2001]

Designação [Jacobson, <i>et al.</i> , 1999]	Designação [Rational, 2001]
<i>Architect</i>	<i>Software Architect</i>
<i>System Integrator</i>	<i>Integrator</i>
<i>System Tester</i>	<i>Tester</i>

Assim, as inconsistências descritas permitem concluir que [Rational, 2001] também não constitui uma referência estável para o elenco processual que se pretendia caracterizar ao longo desta secção. Por isso, e dado que um dos objectivos assumidos em [Kruchten, 2004] é exactamente proporcionar a cada interveniente no RUP a orientação de que necessita para cumprir a sua função, foi esta a obra eleita para prosseguir o processo de investigação. No que concerne à terminologia, [Kruchten, 2004] parece estar alinhada com [Rational, 2001], apelidando os papéis de *Roles* e não de *Workers*. Também se verificam semelhanças na categorização dos vários *Roles*, com a pequeníssima diferença de [Kruchten, 2004] enquadrar num grupo de *Production and Support* alguns elementos que em [Rational, 2001] pertencem ao grupo genérico *Other Roles*. Para além disso, o Apêndice A de [Kruchten, 2004] encerra um sumário dos trinta papéis que, segundo o autor, se encontram envolvidos no RUP. Contudo, em [Rational, 2001] encontram-se tipificados trinta e um papéis, o que indicia desde logo a existência de alguma diferença. Assim, e após uma análise mais detalhada, podem ser retiradas as seguintes conclusões:

- Dois dos trinta e um papéis referenciados em [Rational, 2001] não constam de [Kruchten, 2004], a saber: *Requirements Reviewer* e *Project Reviewer*;

- Tal como identificado na Tabela 2.3, há cinco papéis que não são referenciados em [Rational, 2001] mas que constam de [Kruchten, 2004];
- No [Rational, 2001], o papel *User-Interface Designer* é englobado no grupo *Analyst*, enquanto [Kruchten, 2004] o define como *Developer*;
- Embora seja suposto⁴⁷ que o Apêndice A de [Kruchten, 2004] esgote a totalidade do elenco processual, há vários papéis (identificados na Tabela 2.4) que, apesar de não constarem deste, são referenciados ao longo da obra;
- O papel *Reviewer* definido no Apêndice A de [Kruchten, 2004] possui uma dimensão genérica que não justifica a sua existência *de per se*, dada a existência de outros papéis da mesma natureza mas com um âmbito mais específico (por exemplo, o *Architecture Reviewer*, *Business Reviewer*, etc.). Para além disso, a definição do papel remete para um outro (*Technical Reviewer*) que não se encontra definido ou referenciado em mais lado nenhum;
- O papel *Business-Model Reviewer* definido em [Rational, 2001] corresponde ao papel *Business Reviewer* descrito em [Kruchten, 2004];
- Quatro dos dez papéis referenciados em [Jacobson, et al., 1999] não constam de [Kruchten, 2004], a saber: *Component Engineer*, *Integration Tester*, *Use-Case Engineer* e *Use-Case Specifier*.

Considerando que Philippe Kruchten é *Lead Architect* do RUP, é um pouco surpreendente que esta obra venha, à semelhança das anteriores, confundir ainda mais a composição do elenco processual, na sequência das inconsistências que acabam de ser relatadas. Decidiu-se então estender a investigação à documentação gerada pela ferramenta *Method Composer*⁴⁸ da *Rational* (ver [Rational, 2006]), no sentido esclarecer em definitivo a correcta composição do elenco processual. Ao analisá-la, pode concluir-se que o seu conteúdo se encontra bastante próximo do Apêndice A de [Kruchten, 2004], parecendo confirmar que é esse o caminho que a evolução da metodologia tem vindo a trilhar. Exemplos disso são a existência do grupo *Production and Support*, a existência dos papéis *Management*

⁴⁷ A julgar pelo texto introdutório do mesmo.

⁴⁸ Disponibilizada gratuitamente durante um período de experiência de 30 dias.

Reviewer, *Test Manager*, *Test Analyst*, *Reviewer* e *Review Coordinator*, a inexistência dos papéis de *Requirements Reviewer*, *Project Reviewer*, *Architecture Reviewer*, *Code Reviewer* e *Design Reviewer*, para além do facto do *User-Interface Designer* se encontrar classificado como *Developer* e não como *Analyst*. Contudo, em abono da verdade, convém também referir que foram detectadas ligeiras incongruências em relação às restantes referências, como sejam a inexistência de *Business-Model Reviewer* (existente em [Rational, 2001]) e de *Business Reviewer* (existente em [Kruchten, 2004]), para além da classificação do *System Administrator* como *Manager* e não como *Other* (como em [Rational, 2001]) nem *Production and Support* (como em [Kruchten, 2004]).

Tabela 2.3 – Papéis definidos em [Kruchten, 2004] mas ausentes em [Rational, 2001]

Papel	Descrição
<i>Management Reviewer</i>	É responsável pela avaliação do planeamento do projecto e dos principais artefactos produzidos em determinados momentos-chave do ciclo de vida do projecto.
<i>Test Manager</i>	É o responsável geral pelo sucesso do esforço de testes, promovendo a qualidade e a realização de testes. Para isso, necessita de desenvolver actividades de planeamento e gestão de recursos e de resolver os problemas que possam impedir o esforço de testes.
<i>Test Analyst</i>	É responsável por identificar e definir os testes necessários, monitorizar o progresso e resultados dos testes detalhados em cada ciclo de testes e avaliar a qualidade global experienciada em resultado das actividades de teste. Tipicamente, tem ainda a responsabilidade adicional de representar de forma adequada as necessidades dos <i>stakeholders</i> que não têm representação directa ou regular no projecto.
<i>Reviewer</i>	É um papel genérico, responsável por proporcionar comentários em tempo útil à equipa de projecto sobre os artefactos por eles produzidos (ver também <i>Management Reviewer</i> e <i>Technical Reviewer</i>).
<i>Review Coordinator</i>	É responsável por facilitar as revisões e inspecções formais, assegurando que elas ocorrem quando necessário e que são conduzidas de forma satisfatória e de acordo com os processos instituídos.

Pelo exposto, é difícil afirmar com total propriedade qual a completa e correcta composição do mesmo, o que, dada a subjectividade inerente, poderá não só dificultar a adopção da metodologia como também contribuir para a sua descredibilização.

Tabela 2.4 – Omissões do Apêndice A de [Kruchten, 2004]

Papel	Página da Referência	Descrição
<i>Architecture Reviewer</i>	176	Responsável por rever os artefactos-chave criados no âmbito da disciplina de <i>Analysis and Design</i> .
<i>Business Reviewer</i>	147	Responsável por rever os artefactos criados no âmbito da disciplina de <i>Business Modeling</i> .
<i>Code Reviewer</i>	193	Responsável por inspecionar o código fonte gerado e aferir a sua qualidade e conformidade com as respectivas políticas definidas para o projecto.
<i>Design Reviewer</i>	176	Responsável por rever os artefactos-chave criados no âmbito da disciplina de <i>Analysis and Design</i> .

Assim, e dado não ter sido possível encontrar uma fonte de informação inequívoca e completa sobre esta matéria, é pertinente coligir os dados que foram sendo recolhidos no sentido de reconstituir uma composição verosímil do elenco processual, que possa servir de

base ao trabalho a desenvolver no capítulo seguinte. Assim, e de modo a não alongar ainda mais esta já de si extensa secção, são apresentados no Anexo I o conjunto de papéis que, como corolário dos comentários tecidos nesta secção, se afiguram como mais relevantes na aplicação do RUP.

2.4. Aplicações do RUP

As secções anteriores proporcionam uma introdução aos principais conceitos em que o RUP se encontra alicerçado, contribuindo assim para uma maior tomada de consciência sobre o mesmo por parte das organizações que ponderem a sua adopção. Contudo, para além de dar a conhecer a metodologia, considerou-se igualmente importante proceder a uma análise de artigos sobre esta, especialmente se possibilitassem a sua comparação com abordagens alternativas ou permitissem retirar ensinamentos úteis sobre a sua aplicação em contextos reais de interesse para uma organização PME. Assim, e dado não ser objectivo desta dissertação efectuar uma monografia da obra publicada sobre esta temática, decidiu-se limitar a cinco o número de artigos analisados, por se considerarem suficientes para explorar diferentes perspectivas e vivências da metodologia.

Falando um pouco sobre o leque de artigos seleccionados, [Hirsch, 2002] transmite uma perspectiva muito pragmática sobre a forma como o RUP poderá ser configurado de modo a “agilizar” a sua aplicação (obviamente sem abdicar de nenhum componente processual considerado essencial) e, assim, provar a possibilidade da sua adopção bem sucedida em contextos PME (de que a organização a que o artigo se refere é exemplo). Nesse sentido, o caminho seguido passou pela simplificação significativa da lista de artefactos a produzir, na sequência de uma análise custo/benefício de cada um dos artefactos previstos pela metodologia. Seguindo uma lógica completamente diferente, [Fernandes, Duarte, 2005] apresenta uma abordagem de configuração do RUP orientada essencialmente para organizações que desenvolvam software de uma forma orientada aos processos, o que poderá ser apropriado para entidades de pequena dimensão em que não se justifique a existência de uma estrutura funcional. Adicionalmente, para além de ser (à semelhança do anterior) praticamente omissa no que respeita ao elenco processual que operacionaliza a metodologia, são apresentados alguns resultados da aplicação da mesma a um contexto real, dando especial destaque ao conjunto de artefactos de modelação de negócio cuja produção é considerada essencial. Este artigo alerta ainda para uma eventual necessidade de reestruturação interna em organizações que adoptem o RUP, de modo a suprir as suas dificuldades na composição do elenco processual envolvido. Em [Duarte, *et al.*, 2006], não só se dá continuidade a esta linha

de orientação, analisando ainda com maior profundidade os artefactos de modelação de negócio, como também se apresenta uma forma de configurar a metodologia que permita incorporar a melhoria processual para, dessa forma, permitir às organizações adoptantes obter uma melhor classificação na escala CMM.

Por outro lado, e no sentido de identificar as forças e fraquezas do RUP quando comparado com abordagens alternativas, foram também examinados alguns artigos mais críticos em relação ao mesmo. Segundo [Hesse, 2003], o RUP é um pouco confuso e demasiadamente complexo e sofisticado para ser passível de uma aplicação prática bem sucedida, motivos que levam o autor a promover as virtudes do EOS. Contudo, e mesmo sem abordar em grande detalhe as questões relacionadas com o elenco processual, alega-se que o RUP não enquadra da melhor forma os vários intervenientes e não envolve suficientemente os utilizadores finais durante a fase de transição. O OPEN é uma outra metodologia alternativa, comparada qualitativamente com o RUP em [Henderson-Sellers, *et al.*, 2001], nomeadamente no que concerne aos conceitos subjacentes a ambas, evidenciados nos respectivos meta-modelos. Ainda de acordo com este artigo, o RUP é considerado omissivo sobre a forma mais adequada de gerir os recursos humanos envolvidos na sua utilização.

Pelo exposto, a análise (a detalhar nas sub-secções que se seguem) dos artigos seleccionados justifica a pertinência dos objectivos traçados para esta dissertação, ao expor a carência de tratamento das questões relacionadas com a configuração do elenco processual do RUP.

“*Making RUP Agile*” [Hirsch, 2002]

Em termos gerais, e como o próprio título indicia, este artigo denota a preocupação de operacionalizar as recomendações compreendidas no RUP de uma forma suficientemente pragmática (ou “ágil”, nas palavras do autor) para permitir a sua aplicação a contextos de reduzida complexidade. Assim, este artigo relata os resultados obtidos com a adopção do RUP para o desenvolvimento de dois projectos de pequena dimensão, envolvendo entre três a cinco programadores e com uma duração de entre seis a nove meses, tendo ambos sido bem sucedidos. Segundo o autor, os principais desafios que enfrentaram foram o curto período para a implementação (9 meses) e o facto de os requisitos iniciais serem inicialmente muito vagos.

Contextualizando um pouco, a Zühlke Engineering AG é uma empresa suíça de engenharia que actualmente emprega cerca de duzentos engenheiros e cuja dimensão das

equipas de projecto tipicamente varia entre sete e vinte elementos (para projectos de sistemas integrados) ou entre três a dez elementos (para projectos de desenvolvimento de software). Tal como a maioria das organizações desta área, o processo de desenvolvimento empregue pela Zühlke Engineering foi sendo adaptado ao longo do tempo no sentido de colmatar as dificuldades com que se iam confrontando, evoluindo da análise estruturada para a definição de um modelo próprio (do tipo “*Cascata*”). Contudo, por volta de 1998 começou a tornar-se claro para eles⁴⁹ que este já não era satisfatório pelo que, após um levantamento sobre os processos iterativos existentes, decidiram adoptar o RUP tendo por base o seguinte argumentário:

- Era (à data) o único processo iterativo bem documentado que conseguiram encontrar;
- Era muito completo, poupando-lhes o trabalho de criar modelos para os artefactos, de definir linhas de orientação, etc.;
- Dava a sensação de ter sido criado por indivíduos com uma grande conhecimento de causa e experiência prática, não se resumindo a um conjunto de ideias não validadas;
- A documentação do RUP encontrava-se disponível em formato electrónico, facilitando a sua consulta⁵⁰ por todos os intervenientes;
- Já utilizavam UML nas suas actividades de modelação.

Tomada que estava a decisão de adoptar a metodologia, seguiu-se um processo de análise mais detalhada da mesma, no sentido de definir qual a abrangência das recomendações a incorporar. Em termos gerais, as decisões tomadas foram as seguintes:

- *Artefactos*: De acordo com o próprio RUP, só devem ser produzidos artefactos que aportem valor ao projecto ou aos seus envolvidos, pelo que a aplicação da metodologia a pequenos projectos carece de um cuidadoso processo de selecção. Por isso, e como será analisado em maior detalhe mais adiante, a Zühlke Engineering decidiu reduzir consideravelmente o lote de artefactos a utilizar;

⁴⁹ Por motivos que o artigo não especifica.

⁵⁰ Optaram por não alterar a documentação do RUP para não terem de o voltar a fazer para cada versão nova que fosse publicada. Em vez disso, criaram um manual interno que detalha as adaptações que fizeram ao processo.

- *Actividades*: Não alteraram quaisquer actividades e decidiram não as utilizar no planeamento detalhados das iterações (ao contrário do que o RUP sugere), já que entenderam que os seus recursos humanos eram suficientemente experientes na modelação e outras técnicas empregues no RUP. Para além disso, as actividades e as suas agregações (apelidadas de *workflow details*) eram bastante detalhadas⁵¹, pelo consideraram não fazer sentido planear com tanto detalhe o trabalho levado a cabo por intervenientes experientes. Em vez disso, criaram um guia de referência com a descrição das várias actividades;
- *Papéis*: Decidiram não fazer a atribuição formal de papéis aos intervenientes, tendo apenas utilizado o elenco processual definido no RUP para verificar se a equipa possuía todas as competências necessárias;
- *Planeamento do Projecto*: De acordo com o sugerido pelo RUP, decidiram manter dois níveis de planos: um de alto nível, que lista as datas de início e fim de cada iteração/fase e, um mais detalhado para cada iteração, preparado alguns dias antes do seu início e modificado durante a mesma se necessário for. Para além disso, em vez deste planeamento ser constituído pela lista de tarefas a realizar, era composto pela lista de resultados a atingir⁵², no sentido de manter os intervenientes constantemente focados no que realmente é importante assegurar;
- *Fases*: Decidiram adoptar as quatro fases do RUP e as respectivas *milestones* sem qualquer modificação;
- *Iterações*: No sentido de facilitar o controlo, utilizaram iterações de um mês (iniciadas no primeiro dia do mesmo), em que cada uma resulta numa versão da(s) aplicação(ões) testada(s) e devidamente entregue(s) ao cliente. Assim, consideram que uma iteração é um período de planeamento, com uma lista detalhada de objectivos a atingir durante esse período, não tendo nada a ver com a frequência de criação de versões⁵³ da(s) aplicação(ões). Para além disso, apesar de aceitarem incorporar na iteração actual os pequenos pedidos de alteração, os de maior dimensão ficavam adiados para futuras iterações, dado implicarem um

⁵¹ Por exemplo, “descrever caso de uso”, “escrever caso de teste para classe”, etc.

⁵² Tipicamente, cada iteração tinha cerca de 12 funcionalidades a implementar e 10 pedidos de alteração e correcções de *bugs*.

⁵³ Que, tal como proposto pelo RUP, eram criadas diariamente.

replaneamento global do projecto;

- *Controlo do Projecto*: Utilizaram reuniões semanais com toda a equipa como a principal forma de controlo do projecto. Cada resultado a atingir era da responsabilidade de uma só pessoa, para que não houvessem dúvidas sobre quem devia fazer o quê. No início de cada semana, cada elemento da equipa estima o tempo de que ainda necessita para atingir os seus objectivos nessa iteração, no sentido de detectar precocemente eventuais desvios do plano. Contudo, quando tal acontece não há lugar a prolongamento da iteração, pelo que a funcionalidade em causa passa para a iteração seguinte⁵⁴.

Detalhando um pouco mais a abordagem seguida no que concerne aos artefactos, o processo de selecção tomou por base as seguintes questões: “*qual o valor que este artefacto representa para o cliente ?*” e “*quais as consequências de não dispormos deste artefacto ?*”. Como resultado desse processo, a Zühlke Engineering decidiu-se pelo conjunto identificado na Tabela 2.5.

*Tabela 2.5 – Lista de Artefactos utilizados pela Zühlke Engineering AG
(a) Requisitos, Concepção e Análise, Implementação e Testes*

Artefacto	Comentário	Formato
<i>Requisitos</i>		
<i>Documento de Visão</i>	Descreve os objectivos do projecto, tipos importantes de utilizadores e todas as funcionalidades do produto, que se encontram univocamente identificadas para que possam ser facilmente referenciadas pelos outros artefactos.	Documento textual com cerca de 15 páginas.
<i>Modelo de Casos de Uso</i>	Descreve todos os casos de uso do sistema num único documento. Os casos de uso foram utilizados para melhor compreender o sistema, mas não para planeamento do projecto. As versões iniciais do documento incluíam maquetes das interfaces gráficas, que em versões posteriores foram substituídas por imagens das mesmas, entretanto desenvolvidas.	Documento textual com cerca de 20 páginas.
<i>Concepção e Análise</i>		
<i>Arquitectura de Software</i>	Descreve o funcionamento a alto nível do sistema. Contém alguns diagramas UML juntamente com uma breve descrição dos mecanismos mais importantes.	Documento textual com cerca de 12 páginas.
<i>Modelo de Concepção</i>	Modelo detalhado do sistema, em termos de objectos e seu agrupamento em <i>packages</i> . Utilizaram a ferramenta CASE Together/J para efectuar a modelação dos objectos.	Projecto Together/J.
<i>Implementação</i>		
<i>Modelo de Implementação</i>	Na terminologia RUP, o modelo de implementação consiste numa colecção de artefactos necessários para construir o sistema, ou seja, código fonte, <i>makefiles</i> , ficheiros de configuração, etc.	Código Fonte em Java, <i>makefiles</i> , ficheiros de Jbuilder.
<i>Testes</i>		
<i>Lista de Defeitos</i>	Uma lista de todos os defeitos (resolvidos ou por resolver), incluindo uma breve descrição de cada um.	Documento textual com cerca de 3 páginas.

⁵⁴ Segundo a Zühlke Engineering esta política deve-se ao facto de quer eles quer os seus clientes preferirem ter uma versão pronta no prazo previsto mas com menos funcionalidades, em detrimento de uma versão completa mas fora de tempo.

Da aplicação do RUP, a Zühlke Engineering conseguiu retirar vários ensinamentos que poderão ser de utilidade a organizações com contextos competitivos similares. Em primeiro lugar, consideram que a adopção de uma abordagem iterativa foi crucial para a conclusão bem sucedida⁵⁵ (dentro do prazo e orçamento) de ambos os projectos, dada a usual volatilidade dos requisitos no decurso do mesmo. Contudo, tal só pôde ser conseguido fruto de um forte envolvimento dos clientes no planeamento e monitorização do projecto, nomeadamente através de uma decisão conjunta sobre o conteúdo⁵⁶ de cada iteração.

Tabela 2.5 – Lista de Artefactos utilizados pela Zühlke Engineering AG

(b) Deployment, Gestão da Configuração e Mudança, Gestão de Projecto e Contextualização

Artefacto	Comentário	Formato
<i>Deployment</i>		
<i>Notas da Distribuição</i>	Escritas para cada entrega feita ao cliente.	Documento textual com cerca de 2 páginas.
<i>Artefactos de Instalação</i>	Ficheiros executáveis, de configuração, procedimentos de instalação, documentação, etc. necessários para proceder à instalação do sistema. Em cada distribuição, um conjunto de artefactos de instalação é criada e entregue ao cliente.	Ficheiro ZIP.
<i>Gestão da Configuração e Mudança</i>		
<i>Plano de Gestão de Configurações</i>	Describe as políticas relativas ao controlo de versões, gestão de <i>releases</i> e de pedidos de alteração. Utilizaram CVS para gestão de versões e de alterações. Todos os artefactos do projecto foram sujeitos a controlo de versões.	Documento textual com cerca de 8 páginas.
<i>Lista de Pedidos de Alteração</i>	Uma lista de todos os pedidos de alteração (resolvidos ou por resolver), incluindo uma breve descrição de cada um.	Documento textual com cerca de 4 páginas.
<i>Gestão de Projecto</i>		
<i>Plano de Desenvolvimento</i>	Um plano de alto nível com uma lista de todas as iterações planeadas. Para cada iteração deve conter a identificação dos principais objectivos a atingir, datas de início e fim. Este plano é actualizado em cada iteração.	Documento textual com cerca de 8 páginas.
<i>Plano da Iteração</i>	Um plano detalhado para cada iteração, identificando que funcionalidades, pedidos de alteração, correcções de <i>bugs</i> e restantes artefactos, bem como de quem está encarregue de os produzir.	Documento textual com cerca de 4-6 páginas (por iteração).
<i>Avaliação da Iteração</i>	Um sumário dos resultados da iteração, ou seja, de quais os objectivos atingidos, que pedidos de alteração e correcções de <i>bugs</i> foram efectivamente realizados e quais os motivos para os possíveis desvios ao plano.	Documento textual com cerca de 2-5 páginas (por iteração).
<i>Contextualização</i>		
<i>Caso de Desenvolvimento</i>	Describe como o RUP foi adaptado para um projecto, consistindo essencialmente numa lista dos artefactos que foram utilizados e nos que não foram.	Documento textual com cerca de 6 páginas.
<i>Recomendações de Programação</i>	Reutilizaram recomendações previamente existentes internamente no que concerne à programação em linguagem Java.	Documento textual com cerca de 10 páginas.

⁵⁵ O autor refere que o segundo projecto não foi tão bem sucedido quanto o primeiro, não pela metodologia em si, mas sim por motivos imputáveis ao cliente (que deu resposta atempada a artefactos entregues e não disponibilizou os seus recursos-chave para que fosse possível efectuar um rigoroso levantamento de requisitos).

⁵⁶ Em termos de funcionalidades, pedidos de alteração, correcções de *bugs*, etc.

De um ponto de vista estatístico, a contabilização do tempo dispendido em cada tarefa por cada um dos recursos envolvidos permitiu concluir que apenas 7% do tempo de projecto foi aplicado em actividades de planeamento, gestão de requisitos e de pedidos de alteração, enquanto o tempo afecto à gestão de projectos variou entre 5% e 10%. Para além disso, consideraram que o facto de terem adoptado uma metodologia já existente e com provas dadas lhes permitiu poupar muito tempo.

No que diz respeito ao RUP, o autor deixa algumas sugestões no sentido de potenciar a sua adopção de uma forma suficientemente àgil:

- Deve proceder-se a uma escolha criteriosa de um pequeno subconjunto de artefactos a produzir e limitar o seu nível de detalhe ao que for razoável. Por exemplo, dos mais de 80 artefactos propostos pelo RUP, apenas 10-12 fazem sentido em pequenos projectos;
- O planeamento das iterações deve focar-se nos resultados desejados e não na lista de actividades que serão realizadas durante a mesma. Nesse sentido, a descrição das actividades e das disciplinas na documentação em formato electrónico do RUP pode ser considerada como a documentação de referência que pode ser consultada quando é necessário, em vez de ser usada como base para o planeamento detalhado de cada iteração.

Pelo exposto, pode concluir-se que este artigo se reveste de grande utilidade para organizações produtoras de desenvolvimento de software que, apesar da sua pequena dimensão, pretendam adoptar eficazmente uma metodologia madura e com provas dadas como o RUP. Nesse sentido, a lista de artefactos proposta é uma boa base de partida que poderão adaptar ao seu contexto. Contudo, o facto de não terem procedido a uma atribuição formal de papéis impede que se retirem conclusões sobre como deverá ser a constituição da equipa de projecto numa PME.

“A reference framework for process-oriented software development organizations”
[Fernandes, Duarte, 2003]

Seguindo uma linha completamente diferente do anterior, este artigo parte de um modelo de referência para organizações genéricas, com o objectivo de encontrar um outro mais específico e melhor adaptado à realidade de organizações que desenvolvem software de uma forma orientada aos processos. Para isso, os autores propõem uma forma de operacionalizar alguns processos de acordo com as disciplinas previstas pelo RUP, motivo

pelo qual o artigo foi sujeito à análise que se segue. Adicionalmente, o artigo termina com a apresentação dos resultados da aplicação das propostas dos autores a um caso de estudo.

Nesse sentido, e após elucidar o leitor sobre as especificidades que caracterizam a modelação de processos, os autores recorrem a várias referências para proceder a uma breve comparação entre duas abordagens distintas para o fazer, embora não clarificando o porquê da sua escolha em detrimento de outras também existentes. Note-se que, dado não ser este o propósito deste documento, as conclusões comparativas que se passam a apresentar emanam integralmente do artigo em causa e não de actividades de investigação que tenham sido conduzidas no âmbito desta dissertação, pelo que os interessados na sua fundamentação deverão consultar o artigo e respectivas referências.

Uma das abordagens referidas é a metodologia de domínio público OPEN, proposto para suportar processos flexíveis no desenvolvimento de aplicações de alta qualidade. Apesar de descrito como tendo uma forte influência técnica, o OPEN é considerado rico e bem documentado, útil tanto para gestores como para engenheiros de software. Para além disso, combina a adaptabilidade necessária à construção de processos que sirvam as necessidades de um domínio específico, ao mesmo tempo que permite a adaptação contínua destes de acordo com as necessidades que forem surgindo.

Por outro lado, o RUP é caracterizado como sendo altamente customizável⁵⁷ e com potencial para facilitar a comunicação e a execução e gestão dos processos. Adicionalmente, consideram que possui um nível de detalhe particularmente direccionado para gestores não-técnicos, embora com uma arquitectura que dificulta um processo iterativo e encontrando-se muito (talvez demasiado) centrado no modelo de casos de uso. Os autores referem também a existência de uma extensão ao RUP denominada de RUP SE (*RUP for Systems Engineering*) que, ao endereçar o desenvolvimento de sistemas, propicia uma visão mais abrangente de projectos que também incluam o desenvolvimento de hardware e que, em virtude disso, careçam da definição de papéis específicos.

Quando comparado com o RUP, o OPEN é considerado mais sólido na área de desenvolvimento, manutenção e métricas qualitativas, deixando transparecer as suas origens marcadamente técnicas. Apesar disso, os autores decidiram adoptar o RUP por o considerarem bastante bem documentado, suportado por ferramentas e, mediante configuração e selecção do subconjunto de artefactos a produzir, adaptável a cenários de

⁵⁷ Através da ferramenta RUP Builder e respectivos *plug-ins*.

reduzida complexidade, tal como vem comprovar a experiência da sua utilização com pequenas equipas, relatada no artigo anterior.

Após caracterizar a realidade das organizações orientadas aos processos e descrever a forma como usualmente estas se organizam internamente (como um conjunto de centros de competência e processos), os autores apresentam o modelo genérico representado pela Figura 2.4 para enquadrar os processos⁵⁸ de uma organização.

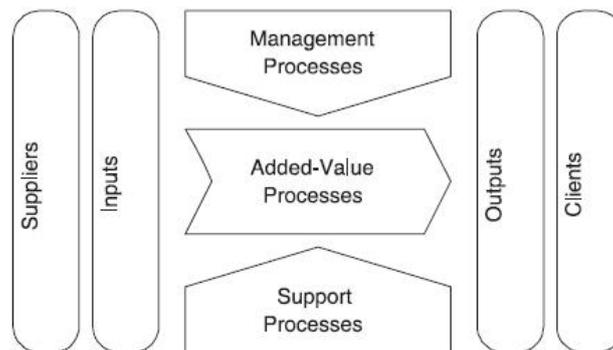


Figura 2.4 – Modelo Genérico para Organizações Orientadas aos Processos

A ideia subjacente a este modelo é que, para além dos processos estabelecidos no sentido de criar valor para o cliente (que derivam da missão da organização), existem no interior destas organizações vários outros tipos de processos essenciais para a concretização dos seus objectivos. Entre esses, podemos encontrar (à esquerda da figura) os processos relacionados com as compras e gestão de fornecedores da organização, essenciais para garantir que obtém os recursos de que necessita para o desenrolar da sua actividade, bem como os processos destinados a assegurar que as mais-valias produzidas pela organização chegam até aos seus clientes (à direita da figura). Adicionalmente, a organização necessita de operacionalizar determinados processos (na parte inferior da figura) que suportam o seu funcionamento (gestão de recursos humanos, contabilidade, etc.) e outros (na parte superior da figura) que destinados a gerir adequadamente o mesmo, como sejam a definição do modelo de negócio, da estratégia a seguir, etc.

Seguidamente, os autores descrevem o elenco de papéis sobre o qual uma organização orientada aos processos se encontra alicerçada, para além de proporem uma abordagem (que consideram pragmática) para a adopção deste tipo de orientação. Contudo, o modelo descrito é bastante genérico e não tem em conta as particularidades inerentes ao desenvolvimento de

⁵⁸ Que os autores definem como um grupo de actividades que tem como *inputs* e *outputs* um conjunto de serviços e/ou materiais.

software. Por isso, a maior contribuição e inovação deste artigo resulta do aproveitamento do enquadramento que o RUP proporciona para concretizar a adaptação do referido modelo a esse tipo de contexto, sumarizado na Figura 2.5.

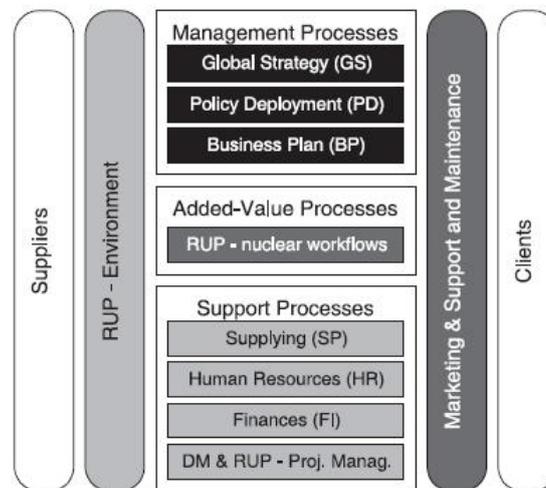


Figura 2.5 – Modelo Genérico para Organizações Produtoras de Software e Orientadas aos Processos

Assim, os autores propõem que:

- Dada a intangibilidade inerente ao software, a sua produção não dá origem ao consumo de matérias-primas. Assim, neste tipo de organizações o processo de compras pode instanciar-se na disciplina de *Environment* do RUP, já que as suas actividades irão não só incluir a procura e selecção das ferramentas mais apropriadas ao seu funcionamento, como também as necessárias à sua adaptação à realidade da organização e dos recursos à sua disposição;
- Os processos⁵⁹ portadores de valor acrescentado (que os autores apelidam de *time-to-market*) devem assentar nas seis principais disciplinas definidas no RUP: *Business Modelling, Requirements, Analysis and Design, Implementation, Test e Deployment*. Para além disso, e de acordo com as recomendações da própria metodologia, as organizações devem recorrer a diagramas de actividade (opcionalmente complementados com diagramas de estado, de objectos de negócio, etc.) para descrever a sua realidade;
- Um dos processos de suporte para os quais o RUP poderá ser relevante foi

⁵⁹ É usual que este tipo de organizações se encontre a desenvolver vários projectos em simultâneo.

apelidado pelos autores de *Data Management*⁶⁰ e compreende a gestão da informação relacionada com a disciplina de *Project Management* do RUP, no âmbito da qual são produzidos indicadores sobre o andamento dos projectos em curso e sobre os resultados da aplicação da metodologia de desenvolvimento (neste caso, o RUP). A informação produzida servirá de base a processos de tomada de decisão sobre potenciais ajustes a efectuar à gestão de projectos ou ao processo de desenvolvimento.

Um outro aspecto interessante deste artigo consiste no relato da aplicação do modelo proposto (e, conseqüentemente, da adopção do processo RUP e da notação UML) a uma organização em concreto (a unidade da Blaupunkt em Braga), no âmbito do seu projecto “*Premium Wage*”⁶¹ que, dado o seu potencial impacto social, era considerado extremamente crítico.

Tabela 2.6 – Utilização de Artefactos de Business Modelling

Artefacto	Descrição	Utilização	Motivação
<i>Business Rules</i>	Definição das políticas e condições que se devem verificar do ponto de vista do negócio.	Sim	Rigor na descrição de um novo processo de negócio.
<i>Business Use Case Model</i>	Visa mostrar como o negócio está a ser percebido e conduzido por todos os intervenientes, modelando os processos de negócio e suas interações com terceiros, usualmente sob a forma de diagramas de actividade, no sentido de clarificar “ <i>quem faz o quê</i> ”.	Sim	Primeira descrição das funcionalidades e dos actores relevantes no negócio da organização.
<i>Business Glossary</i>	Repositório de todos os termos e expressões associadas ao negócio, no sentido de facilitar a comunicação entre os intervenientes.	Não	A terminologia do negócio é comum às organizações de destino e produtora do software.
<i>Business Object Model</i>	Modelo dos objectos que sustentam o modelo de casos de uso de negócio.	Sim	Concretização dos <i>Business Use Cases</i> .
<i>Business Vision</i>	Captura os objectivos de uma determinada actividade de modelação de negócio, definindo o que deve ser modelado e porquê.	Não	A visão do negócio é comum às organizações de destino e produtora do software.
<i>Supplementary Business Specification</i>	Definições relevantes sobre o negócio que não se encontrem reflectidas nos <i>Business Use Case</i> e <i>Business Object Models</i> .	Não	Os <i>Business Use Case</i> e <i>Business Object Models</i> foram considerados suficientes.
<i>Target-Organization Assessment</i>	Descreve o estado (em termos de processos, ferramentas, recursos, competências, clientes, problemas, etc.) a organização de destino do(s) sistema(s) a desenvolver no âmbito do projecto.	Não	A organização de destino é perfeitamente conhecida pela equipa de desenvolvimento.
<i>Business Architecture Document</i>	Análise exaustiva do negócio, utilizando várias perspectivas arquitecturais para representar diferentes aspectos do negócio.	Não	Os <i>Business Use Case</i> e <i>Business Object Models</i> foram considerados suficientes.
<i>Organizational Units</i>	Uma unidade organizacional é uma colecção de papéis de negócio, entidades de negócio, relacionamentos, concretizações de casos de uso, diagramas e outras unidades organizacionais. É utilizado para estruturar o <i>Business Object Model</i> , dividindo-o em partes mais pequenas.	Sim	Mapeamento das funcionalidades dos processos de negócio para a estrutura da organização de destino.

⁶⁰ Num contexto fabril, este processo poderia assegurar a disponibilidade de índices de qualidade, por exemplo.

⁶¹ Com o objectivo de desenvolver uma aplicação para calcular o pagamento de remuneração extra aos funcionários de acordo com a sua produtividade.

Como já foi referido anteriormente, uma das formas de adaptar o RUP a organizações/contextos de menor dimensão passa pela selecção do conjunto dos artefactos a produzir e do seu nível de detalhe que, de acordo com os autores, resulta da análise das características do projecto, da organização e das restrições temporais. Por considerarem que o sucesso do projecto depende em larga medida de uma correcta compreensão dos processos de negócio a suportar, o artigo dedica especial atenção aos artefactos associados à disciplina de *Business Modelling*, quer estes digam respeito a actividades de reengenharia de processos já existentes, quer a novos processos a introduzir. A Tabela 2.6 pretende sumarizar as decisões tomadas no âmbito deste projecto concreto sobre este tipo de artefactos.

A aplicação ao caso de estudo permitiu aos autores retirar várias conclusões sobre a utilidade e aplicabilidade do RUP, entre as quais:

- O RUP em geral (e os artefactos da disciplina de *Business Modelling* em particular) apresentou bons resultados na modelação dos processos de negócio;
- Os artefactos UML, enriquecidos com os estereótipos propostos pelo RUP, facilitaram consideravelmente a comunicação entre todos os intervenientes;
- Os artefactos gerados de acordo com o RUP (entre os quais o código fonte) apresentam uma forte potencialidade de reutilização;
- O RUP provou poder ser adaptado à realidade das organizações e seus projectos;
- O conjunto de disciplinas do RUP permite lidar adequadamente com a complexidade e evita o esquecimento de algum aspecto importante relacionado com o desenvolvimento de software, diminuindo o risco de insucesso;
- A adopção do RUP e da notação UML necessita de ser acompanhada de formação aos envolvidos;
- A composição do elenco processual do RUP pode implicar um processo de reestruturação interno, no sentido de assegurar todas as valências necessárias;
- A utilização do RUP, mesmo pela primeira vez, não resulta num atraso na concretização da solução.

Assim, e perante estes resultados maioritariamente positivos, a organização decidiu normalizar o seu processo de desenvolvimento no sentido de adoptar definitivamente o RUP.

Contudo, os autores chamam ainda a atenção não só para o facto da necessidade contínua de revisão e adaptação do processo em utilização, como para o perigo de em organizações matriciais, a existência de departamentos poder bloquear o desenvolvimento de soluções que, em virtude dos novos processos de negócio identificados, ameacem a existência ou a influência destes, pelo que as organizações deverão estar conscientes deste risco. Em jeito de conclusão, os autores defendem que, sempre que uma organização puder ser representada pelo modelo da Figura 2.4, o RUP pode ser utilizado para assegurar a modelação, eventual reengenharia e transformação dos seus processos de negócio, novos ou já existentes, em aplicações que os suportem adequadamente.

Porém, e apesar da generalidade das conclusões emanadas pelo artigo serem pertinentes, não podem deixar de ser tecidas as seguintes observações ao seu conteúdo: não obstante a sua considerável abrangência, o RUP *de per si* não cobre a totalidade dos aspectos críticos do processo de desenvolvimento de software, de que são exemplo as pouquíssimas referências ao processo de preparação das futuras actividades de suporte e manutenção às aplicações desenvolvidas no âmbito do projecto. Para além disso, também não parece correcto dizer que o impacto no prazo de execução em virtude da adopção do RUP possa ser considerado negligenciável, principalmente em organizações com processos de desenvolvimento pouco formalizados, com recursos pouco experientes ou não familiarizadas com a modelação UML. Assim, para além de preparar adequadamente o processo de mudança (preparando acções de formação, documentos de ajuda, etc.), a organização adoptante deverá ser cautelosa na selecção do momento para a despoletar, de forma a não fazer perigar nem o sucesso do projecto nem o da adopção da metodologia.

“Business Modeling in Process-Oriented Organizations for RUP-based Software Development” [Duarte, et al., 2006]

Este artigo vem dar continuidade ao trabalho desenvolvido em [Fernandes, Duarte, 2003], actualizando o modelo representado pela Figura 2.5 para passar a incluir mais um processo de gestão denominado *Change Management*, destinado a permitir à organização a recolha, organização e gestão apropriada dos indicadores e experiências resultantes da aplicação dos restantes processos e, conseqüentemente, possibilitar que sejam desencadeadas as alterações necessárias e atempadas para assegurar a optimização contínua do desempenho da organização. Segundo os autores, este novo processo é essencial para qualquer

organização produtora de software que pretenda atingir os níveis CMM mais elevados⁶², que impõem a preocupação com a melhoria contínua como parte de cada um dos processos existentes no interior da organização.

Apesar de bastante útil para enquadrar o funcionamento da organização e delimitar as áreas de intervenção dos vários envolvidos, a Figura 2.5 não dispensa (e até exige) a decomposição de cada um dos processos num conjunto de actividades a realizar que, em última análise, definam de forma inequívoca “*quem é responsável por fazer o quê e quando?*”. Assim, e assumindo a utilização das disciplinas do RUP e da modelação UML na operacionalização deste modelo, as organizações devem socorrer-se de diagramas para auxiliar a compreensão e execução dos seus processos e sub-processos, de que a Figura 2.6 é exemplo.

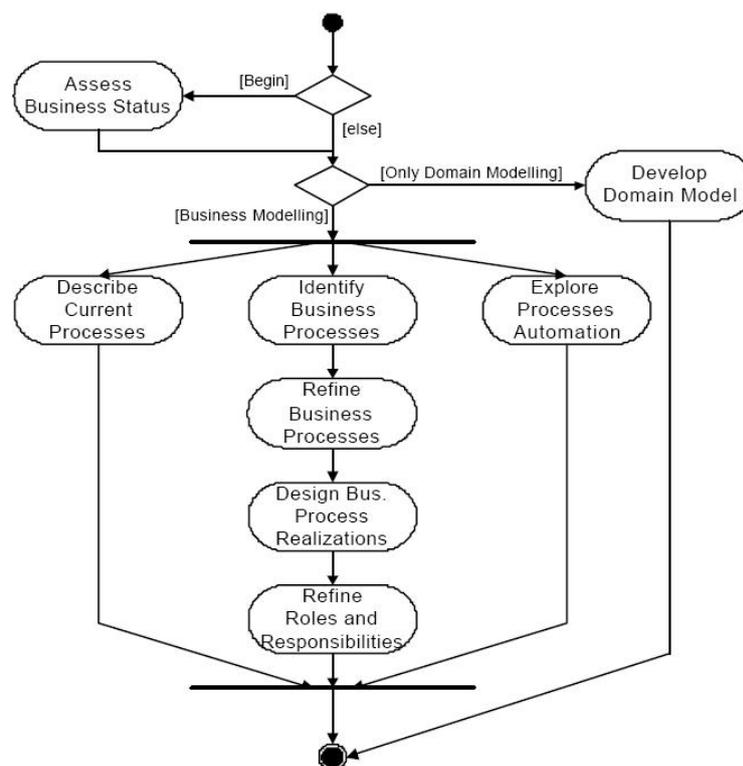


Figura 2.6 – Diagrama de Actividades do Sub-Processo de Business Modelling

Se necessário for, cada uma das actividades que compõem este tipo de diagramas pode ser decomposta em sub-actividades, até que seja possível responsabilizar directamente um determinado papel pela sua execução e pela produção dos artefactos (nem todos obrigatórios) que lhe estão associados.

⁶² O que, nos actuais contextos competitivos, se revela cada vez mais necessário.

O que diferencia este artigo de [Fernandes, Duarte, 2003] não é tanto a sua essência mas sim a sua profundidade, já que a sua principal mais-valia consiste na análise dos artefactos associados à disciplina de *Business Modelling* que foram produzidos no projecto em causa (e que se encontram descritos na Tabela 2.6) que, por considerar demasiadamente aprofundada para esta secção (que se pretende exígua), não se irão aqui detalhar.

“Dinosaur meets Archaeopteryx? or: Is there an alternative for Rational's Unified Process?” [Hesse, 2003]

O título com laivos humorísticos⁶³ dá o mote a um artigo em que o autor, apesar de reconhecer o contributo do RUP para a formalização dos processos de desenvolvimento de software e para a disseminação das práticas de modelação, procede a uma análise extremamente pragmática ao RUP, que considera um pouco confuso e demasiadamente complexo e sofisticado para uma aplicação prática. Pelo caminho, o autor elege os sete aspectos do RUP que considera mais discutíveis e para os quais propõe alternativas. Finalmente, é apresentado o modelo EOS [Hesse, 1997] (de que W. Hesse é também autor) e que, segundo este, tem potencial para suprir os defeitos que aponta ao RUP.

O primeiro conceito-chave que o autor contesta é a estruturação do ciclo de desenvolvimento do RUP nas quatro fases descritas na secção 2.2, decompostas em uma ou mais iterações e terminadas por um determinado marco, que considera herança do tradicional modelo de “*Cascata*” e não suportar adequadamente as mais recentes abordagens de desenvolvimento baseadas em componentes. Para cada fase, o RUP define um conjunto de actividades que devem⁶⁴ ter lugar e que se encontram organizadas de acordo com nove disciplinas que, nas suas palavras, para além de representarem resquícios das metodologias que estiveram na génese do RUP como hoje o conhecemos, não passam das “tradicional” fases do modelo em “*Cascata*”. Segundo o artigo, o resultado é um complexo emaranhado conceptual e terminológico de difícil aplicação em projectos compostos por vários sistemas que evoluem de forma assíncrona e independente. Aliás, a segunda tese dá continuidade a esta ideia, alegando que a primazia das fases relega para segundo plano a preponderância da engenharia de software e, com isso, desmerece a sua pretensão de ser ser “*architecture-centric*”. Assim, a metodologia assume (através das suas disciplinas) uma

⁶³ O título estabelece a relação entre o *dinossauro* UML (assim apelidado por K.D. Schewe em [Schewe, 2001] e a *Archaeopteryx* – denominação da ave mais antiga de que há registo – RUP (em virtude da miríade de documentos, orientações, ferramentas, etc que, na opinião do autor, o tornam igualmente sobredimensionado e menos viável).

⁶⁴ Tal como o RUP aconselha, apenas caso a organização adoptante considere que cada uma delas aporta mais-valia.

maior preocupação com a criação de modelos que, apesar constituírem uma representação da arquitectura, não desencadeiam as acções necessárias para a fazer acontecer. De seguida, alega-se que sendo meritória, a adopção de iterações no interior das fases “clássicas” não é eficaz porque, ao ter lugar num contexto delimitado pelas decisões tomadas em fases anteriores, cria uma resistência adicional nas situações em que os resultados obtidos aconselham uma completa reconcepção da solução, dadas as prováveis reticências da gestão do projecto em assumir tal retrocesso.

Por outro lado, o autor insurge-se contra a coexistência de fases e disciplinas que, na sua opinião, para além de redundante⁶⁵ resulta numa complexidade desnecessária. Aliás, alega que as actividades de algumas disciplinas⁶⁶ não têm nada em comum entre duas fases distintas (por exemplo, a disciplina *Analysis & Design* na fase *Elaboration* ou na fase *Construction*), acabando estas por corresponder essencialmente a tipos de actividades. Mais, defende que as várias mudanças terminológicas⁶⁷ que têm caracterizado a metodologia parecem comprovar esta tese, denotando a dificuldade dos seus autores em justificar a sua necessidade. Não obstante considerar que, a existirem disciplinas, se justificaria uma ligada à qualidade e outra à usabilidade, o autor não questiona a pertinência das disciplinas de *Business Modelling*, *Configuration & Change Management*, *Project Management* e *Environment*).

Para o evitar estes problemas, defende-se que o papel das fases deveria ser secundarizado e que as actividades e iterações fossem alternativamente associadas a sistemas e blocos arquitecturais, representando ciclos de desenvolvimento compostos por uma sequência de actividades de análise, concepção, implementação, teste e distribuição.

Uma outra ideia forte do artigo é que o RUP é de difícil aplicação a projectos de grande complexidade devido à sua visão monolítica do processo de desenvolvimento, que apenas considera (pelo menos de forma explícita) a existência de um único produto. Contudo, a abrangência⁶⁸ inerente aos projectos actuais carece de metodologias capazes de ajudar os profissionais a lidar com várias linhas de desenvolvimento independentes e assíncronas, com objectivos distintos (e eventualmente complementares), assegurando uma eficaz gestão dos

⁶⁵ Por quase haver uma correspondência de um para um entre o pico das principais disciplinas e as fases de natureza similar (*Requirements* → *Inception*; *Analysis & Design* → *Elaboration*; *Implementation & Test* → *Construction*; *Deployment* → *Transition*).

⁶⁶ *Requirements*, *Analysis & Design*, *Implementation*, *Test* e *Deployment*.

⁶⁷ As actuais *disciplines* derivam dos anteriores *workflows* que, por sua vez, já descendiam dos *process components*.

⁶⁸ De objectivos a alcançar, de tecnologias e ferramentas a utilizar, de competências a mobilizar, etc.

recursos envolvidos que, por vezes, são partilhados entre elas. Para ultrapassar esta limitação, alerta-se para a utilidade de alguns conceitos oriundos da engenharia de software ao propôr a decomposição recursiva do processo de desenvolvimento em sub-processos hierárquicos (correspondentes aos vários produtos/componentes existentes) que evoluem de acordo com a sua própria linha de desenvolvimento.

Adicionalmente, o autor diz-se convicto de que o RUP não suporta de forma conveniente a actividade do gestor de projecto, ao não lhe proporcionar critérios objectivos⁶⁹ para avaliar o progresso do projecto nem definir suficientes pontos de controlo⁷⁰. Por isso, defende-se a adopção de pontos de controlo que representem estágios⁷¹ pré-determinados da evolução prevista do projecto. Assim, se a cada estágio for associada uma data objectivo para a sua concretização, o gestor de projecto dispõe de uma ferramenta mais precisa para controlar a evolução dos trabalhos.

Finalmente, alega-se que o RUP não enquadra de forma adequada os intervenientes de um processo de desenvolvimento de software e que não envolve suficientemente os utilizadores finais durante a implementação das aplicações que lhes vão ser colocadas à disposição. Assim, o autor considera que a actividade de revisão do modelo de casos de uso (perto do início do projecto) e que a fase de transição (que apenas tem lugar após a maior parte do desenvolvimento estar concluído) são claramente insuficientes para assegurar a satisfação das expectativas dos utilizadores. Em vez disso, propõe-se a integração de uma actividade de “*usar e rever*” em cada ciclo de desenvolvimento, complementada com um processo que inclui papéis e actividades com o objectivo de, em conjunto com os utilizadores finais, rever todos os produtos que resultam das actividades de desenvolvimento.

Pelo exposto, o autor pretendeu aprender com os (supostos) erros do RUP e condensar as soluções anteriormente apresentadas no modelo EOS, que apresenta no artigo. Em jeito de balanço, pode dizer-se que algumas das teses enunciadas são efectivamente pertinentes: a utilidade do RUP é consideravelmente diminuída em projectos que envolvam várias linhas de desenvolvimento, as *milestones* que propõe são insuficientes para servir de base ao controlo

⁶⁹ Refere como exemplo desta subjectividade a frase “*modelo de use cases 10%-20% completo*” como condição de transição de uma fase para outra já que, para além de dificilmente mensurável, o que é relevante é se está feito ou não.

⁷⁰ Já que as *milestones* propostas pelo RUP para o final de cada fase podem, em projectos de alguma dimensão, distar entre si vários meses.

⁷¹ Por exemplo, “*Desenho do sistema A concluído*”, “*Implementação da funcionalidade B terminada*”, etc.

do projectos⁷², etc. Porém, nem todas as ideias-chave apresentadas no artigo são necessariamente contraditórias (podendo até ser complementares) com as propostas do RUP: por exemplo, pode considerar-se que a estruturação em quatro fases do RUP se aplica ao projecto como um todo⁷³ e não directamente às actividades de desenvolvimento dos vários sistemas/produtos que, se desejado, poderão ser enquadrados numa abordagem hierárquica e recursiva como o autor tanto apregoa.

**“A qualitative comparison of two processes for object-oriented software development”
[Henderson-Sellers, *et al.*, 2001]**

Aquando da análise efectuada a [Fernandes, Duarte, 2003], foram brevemente relatadas algumas conclusões que o artigo veiculava sobre a comparação entre o RUP e o OPEN. Seguindo uma linha análoga, este artigo procede a uma aprofundada comparação qualitativa entre estes dois processos, tendo como base um conjunto de critérios seleccionados pelos autores.

Assim, começa-se por advogar a flexibilidade que um processo de desenvolvimento de software deve revelar no sentido de poder ser adaptado a vários contextos e de auxiliar a organização a amadurecer. Esta característica é ainda mais importante em organizações que operem (ou pretendam operar) nos níveis CMM mais elevados, já que o próprio processo deve incorporar mecanismos que suportem a sua evolução de acordo com os resultados que forem sendo obtidos com a sua aplicação. Aliás, alega-se que a facilidade de adopção de processos mais complexos e sofisticados é proporcional à classificação da organização na escala CMM. De seguida, procede-se a uma breve introdução a ambos os processos sendo que, no caso do RUP, se chama a atenção para a proximidade deste em relação às teses de Walker Royce⁷⁴ e para algumas incongruências em relação ao *Unified Process*.

Dando início à comparação propriamente dita, avalia-se a flexibilidade de cada um dos processos na medida da sua capacidade de se adaptar à utilização de várias ferramentas, linguagens de programação, papéis, dimensão, etc., dada a baixa probabilidade de um mesmo projecto poder ser adoptado *ipsis verbis* em contextos substancialmente diferentes. Assim, usualmente as organizações procuram processos que possam configurar de acordo com as suas necessidades. Neste aspecto, o OPEN é caracterizado como sendo um referencial

⁷² Embora possa ser complementada com a avaliação da conclusão dos vários artefactos.

⁷³ Que usualmente têm um “ciclo” próprio, passando pela aquisição, negociação, etc.

⁷⁴ Cujas colaborações são assumidas em [Kruchten, 2004].

baseado num repositório de componentes de processo, que possibilita a personalização do processo à realidade da organização. Adicionalmente, este referencial também potencia a sua própria extensibilidade e melhoria contínua o que, conseqüentemente, ajuda a aproximar a organização dos níveis cimeiros do CMM. Por outro lado, os autores apresentam o RUP como uma instância pré-configurada do seu próprio meta-modelo⁷⁵, podendo ser classificado como minimamente personalizável. Esta personalização é suportada por um *Development Kit*⁷⁶ (disponibilizado juntamente com a metodologia) que, entre outras coisas, inclui pré-configurações para contextos-tipo (RUP completo, RUP para pequenos projectos, etc.), no sentido de tornar o processo mais célere. Contudo, a abrangência desta adaptação encontra-se limitada quer pelo facto de ter ser orientado pelos casos de uso, quer pelas estreitas ligações entre a metodologia em si e o leque de ferramentas comercializado pela *Rational* que, pelo menos à data, não disponibilizava mecanismos para incorporar comentários à sua aplicação, necessário para atingir o nível 5 do CMM.

Por outro lado, pretende-se estudar não só alguns aspectos relacionados com a terminologia e conceitos utilizados⁷⁷, como também as características dos projectos a que a metodologia pode ser aplicada, entre as quais a extensão da cobertura do ciclo de desenvolvimento de software. Enquanto o OPEN emprega termos condizentes com os *standards* desta área do conhecimento, o RUP encerra uma linguagem propositadamente diferente, donde resulta um maior esforço interpretativo. Porém, e apesar de enfrentarem algumas dificuldades⁷⁸, após análise de ambos os meta-modelos, os autores conseguiram estabelecer paralelos⁷⁹ entre a maior parte dos conceitos existentes em ambos. Focando na cobertura do ciclo de desenvolvimento, alega-se que ao contrário do OPEN, o RUP não dá a devida atenção a alguns aspectos de grande importância para o negócio de uma organização produtora de software, como sejam a reutilização e manutenção de software que desenvolve e

⁷⁵ O artigo, datado de 2000, alerta para o facto de estar prevista para 2001 a instanciação do RUP de acordo com o SPEM o que, como referi na secção 2.2, se veio a confirmar.

⁷⁶ A Rational alerta para o facto de quanto mais personalizado um processo for, mais difícil e moroso se tornará repercutir as alterações em futuras versões da metodologia.

⁷⁷ Que devem encontrar-se bem definidos, de forma não ambígua e, sempre que possível, não confundir o significado de palavras comuns. Para além disso, é desejável que encontrem correspondência directa em boas práticas e *standards*.

⁷⁸ O mapeamento de alguns conceitos do RUP (por exemplo, os *Phase* e *Workflow*) para o OPEN só poderem ser concretizados à custa de um longo processo dedutivo.

⁷⁹ Seguem-se alguns exemplos: OPEN \Rightarrow RUP, *Stage* \Rightarrow *Phase*, *Activity* \Rightarrow *Workflow*, *Producer* \Rightarrow *Worker*, *Work Product* \Rightarrow *Artifact*, *Task* \Rightarrow *Activity*, *Techniques* \Rightarrow *Guidelines*, *Assertions* \Rightarrow *Triggers*, etc.

a gestão de questões com impacto em vários projectos⁸⁰. Ainda em relação ao RUP, para além de chamar a atenção para algumas incongruências conceptuais⁸¹, o artigo descreve-o como sendo um processo em “*Cascata*” formado por uma sequência de vários elementos dessa mesma natureza.

Outro assunto que mereceu a atenção dos autores foi a existência de um possível meta-modelo subjacente a cada uma das metodologias, no sentido de não depender de descrições textuais potencialmente imprecisas. Assim, foi possível apurar a sua existência em ambas, tal como descrito nas Figura 2.7 e Figura 2.8.

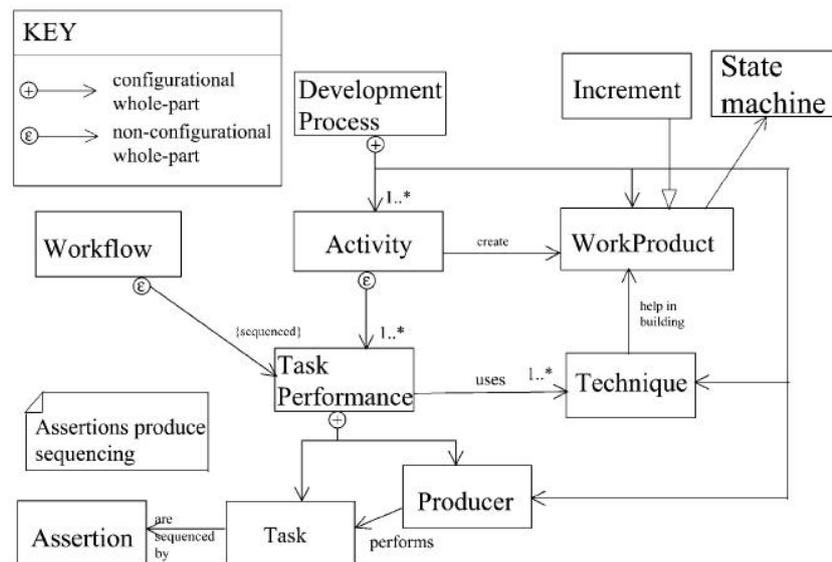


Figura 2.7 – Meta-modelo do OPEN

Também a componente de gestão de projectos foi alvo de escrutínio, tendo-se concluído que no OPEN os elementos deste tipo não se encontram confinados a actividades marcadamente dessa natureza, estando também presentes em actividades de índole genérica e em várias técnicas. Adicionalmente, o OPEN sugere que para cada projecto seja criado um documento⁸² (denominado *Project Charter*) assinado pelos principais intervenientes que, após elencar os condicionalismos do mesmo, estabeleça as directrizes que deverão guiar as actividades de gestão de projecto. Por seu lado, o RUP agrupa este tipo de actividades numa disciplina independente, paralela às restantes, e com três objectivos principais: 1)

⁸⁰ Por exemplo, a gestão racional de recursos partilhados entre vários projectos.

⁸¹ Os autores classificam-no como *incremental* e não como *iterativo*, já que os ciclos existentes parecem visar mais a disponibilização de novas funcionalidades do que a reestruturação de componentes já existentes.

⁸² Subdividido em *Business Statement, Problem Statement, Description, Context Model, Methodology, Project Resource Estimates and Schedules, Quality Assurance, Post-Implementation Review, Risk Analysis and Contingency Planning*.

proporcionar um referencial para a gestão de projectos de desenvolvimento de software; 2) proporcionar orientações para o planeamento de fases e iterações, selecção de pessoal, execução e monitorização de projectos; 3) proporcionar um referencial para o processo de gestão de risco⁸³. Contudo, o RUP é considerado omissivo no que concerne à gestão de pessoas, orçamentos, contractos, métricas e reutilização.

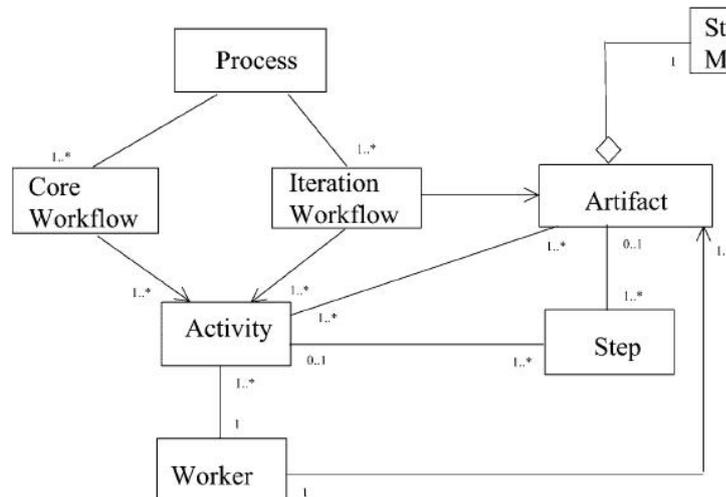


Figura 2.8 – Meta-modelo do RUP

Para finalizar, os autores tecem alguns comentários sobre vários aspectos de ambas as metodologias que, não se enquadrando nos vectores abordados anteriormente, são igualmente relevantes na sua avaliação:

- Apesar de potencialmente facilitar a comunicação entre cliente e fornecedor, a excessiva importância dada pelo RUP aos casos de uso deixa sem resposta os requisitos (nomeadamente os não funcionais) que não podem ser representados dessa forma. Em contrapartida, para além de suportar essa abordagem, o OPEN também suporta várias outras (*responsability-driven*, *data-driven*, etc.), já que é improvável que o mesmo seja adequado para todos os contextos;
- Uma das principais mais-valias do RUP é o conjunto de ferramentas da *Rational* que lhe dão suporte e que auxiliam os elementos da equipa nas actividades de gestão de configurações, teste, modelação, etc. Dada a sua natureza independente, o OPEN não se encontra associado a nenhuma ferramenta⁸⁴ em particular;

⁸³ Embora proporcione pouco detalhe sobre como evitar, transferir ou aceitar o risco (através de mitigação ou de definição de um plano de contingência).

⁸⁴ Surgiu recentemente uma ferramenta *open-source* que suporta a customização de vários processos de desenvolvimento de software, entre os quais o OPEN e o RUP. Para mais informações, consultar <http://www.eclipse.org/epf/>.

- O OPEN inclui um grande número de tarefas e técnicas dedicadas à qualidade, estimativa de custos, concepção e estabelecimento de métricas. Adicionalmente, e ao contrário do RUP, adapta-se bem a vários tipos de projectos⁸⁵ e suporta qualquer notação de modelação, não forçando a utilização da UML;
- O RUP é demasiadamente optimista e não promove uma frequente monitorização e análise de risco, o que reduz a sua eficácia na detecção de situações de risco antes da fase de testes, quando já poderá ser tarde demais para reagir.

Assim, a abrangência da comparação levada a cabo por este artigo serve pelo menos para alertar os interessados em aplicar qualquer uma das referidas metodologias para os alguns dos seus vícios e virtudes, para que os possam levar na devida conta no seu processo de selecção.

2.5. Conclusões

Ao longo deste capítulo procurou-se proporcionar uma maior compreensão sobre o complexo, detalhado e exigente processo de desenvolvimento de software que se esconde por detrás da pequena sigla RUP, cuja abordagem extremamente prática que parece resultar da experiência profissional acumulada pelos seus criadores ao longo de vários anos, durante os quais se confrontaram diariamente com os problemas que o processo pretende evitar ou minimizar. Contudo, e apesar da enorme abrangência que tem (ou pelo menos pretende ter), escalpelizando quem deve fazer o quê, quando e o que deve resultar dessa actividade, o sucesso na aplicação do RUP⁸⁶ depende, em parte, de uma mudança cultural de todos os seus intervenientes (incluindo os externos à organização, como sejam os clientes), tradicionalmente mais familiarizados com o processo em “*Cascata*”. Vimos também que a operacionalização do RUP se baseia na participação de um vastíssimo elenco processual de quase quarenta elementos cuja composição, de acordo com a literatura específica da área, não é consensual. Por outro lado, a análise levada à cabo sobre algumas aplicações (de que é exemplo [Hirsch, 2002]) do RUP relevantes para o contexto PME indicia que, tal como preconizado pelos autores do RUP, a sua aplicação em PMEs pode ser bem sucedida mediante uma análise e selecção dos componentes do processo pertinentes nesse contexto

⁸⁵ Não apenas ao desenvolvimento de projectos de raiz e isolados, mas também ao desenvolvimento de novas funcionalidades para uma aplicação já existente (na forma de repetições do ciclo de vida do projecto), desenvolvimento contínuo e concorrente de projectos (com tarefas e técnicas para gerir a reutilização, alocação de recursos, gestão de pessoas e os papéis organizacionais).

concreto. Porém, os artigos analisados expuseram também que, ao não prever mecanismos para incorporar os comentários à aplicação do processo, a adopção *tout cour* do RUP por organizações que pretendam atingir níveis elevados na escala CMM não é recomendada.

Adicionalmente, evidenciou-se a falta de informação existente no que diz respeito a questões relacionadas com o elenco processual do RUP, o que, agravado pela carência de literatura científica dedicada especificamente a esse tema, pode constituir uma barreira que é necessária ultrapassar no seu processo de adopção. Convém ainda referir que o RUP é omissivo sobre algumas questões relevantes, como sejam: o desenvolvimento contínuo e concorrente de projectos que, como foi visto em [Henderson-Sellers, *et al.*, 2001], implica preocupações com actividades de preparação de suporte, manutenção e gestão de recursos partilhados; a possível existência de intervenientes (que não o gestor de projecto) encarregues (ou com essa capacidade) de facilitar a interacção com terceiros (sejam eles clientes, fornecedores, parceiros, etc.), o que é extremamente frequente em culturas latinas.

Pelo exposto, é possível não só colmatar estas falhas como também contribuir para uma mais fácil aplicação do RUP em contextos PME procedendo a uma configuração do mesmo com esse objectivo, consubstanciada na revisão e simplificação do conjunto de papéis envolvidos na sua operacionalização, a desenvolver nos capítulos que se seguem.

⁸⁶ Ou qualquer outro método que advogue o desenvolvimento iterativo.

3. Configuração do RUP para Utilização em PMEs

Tomando como base o elenco processual previamente apresentado, este capítulo descreve um processo de simplificação levado a cabo com o objectivo de obter um modelo (denominado de *base*) passível de ser efectivamente aplicado em contextos PME, consubstanciado num conjunto de treze papéis distintos. Contudo, e antecipando o caso de este modelo ainda poder ser considerado como demasiadamente exigente para ser adoptado por organizações de reduzida dimensão, é também proposto um modelo alternativo (denominado de *reduzido*) que, fruto de uma condensação e reorganização de funções, reduz para oito a dimensão do lote de intervenientes processuais. Adicionalmente, e para evitar a ocorrência de situações que possam lesar a eficácia dos modelos propostos, são ainda recomendadas algumas restrições à acumulação de determinados papéis, quer dentro do mesmo projecto quer entre projectos distintos. Por outro lado, e no sentido de auxiliar as organizações adoptantes a enquadrar devidamente o modelo por si seleccionado, é apresentado um mapeamento entre os vários papéis e as funções usualmente encontradas numa PME ligada ao desenvolvimento de software.

3.1. Introdução

No capítulo anterior procedeu-se a uma análise crítica do RUP, descrevendo os seus conceitos-chave, intervenientes, fases e disciplinas para, dessa forma, enquadrar o trabalho a desenvolver no âmbito desta dissertação. Para além disso, foram também examinados alguns casos de aplicação do RUP a vários tipos de projectos e organizações, com alguns resultados positivos⁸⁷.

Contudo, e sem prejuízo das conclusões apresentadas na secção 2.5, quem conhecer o “teatro de operações” típico de uma PME nacional do sector do desenvolvimento de software terá alguma relutância em acreditar na aplicabilidade (pelo menos de uma forma purista) do RUP nesse tipo de contexto, dadas algumas das suas especificidades.

⁸⁷ Note-se que a ausência de determinados factores críticos (como a colaboração do cliente) é suficiente para pôr em risco o sucesso de qualquer projecto, seja qual for o método observado.

Em primeiro lugar, quer os criadores do RUP, quer uma boa parte dos que o têm aplicado são originários de contextos onde, tipicamente, as organizações atingem dimensões incomensuravelmente superiores às PME's nacionais, sendo que esse factor escala poderá potenciar:

- A disponibilidade de um maior número de recursos humanos para o desempenho dos papéis definidos no processo, tornando virtualmente desnecessária a acumulação de vários papéis por um mesmo interveniente (com toda a complexidade que lhe é inerente);
- A disponibilidade de recursos financeiros mais significativos, que possam ser investidos em ferramentas de desenvolvimento⁸⁸ que facilitem a colaboração entre os vários intervenientes, a integração das suas prestações e a reutilização do trabalho efectuado anteriormente;
- A oportunidade de participar em projectos de grande dimensão, duração e complexidade, nos quais seja possível aplicar um processo tão abrangente como o RUP em toda a sua plenitude;
- A adopção do desenvolvimento/pensamento iterativo e incremental, na sequência dos amplos períodos de implementação dos projectos em que usualmente se vêem envolvidas. Pelo contrário, a reduzida dimensão dos projectos típicos dos contextos PME e a (ainda) reduzida cultura iterativa dos clientes⁸⁹ não têm contribuído para a disseminação dos processos iterativos pelas organizações de menor dimensão;
- A especialização de cada interveniente num determinado (e único) papel. Obviamente, isso torna-se muito difícil de assegurar quando os recursos humanos disponíveis são mais limitados, uma vez que é usualmente necessário que um indivíduo desempenhe, de forma simultânea ou alternada, vários papéis. Assim, deixa de existir especialização (pelo menos de forma generalizada) para passar a existir polivalência, embora com menor conhecimento e experiência em cada um dos papéis.

⁸⁸ De que a oferta da *Rational* é exemplo.

⁸⁹ Que, com frequência, não estão psicologicamente preparados para ver o sistema antes de este se encontrar finalizado (com alguns erros por corrigir, com interfaces de utilizador por finalizar, etc), potenciando conclusões erradas (falta de qualidade, reduzida usabilidade, etc). Pelo exposto, é usual que os clientes prefiram um *big bang*.

Para além disso, usualmente, as organizações de menor dimensão apresentam um modo de funcionamento orgânico⁹⁰ (em detrimento de mecânico⁹¹), tendendo a privilegiar a eficácia (em detrimento da eficiência). Nesse sentido, a existência e cumprimento de procedimentos normalizados e com um elevado nível de detalhe é algo a que os colaboradores deste tipo de organizações poderá não estar habituado.

Adicionalmente, ao definir-se como um processo orientado pelos casos de uso e *architecture-centric*, o RUP assume que os seus intervenientes, quer técnicos quer não-técnicos, não só têm acesso a ferramentas de modelação UML, como possuem o conhecimento necessário para as utilizar correctamente. Contudo, apesar de existirem ferramentas de modelação UML de utilização livre de licenciamento, usualmente estas apresentam funcionalidades limitadas e apenas se adequam a contextos de utilização individuais e de baixa complexidade (por vezes apenas permitem a criação de um reduzido número de diagramas ou objectos). Logo, a adopção generalizada e distribuída pela organização da modelação UML, com colaboradores distintos a trabalhar de forma cooperativa e concorrente sobre um conjunto de diagramas, pode implicar a utilização de aplicações mais completas e integradas⁹² que podem atingir custos demasiadamente elevados para uma PME. Por outro lado, é essencial que os intervenientes no processo dominem quer a notação UML quer a ferramenta utilizada para a consubstanciar que, consoante o seu grau de sofisticação, poderá encerrar uma complexidade considerável. Nesse sentido, poderia ser necessário ministrar a todos os envolvidos acções de formação sobre esses tópicos, resultando num potencial incremento dos custos de adopção do processo, nem sempre tolerável para uma PME.

A contribuir para essa inadequação está também a longa lista de artefactos que prevê, nem sempre compaginável com a realidade dos projectos de menor dimensão por daí resultar, inevitavelmente, um considerável custo de produção dos mesmos (que não possa ser repercutido para a entidade cliente).

Por último, convém realçar que o RUP, enquanto produto, é fornecido na forma de um

⁹⁰ Caracterizado por uma descentralização do poder e do saber (sustentada em trabalho de equipa/em rede), por uma polivalência/flexibilidade dos indivíduos e pela informalidade de processos.

⁹¹ Caracterizado por uma centralização do poder (sustentada numa hierarquia rígida), por uma especialização dos indivíduos e pela existência de rotinas claras e formalizadas para as várias situações previstas, que devem ser observadas com rigor.

⁹² Que, por exemplo, efectuem a gestão do processo de aprovação de cada diagrama alterado ou integrem com um sistema de controlo de versões que mantenha registo de todas as versões existentes de um determinado diagrama e impeça perdas de informação.

conjunto de páginas HTML [Rational, 2006], destinadas a guiar a participação dos vários intervenientes no processo. Contudo, o conteúdo (quer texto quer imagens) destas encontra-se em língua inglesa o que, para além de potencialmente diminuir o ritmo da sua leitura, poderá dificultar a sua compreensão por alguém pouco familiarizado com a língua.

Dito isto, pode concluir-se que o RUP, tal qual se encontra proposto, não é passível de aplicação (bem sucedida) em contextos PME, em virtude do elevado alforge processual que encerra. Contudo, convém referir que essa desadequação não resulta da incorrecção das ideias ou pressupostos que o RUP encerra, já que este, a peca por alguma coisa, peca pela sua enorme ambição de abrangência. Para além disso, o RUP não é proposto como um dogma que apenas pode ser adoptado na sua plenitude. Pelo contrário, os seus autores conceberam-no como um processo que não só pode, como deve ser adaptado aos condicionalismos das organizações receptoras. Para esse efeito, o fornecimento da metodologia inclui uma aplicação (denominada de *RUP Builder*) que permite a criação de uma versão específica da mesma, de acordo com as características do projecto em causa. Com esta ferramenta é possível seleccionar as áreas do processo relevantes e, recorrendo a uma biblioteca de componentes de processo⁹³, gerar uma configuração do RUP (e respectiva documentação) apropriada a esse cenário de utilização. Infelizmente, apesar de útil, a documentação gerada encontra-se (à semelhança da completa) em língua inglesa, o que, apesar de não impedir o seu uso, induz perdas de eficiência em virtude do esforço interpretativo adicional associado a cada utilização. Assim, é precisamente neste processo de adaptação que se considera que o contributo desta dissertação poderá ser mais pertinente, ao demonstrar como o elenco processual inerente à metodologia pode ser configurado, sem necessidade de recorrer à utilização de nenhuma aplicação comercial, no sentido de viabilizar a sua adopção por organizações de dimensão PME. Por conseguinte, ao longo da secção 3.2 proceder-se-á à apresentação dos princípios que serviram de orientação a esse esforço de simplificação do elenco processual.

Por outro lado, o RUP não deve ser encarado como uma antítese dos denominados *métodos ágeis*⁹⁴ já que, segundo os seus autores, visa proporcionar às organizações um conjunto de recomendações relacionadas com o processo de desenvolvimento de software que só devem ser individualmente aceites caso lhes seja reconhecida uma inerente e

⁹³ Relativos a certas tecnologias, domínios de utilização, ferramentas, etc.

⁹⁴ Definido *agilidade* como a capacidade de reagir de forma satisfatória e atempada a um determinado acontecimento.

significativa mais-valia. Aliás, em entrevista recente⁹⁵, Per Kroll⁹⁶ corrobora essa ideia, realçando os pontos comuns entre ambas (como sejam o desenvolvimento iterativo e clara definição de responsabilidades) e caracterizando o RUP como uma base de conhecimento que, segundo ele, não deve ser adoptada (pelo menos de uma só vez) na sua totalidade. Porém, apesar da obra publicada por este autor⁹⁷ no sentido de auxiliar as organizações no seu esforço de agilização do RUP, esta corrente de opinião parece ignorar que, para simplificar algo, é necessário compreendê-lo primeiro, o que poderá implicar um período de estudo de que uma boa parte das organizações PME simplesmente não dispõe.

Pelo exposto, uma organização nacional de reduzida dimensão na área do desenvolvimento de software deve olhar para o RUP como um enquadramento para o seu processo de desenvolvimento, procurando destilar os conceitos e ideias-chave passíveis de fácil adaptação ao seu contexto, deixando os restantes para ponderação e/ou adopção futura. Nesse sentido, para potenciar a sua flexibilidade e capacidade de reacção às, praticamente inevitáveis, alterações de requisitos, a organização poderá adoptar um processo de desenvolvimento iterativo, destinado a permitir que os seus clientes tomem contacto com versões preliminares dos artefactos tão cedo quanto possível. Dessa forma, será naturalmente incentivada a criação de rotinas⁹⁸ que aumentem a sua eficiência⁹⁹ e eficácia¹⁰⁰. Contudo, para minimizar os possíveis efeitos de atrito, as organizações devem envolver no processo os seus colaboradores e clientes, esclarecendo-os sobre as vantagens daí resultantes. Adicionalmente, poderão ser definidas boas práticas sobre o conjunto mínimo e limitado de artefactos que deverão ser produzidos no âmbito de cada projecto, no sentido de assegurar a organização do seu processo de desenvolvimento e a documentação apropriada às futuras actividades de exploração e manutenção. Porém, e dado que a lista de artefactos proposta pelo RUP é bastante extensa, a organização poderá considerar abordagens simplificadas, como a apresentada em [Hirsch, 2002]. Assim, artefactos que não constem desse conjunto apenas devem ser produzidos caso o cliente os valorize e esteja disposto a custear.

⁹⁵ Que pode ser visionada em <http://www.infoq.com/interviews/rup-agile-per-kroll>.

⁹⁶ Director da Rational Software Corporation e actual responsável pelo desenvolvimento e gestão do RUP.

⁹⁷ Ver [Kroll 2003] e [Kroll 2006].

⁹⁸ Integração de componentes, passagem a produção, publicação de documentação, etc.

⁹⁹ Derivada da crescente habituação às mesmas e da consequente optimização daí resultante.

¹⁰⁰ A periodicidade da sua execução contribui para a minimização de situações de esquecimento.

Por outro lado, no que diz respeito ao planeamento-base a utilizar e respectivas *milestones* será interessante analisar as ideias expressas em [Hirsch, 2002] já que, sem renegar as recomendações do RUP, procura chegar a um conjunto pragmático e objectivo de momentos-chave relevantes num processo de desenvolvimento de software. Caso o domínio da notação UML não se encontre disseminado de forma homogénea pelos recursos humanos da organização com participação no seu processo de desenvolvimento de software, esta poderá promover a frequência por parte destes de acções de formação sobre essa temática. Inclusivamente, e dada a usual limitação de recursos financeiros, a formação poderá ser ministrada por um elemento da equipa técnica com experiência na área.

No que concerne à dificuldade de acesso ao leque de ferramentas recomendadas pela *Rational* para suportar a aplicação do RUP, e apesar de não ser possível obter uma integração tão transparente e confortável¹⁰¹ para o utilizador, esta poderá ser superada adoptando um conjunto de aplicações que, apesar de independentes, permitam cobrir as várias necessidades dos elementos da equipa de projecto, de que as seguintes são exemplo:

- *Levantamento de Requisitos:* A utilização de uma ferramenta específica para efectuar a gestão de requisitos (como por exemplo, o [*Rational RequisitePro*]) poderá ser substituída pelo uso combinado de um processador de texto¹⁰² (para elencar e descrever textualmente os requisitos identificados) e de uma aplicação de modelação UML¹⁰³. Para suprir o facto de algumas destas aplicações não facilitarem o trabalho cooperativo e integrado dos vários participantes, poderá ser utilizada um servidor de controlo de versões¹⁰⁴ e/ou uma aplicação de gestão documental¹⁰⁵;
- *Planeamento:* Para a elaboração dos habituais diagramas de Gantt destinados a enquadrar a execução do projecto, poderá ser utilizada a ferramenta comercial mais popular para esse efeito (o [*Microsoft Project*]) ou, no caso de restrições orçamentais mais significativas, aplicações de utilização livre como o [*Mr Project*],

¹⁰¹ Em virtude da complexidade resultante de ter que utilizar uma dezena de aplicações distintas em vez de duas ou três.

¹⁰² De natureza comercial (como o [*Microsoft Word*], por exemplo) ou até livre (como o [*Open Office*], por exemplo).

¹⁰³ Com maiores funcionalidades e um licenciamento acessível (como o [*Magic Draw*] ou o [*Visual Paradigm*], por exemplo) ou com funcionalidades mais limitadas e de utilização livre (como o [*Dia*], por exemplo).

¹⁰⁴[*CVS*], [*Subversion*], etc.

¹⁰⁵ Como por exemplo o [*Alfresco*], que é de utilização livre.

por exemplo;

- *Comunicação*: O sucesso do trabalho em equipa depende, em grande medida, de uma eficaz comunicação entre os seus elementos, que cada vez mais se encontram geograficamente distantes. Nesse sentido, poderá recorrer-se a ferramentas como a [Assembla] que, ao disponibilizar, de uma maneira integrada e gratuita, um ambiente Wiki¹⁰⁶, fóruns de discussão (com alertas associados), salas de *chat*¹⁰⁷, podem funcionar como facilitadores comunicacionais;
- *Gestão de Projecto*: No que diz respeito à afectação de tarefas, planeamento de versões e iterações e gestão de *bugs*, e apesar de estas necessidades serem parcialmente cobertas pela ferramenta [Assembla], poderá ser utilizado o [JIRA], ferramenta com enormes potencialidades (incluindo a possibilidade de registar e consultar o tempo envolvido na realização de cada tarefa) e com um licenciamento bastante acessível (pelo menos nas versões mais limitadas);
- *Desenvolvimento*: Para suportar a actividade dos programadores afectos ao projecto, existem várias aplicações de utilização livre com enormes potencialidades e fácil integração com servidores de controlo de versões (como, por exemplo, o [Subversion] disponibilizado gratuitamente pela ferramenta [Assembla]), de que o [Eclipse] é exemplo;
- *Gestão do Processo*: Como já foi referido anteriormente, o RUP inclui uma ferramenta dedicada à gestão e adaptação do próprio processo. Porém, mesmo numa área tão específica quanto esta, existem alternativas interessantes que poderão ser consideradas, como seja o [Eclipse Process Framework], que permitiriam não só definir o processo específico adoptado pela organização, como também gerar documentação semelhante à fornecida juntamente com o RUP no sentido de apoiar o desempenho de cada função, com a vantagem de a mesma se encontrar em português.

¹⁰⁶ Sistema que permite a criação e alteração (colaborativa e concorrente) de páginas Web, promovendo assim uma fácil partilha de informação.

¹⁰⁷ Sistema que possibilita que vários participantes troquem mensagens entre si (de uma forma privada ou pública), entregues praticamente de imediato.

Por último, e no sentido de ultrapassar as dificuldades inerentes à adopção de um elenco processual de trinta e sete elementos, a organização adoptante poderá procurar abordagens que, preservando os princípios subjacentes aos papéis propostos pelo RUP, se revistam de menor complexidade a esse nível. Contudo, dado que a literatura analisada no capítulo anterior parece não responder de forma adequada a esta necessidade, foi precisamente nesta área que se considerou que o contributo desta dissertação poderia ser mais relevante, no sentido de diminuir as barreiras à adopção do RUP por uma PME nacional.

Pelo exposto, é pertinente identificar quais os papéis presentes no RUP que, apesar de importantes em projectos de maior envergadura ou complexidade, poderão ser relegados para segundo plano, no sentido de destilar os intervenientes críticos ao sucesso do sua adopção. Porém, não se quer com isto dizer que as responsabilidades inerentes aos papéis secundarizados devam ser ignoradas. Em vez disso, estas deverão ser assumidas (eventualmente de uma forma mais ligeira) pelos intervenientes remanescentes. Na secção que se segue irá ser explanada a linha de raciocínio seguida no sentido de o concretizar.

3.2. Simplificação do Elenco Processual

Qualquer empresa¹⁰⁸ possui, independentemente da sua dimensão, uma estrutura orgânica (seja mais ou menos formal) que identifica os papéis de cada um dos seus colaboradores, delimita as suas áreas de intervenção e define as responsabilidades que devem assumir no sentido de concretizar os objectivos da organização. Nesse sentido, é vulgar a definição de organigramas e a atribuição de cargos específicos¹⁰⁹ aos seus colaboradores.

Por outro lado, e como já foi explanado na secção 2.3, o RUP caracteriza quase quarenta papéis relevantes para um processo de desenvolvimento de software, associando a cada um deles um determinado conjunto de responsabilidades específicas. Contudo, bastará uma breve análise sobre a aplicabilidade deste elenco processual ao contexto de uma PME para concluir que de acordo com os dados referidos na secção 1.2, a esmagadora maioria das PMEs nacionais no sector de desenvolvimento de software nem sequer possui um número de colaboradores tão elevado (quanto o número de intervenientes previsto). Assim, mesmo considerando que, no contexto de um determinado projecto, uma mesma pessoa possa ser

¹⁰⁸ Excluindo obviamente as empresas unipessoais.

¹⁰⁹ Por exemplo, *Director-Geral, Director de Vendas*, etc.

nomeada para diversos papéis, uma acumulação excessiva¹¹⁰ poderá revelar-se demasiadamente complexa¹¹¹ ou exigente¹¹².

Do exposto resulta a convicção de que, para que seja passível de aplicação num contexto de menor exigência e/ou complexidade, um processo de desenvolvimento de software deverá alicerçar-se na participação de um número bastante mais limitado de intervenientes (com responsabilidades distintas). Porém, um processo linear de adaptação do conjunto de actores definido pelo RUP aos papéis específicos existentes numa determinada organização, pode encontrar dificuldades resultantes das especificidades dos mesmos.

Assim, para que possa ser concretizado de forma mais eficiente, este processo deverá decorrer em três etapas distintas:

1. É necessário desenvolver um esforço para reduzir significativamente o número de papéis relevantes num processo de desenvolvimento de software, no sentido de o tornar mais facilmente compreensível e, conseqüentemente, aplicável. Para isso, cada um dos papéis propostos pelo RUP deverá ser analisado de modo a avaliar se é essencial (e deve ser mantido na sua essência) ou não (e as suas competências devem ser divididas entre um ou mais dos papéis essenciais existentes);
2. Devem ser identificadas as restrições recomendadas à acumulação por uma mesma pessoa de mais do que um papel do conjunto obtido após o passo anterior;
3. É possível propor o mapeamento do novo elenco processual para alguns dos papéis-tipo que podem ser usualmente encontrados numa PME do sector de desenvolvimento de software. Contudo, convém notar que este mapeamento não pretende ser um dogma e que deverá ser revisto no contexto de cada PME que o pretenda adoptar, de acordo com as especificidades do seu figurino organizacional.

Assim, ao longo desta secção irá ser descrito o caminho seguido na concretização do primeiro passo, deixando os restantes para as secções subsequentes. Nesse sentido, passam a apresentar-se dois modelos, apelidados de *Base* e *Reduzido*, correspondentes a processos de

¹¹⁰ Por exemplo, no caso de serem atribuídos a um mesmo colaborador mais do que três papéis distintos.

¹¹¹ Se, no meio de todas as competências que lhe seriam atribuídas, um colaborador perder a consciência de que é suposto que execute algumas delas.

¹¹² No caso de um colaborador, mesmo sabendo que deveria levar a cabo determinada(s) tarefa(s), simplesmente não as consiga compatibilizar com as restantes.

simplificação de menor e maior profundidade, respectivamente. Dessa forma, pretende-se ir ao encontro de duas realidades distintas existentes no universo usualmente denominado de PME. A primeira é a das empresas de dimensão média, que essencialmente procuram um processo que as auxilie a conceber e implementar soluções com bons níveis de qualidade (percepcionados pelo cliente) e a lidar com a complexidade inerente a projectos de média/alta dimensão. A outra realidade é a das empresas de pequena dimensão, principalmente preocupadas em encontrar um processo que lhes permita não só gerir adequada e racionalmente o âmbito dos projectos em que se encontram envolvidas, como também obter melhor coordenação de actividades que possam traduzir em eficiência¹¹³ de utilização dos seus limitados recursos.

Porém, deve lembrar-se que a redução do número de envolvidos num processo de desenvolvimento de software resulta inevitavelmente numa maior criticidade dos desempenhos individuais, já que torna necessário que cada um assuma um leque bastante mais alargado de funções, usualmente sem que disponha de mais tempo ou recursos para as desempenhar. Ou seja, e recorrendo a uma analogia, quando um jogador de uma equipa de futebol é expulso, os que ficam vêm-se na necessidade de assegurar as responsabilidades que lhe estavam confiadas (marcação de um determinado jogador, cobertura de uma determinada área do terreno, etc.). Com isso, aumenta não só o desgaste a que cada elemento estará sujeito, como também a probabilidade de cometerem um erro (por acção ou omissão) que, se houvesse um maior número de pessoas envolvidas no processo, poderia ser evitado por um outro interveniente. Para além disso, nem sempre é fácil encontrar as competências necessárias¹¹⁴ nos intervenientes que ficam para a execução das funções anteriormente atribuídas aos papéis suprimidos.

Dito isto, passa a apresentar-se o modelo base proposto.

3.2.1. Modelo Base

De modo a concretizar os objectivos propostos, foi conduzida uma análise detalhada ao elenco processual do RUP, no sentido de filtrar os papéis considerados essenciais para a sua aplicação. Para alicerçar essa decisão definiu-se que, para ser considerado essencial, um papel deveria satisfazer pelo menos uma das seguintes condições:

¹¹³ Resultante de uma redução de desperdício de tempo (evitando que vários elementos prossigam a mesma actividade sem o saberem), de um aumento de produtividade (em virtude de uma maior consciência/familiarização de cada indivíduo com as funções de que está incumbido), etc.

- **C1:** Ter um leque de responsabilidades atribuídas que quando negligenciadas são, *de per si*, suficientes para inviabilizar o sucesso¹¹⁵ do projecto;
- **C2:** Requerer, para o bom exercício das suas funções, formação ou competências muito específicas e consideravelmente distintas de qualquer um dos outros papéis;
- **C3:** Existam potenciais conflitos de interesse entre o exercício desse papel e a acumulação de outro(s) para o qual(is) seja necessária formação ou competências similares.

Como resultado desse processo de análise, recomenda-se que os papéis que se passam a identificar (juntamente com as respectivas condições aplicáveis) sejam considerados essenciais e, conseqüentemente, pertencentes ao modelo base.

System Analyst [C1, C2]

A gestão de âmbito é fulcral ao sucesso de cada projecto, dada a quase inevitável ocorrência ao longo do mesmo de diferendos de opinião (entre cliente e fornecedor) sobre as características do objecto do contrato. Logo, a participação deste interveniente no processo é importante para, desde o início¹¹⁶, identificar e documentar com o máximo rigor os requisitos (funcionais ou não funcionais) apresentados pelo cliente, no sentido de possibilitar ao fornecedor estimar correctamente o esforço envolvido no cumprimento dos mesmos. Dessa forma, deverá ser possível estabelecer um entendimento comum¹¹⁷ sobre o que um se propõe a adquirir e o outro a fornecer, sobre o qual será futuramente alicerçado o processo de controlo de alterações.

Contudo, para além da relevância deste papel, é extremamente importante que o(s) indivíduo(s) a quem se encontra atribuído possuam uma formação apropriada, incidindo essencialmente em duas vertentes: primeiro, para além de conhecimentos básicos de gestão, é desejável que conheçam detalhadamente o negócio do cliente, para facilitar o seu entendimento sobre as reais motivações e pertinência dos requisitos apresentados por este. Segundo, para que possa desenvolver a sua actividade da forma mais adequada e de acordo

¹¹⁴ Pelo menos a um nível mínimo.

¹¹⁵ Que pode ser medido em termos de cumprimento de prazos estabelecidos, obtenção da qualidade desejada, cumprimento dos níveis de serviço contratados, etc.

¹¹⁶ Desejavelmente logo a partir da fase de aquisição do projecto.

¹¹⁷ Eventualmente consubstanciado numa proposta comercial.

com as melhores práticas definidas, é desejável que o interveniente possua formação e prática da denominada Eng^a de Requisitos (*Software Requirements* na terminologia do SWEBOK como se pode comprovar em [IEEE, 2004]).

User-Interface Designer [C2]

A abrangência da intervenção deste papel num determinado projecto varia consoante a natureza dos objectos desenvolvidos no âmbito do mesmo. Não obstante, mesmo não se podendo considerar este papel como crítico, é um facto que o seu bom desempenho depende em boa medida de uma sólida formação em áreas de conhecimento específicas (como a ergonomia do software), cujo domínio não se pode considerar generalizado nos profissionais da Eng^a de Software.

Database Designer [C2]

À semelhança do papel anterior, também este é considerado essencial ao processo não pela importância do seu desempenho para o sucesso do projecto, mas sim pela especificidade dos seus conhecimentos que, apesar de afluídos no âmbito das licenciaturas em informática, usualmente não assumem a profundidade necessária. Logo, é vulgar que essas lacunas sejam colmatadas com a frequência de acções de formação, frequentemente promovidas pelas entidades associadas a um motor de base de dados específico. Assim, é desejável que possua (pelo menos) as seguintes competências:

- Configuração e optimização de motores de bases de dados;
- Conhecimentos avançados sobre a configuração de índices, *views* e *constraints*;
- Conhecimentos avançados sobre a implementação de *triggers* e *stored procedures*;
- Conhecimentos avançados sobre normalização de modelos de dados.

Implementer [C1]

Como é óbvio, por melhor que seja a arquitectura e concepção de um projecto de desenvolvimento de software, este não será bem sucedido sem a participação deste tipo de intervenientes, que os concretizam na implementação dos sub-sistemas e componentes que sustentam a funcionalidade desejada.

Integrator (ou System Integrator) [C1, C2]

Mesmo num contexto de uma PME (que, por regra, dispõe de um lote limitado de recursos humanos) é vulgar que se encontrem afectados a um determinado projecto vários *Implementers*, prosseguindo, cada um deles, um conjunto de objectivos específico. Contudo, para que seja possível assegurar que os trabalhos decorrem de uma forma harmoniosa e eficiente, é essencial que alguém se encarregue de manter os *Implementers* ao corrente do contexto do projecto, de identificar as tarefas a realizar e de nomear o responsável por cada uma delas. Para além disso, é usual que o trabalho realizado por um dos *Implementers* seja, por vários motivos¹¹⁸, interdependente do trabalho dos restantes. Logo, é importante que alguém se encarregue de decidir a forma como os trabalhos individuais irão ser integrados e incorporados no resultado final do projecto, por exemplo, através da definição das interfaces de ligação entre os vários sub-sistemas.

Adicionalmente, como não será de estranhar, cada um desses *Implementers* estará essencialmente focado na concretização das tarefas de que foi incumbido, desconhecendo (ou pelo menos não estando demasiadamente preocupado) com as tarefas a cargo dos restantes *Implementers*. Portanto, é vital que alguém defina inicialmente as datas críticas do projecto, defina as várias versões a disponibilizar de cada sub-sistema e elabore um plano de integração das mesmas que permita ao *Project Manager* informar o cliente sobre quando estará disponível cada uma das funcionalidades esperadas. Para além disso, espera-se deste interveniente que, ao longo do desenrolar do projecto, acompanhe de perto a actividade de cada *Implementer*, no sentido de assegurar que as mesmas são cumpridas ou que, na impossibilidade, do seu cumprimento, são tomadas medidas para minimizar o impacto resultante (por exemplo, alertando os *Implementers* que delas se encontram dependentes).

Também para o bom desempenho deste papel é importante a existência de formação apropriada, incidindo essencialmente em duas vertentes: (1) são importantes conhecimentos sobre gestão de recursos humanos, que permitam mobilizar os mesmos e rentabilizar trabalho por estes desenvolvido; (2) para que possa desenvolver a sua actividade da forma mais adequada e de acordo com as melhores práticas definidas, é desejável que o interveniente possua formação e prática nas KA (*Knowledge Area*) de *Software Engineering Management* e *Software Engineering Process* na terminologia do SWEBOK [IEEE, 2004].

¹¹⁸ Por exemplo, porque utiliza uma interface disponibilizada por um componente que se encontra a ser desenvolvido por outro *Implementer* (ou vice-versa), porque troca mensagens com um sistema a ser desenvolvido por outro *Implementer* (e necessitam de ser acordados formatos para as mesmas), etc.

***Software Architect* [C1, C2]**

Excluindo desta análise os micro-projectos¹¹⁹, parece indiscutível a mais-valia que a existência de um responsável por definir os alicerces tecnológicos sobre os quais a implementação do projecto deverá assentar ou, quando estes são impostos pelo cliente, estimando os riscos técnicos inerentes e a forma como poderão ser mitigados. Para além disso, deste interveniente espera-se que defina o esqueleto do sistema a ser criado, tipificando os componentes base, definindo as funções e fronteiras de cada um e harmonizando as práticas¹²⁰ de desenvolvimento a utilizar no projecto, aconselhando as *frameworks* a adoptar e delimitando o espaço de intervenção dos *Designers* do projecto.

Para um bom desempenho desta função, afigura-se como relevante a formação base na área da arquitectura e concepção de software, enquadradas na *KA Software Design* do SWEBOK (ver [IEEE, 2004]). Adicionalmente, e dada a evolução frenética da tecnologia que se verifica nos dias de hoje, é crítico que estes intervenientes acompanhem as tendências do mercado e trabalhem constantemente no sentido de saber a cada momento quais as ferramentas e *frameworks* mais apropriadas à prossecução dos objectivos de um determinado projecto.

***Process Engineer* [C1, C2, C3]**

Um processo de desenvolvimento de software no qual participem todos os papéis essenciais identificados nesta secção assume um nível significativo de complexidade que, se não for gerido de forma satisfatória, poderá ter consequências negativas no mesmo, como sejam:

- Desequilíbrio na distribuição de responsabilidades pelos vários papéis, podendo resultar na ineficiência de um ou mais intervenientes, em virtude de um excesso de atribuições;
- Desperdício de recursos resultante de uma falha/omissão na definição das responsabilidades dos vários papéis que, no extremo, pode conduzir a que mais do que um interveniente prossiga actividades concorrentes (e eventualmente incongruentes) entre si;

¹¹⁹ Com uma simplicidade tal que possam ser totalmente concluídos no período de um mês envolvendo um máximo de duas pessoas.

¹²⁰ Por exemplo, no que concerne a língua, normalização de códigos de erro, procedimentos de *code review*, etc.

- Existência de áreas de intervenção não cobertas pelo processo, em virtude de omissões na atribuição de responsabilidades.

Pelo exposto, considera-se essencial a existência de um interveniente especialmente preocupado com a gestão do processo de desenvolvimento, sua adaptação à realidade da organização e monitorização da sua aplicação, no sentido de detectar e concretizar possíveis melhorias ao mesmo. Por outro lado, o desempenho deste papel requiere um conhecimento aprofundado do processo de desenvolvimento em utilização (neste caso o RUP), o que não é essencial para os restantes papéis. Finalmente, é importante promover a independência deste papel em relação aos restantes, no sentido de assegurar a legitimidade de que necessita para efectuar as adaptações e correções que se impuserem.

Project Manager [C1, C2]

De todos os papéis elencados nesta tabela, este é o que assume maior importância e centralidade, dado que dele se espera que possua uma visão global do projecto e que, através da sua interacção com os participantes internos e externos, crie as condições necessárias ao sucesso do mesmo. Para além disso, deve zelar pelo cumprimento atempado de todos compromissos assumidos e pela gestão do âmbito do projecto, explorando as oportunidades de o aumentar em benefício da organização.

Contudo, o bom desempenho de tão importante função carece de formação apropriada em várias áreas: (1) para além de conhecimentos básicos de gestão, é desejável que conheçam o negócio do cliente (ou, em alternativa, que a equipa seja composta por pelo menos um *System Analyst* que o conheça), no sentido de facilitar o seu entendimento sobre as reais motivações e pertinência dos requisitos apresentados por este; (2) para que possa desenvolver a sua actividade da forma mais adequada e de acordo com as melhores práticas definidas, é desejável que o interveniente possua formação e prática de utilização de uma metodologia de gestão de projectos (por exemplo, PMBOK); (3) são também importantes competências de negociação e um forte sentido de organização.

Project Reviewer [C3]

Não se pode dizer que as incumbências deste papel são críticas para assegurar o sucesso do projecto. Apesar de não parecerem existir bases para sustentar que o seu desempenho carece de competências específicas ou razoavelmente diferentes dos restantes papéis, é especialmente importante que o responsável pelo mesmo possua um conhecimento aprofundado da área de negócio em que o projecto é desenvolvido.

Contudo, dado recaírem sobre este papel responsabilidades sobre a verificação e aprovação de vários artefactos (planeamentos, documentos, resultados obtidos, etc.) produzidos pelos outros intervenientes, resultam como óbvios os conflitos de interesse que desaconselham a junção deste papel a qualquer outro, de forma a garantir a sua independência em relação aos elementos que produzem os vários artefactos.

***Test Manager* [C1, C3]**

A qualidade é um dos vectores de um processo de desenvolvimento de software que carece de gestão cuidada, essencial para:

- Reduzir a probabilidade de ocorrência de *bugs* aplicativos;
- Evitar incumprimentos contratuais (de prazo ou âmbito) e eventuais penalizações associadas;
- Minimizar a utilização das linhas de suporte a clientes;
- Melhorar a imagem do produtor de software.

Assim, dado que o sucesso do processo de desenvolvimento depende em grande medida da qualidade do(s) produto(s) resultante(s), torna-se clara a mais-valia de existir um interveniente cuja função primordial consista em zelar pela mesma. Nesse sentido, quando a qualidade é objectivo, a acumulação deste papel com qualquer outro de elevada importância ou nível de actividade pode tornar-se extremamente difícil, pelo menos sem lesar a eficácia de qualquer um dos papéis.

Para além disso, existe um conflito de interesses latente que obsta, *de per se*, a acumulação deste papel com qualquer a maior parte dos restantes papéis¹²¹ essenciais: se cumpre ao *Test Manager* verificar a qualidade¹²² dos produtos que resultam do processo de desenvolvimento, não fará muito sentido que tenha (tido) responsabilidades de relevo na concepção e construção dos mesmos, sob pena de agir como “*juiz em causa própria*”. A verificar-se, tal acumulação poderia conduzir a um enviesamento dos resultados, resultante da omissão (voluntária ou não¹²³) de testes relevantes para ajuizar sobre a real qualidade dos produtos em causa.

¹²¹ Nomeadamente *Software Architect*, *System Integrator* e *Implementer*.

¹²² Seja esta expressa em correcção, fiabilidade, robustez, desempenho, facilidade de utilização, verificação, reparação, evolução, reutilização, portabilidade, interoperabilidade, etc.

¹²³ Dado ser frequente verificar a existência de vícios nos testes realizados por quem implementa determinado produto ou

***System Tester* [C1, C3]**

Como já foi referido, e no sentido de zelar pela qualidade dos artefactos produzidos pela organização, o *Test Manager* é responsável por conceber um plano interno de auditoria de qualidade, cuja execução também lhe compete coordenar. Contudo, a operacionalização do referido plano é alicerçada na contribuição dos *System Testers* que, de acordo com o seu perfil, poderão ser incumbidos de tarefas muito diversas, que vão desde a revisão de artefactos documentais até à realização de testes ao comportamento dos artefactos aplicativos. Pelo exposto, a eficácia do processo interno de auditoria de qualidade depende inexoravelmente da participação do(s) *System Tester(s)*, pelo que é considerado essencial ao processo.

***Course Developer* [C2]**

Dadas as suas incumbências relacionadas com a preparação e coordenação das acções de formação relacionadas com o(s) produto(s) desenvolvido(s), afiguram-se como essenciais competências em áreas como a didáctica e a pedagogia que, à partida, não são tão relevantes para o bom desempenho dos restantes papéis.

***System Administrator* [C1, C2]**

Qualquer projecto de desenvolvimento de software apresenta dependências, maiores ou menores, das TI (Tecnologias da Informação) necessárias à sua concretização. Logo, num projecto de desenvolvimento de software é extremamente importante a existência de um interveniente focado em assegurar a satisfação das necessidades de infra-estrutura¹²⁴ inerentes ao processo, nomeadamente no que concerne a:

- Computadores pessoais dos elementos da equipa de projecto, de acordo com as necessidades específicas de cada um¹²⁵;
- Servidores que suportam a actividade¹²⁶ da equipa de projecto;
- Servidores que suportam os processos de testes e qualidade¹²⁷;

funcionalidade.

¹²⁴ Consubstanciada no hardware e software de base.

¹²⁵ Os *Developers* provavelmente terão necessidade de uma ferramenta IDE (*Integrated Development Environment*), o *Architect* de uma ferramenta de modelação UML, o *Project Manager* de uma ferramenta de planeamento, o *System Analyst* de uma ferramenta de gestão de requisitos, etc.

¹²⁶ Seja esta de desenvolvimento (para o qual podem ser necessários servidores aplicativos, de base de dados, etc) ou de suporte ao mesmo (aplicações de gestão de projectos, gestão documental, etc).

- Servidores que suportam a disponibilização do serviço para o exterior¹²⁸.

Adicionalmente, compete também a este interveniente a definição dos processos a observar nas operações de passagem a certificação/produção. Pelo exposto, não restam dúvidas sobre a importância do bom desempenho desta função, o que contribui para considerar este papel como essencial. Para além disso, a formação deste tipo de intervenientes não pode ser descurada, sendo desejável que possua (pelo menos) as seguintes competências:

- Prática na KA de *Software Configuration* na terminologia do SWEBOK (ver [IEEE, 2004]);
- Administração de sistemas;
- Configuração e optimização de motores de bases de dados;
- Negociação e contratualização de serviços TI (incluindo definição de SLA).

Desta forma, seria possível reduzir para treze a dimensão do elenco necessário à aplicação do processo (em vez dos trinta e nove papéis originais propostos pelo RUP), no sentido de o tornar compatível com o quadro de recursos humanos de uma PME nacional do sector de desenvolvimento de software.

Contudo, o facto de qualquer um dos restantes (vinte e seis) papéis não ter sido considerado como essencial ao processo não significa que se pretenda descartar as suas atribuições ou que este não seja considerado como importante para a eficaz e eficiente aplicação do mesmo. Em vez disso, propõe-se a concretização de um mapeamento dos restantes papéis para um dos que anteriormente foram considerados como essenciais, de acordo com as seguintes directrizes:

1. Os perfis apropriados para os executantes de ambos os papéis devem ser facilmente compatíveis;
2. As atribuições de ambos os papéis deverão, sempre que possível, encontrar-se enquadradas na mesma (sub-)área de conhecimento;

¹²⁷ Para os quais é frequentemente necessário um ambiente de qualidade e, eventualmente, um ambiente de pré-produção.

3. Caso as atribuições de ambos os papéis se encontrem enquadradas em (sub-)áreas de conhecimento distintas, da sua acumulação por um mesmo indivíduo deverão resultar sinergias positivas;
4. Caso se verifique que o destinatário por eleição de um determinado mapeamento não se encontra nas melhores condições para o acumular, quer por já se encontrar responsável por demasiados papéis ou por papéis que implicam um grande envolvimento, deve tentar-se encontrar um outro interveniente que, apesar de não ser a primeira escolha, reúna melhores condições para assegurar o seu eficaz desempenho.

Nesse sentido, passam a apresentar-se os mapeamentos propostos (entre papéis RUP e papéis considerados essenciais no modelo base), analisando a motivação subjacente aos mesmos.

Business Reviewer (ou Business-Model Reviewer) ⇔ Project Reviewer

O RUP caracteriza o *Project Reviewer* como alguém responsável por avaliar artefactos do projecto em determinados momentos-chave, com poder e legitimidade para, se necessário, suspender a sua execução. Logo, esse papel só poderá ser desempenhado por alguém com um elevado nível de responsabilidade e autoridade no interior da organização, eventualmente até ao mais alto nível, já que não é invulgar que os gestores de uma PME se envolvam pessoalmente na supervisão e acompanhamento dos projectos por esta desenvolvidos. Pelo exposto, o *Project Reviewer* possui o perfil apropriado e, conseqüentemente, estará em condições de avaliar os artefactos de negócio (propostas comerciais, modelos de negócio, etc.) produzidos ao longo do projecto.

Business-Process Analyst ⇔ Project Manager

Dado que o *Project Manager* deverá ser o interveniente com maior proximidade em relação ao cliente, poderá ser da sua competência a definição da arquitectura de negócio deste e a descrição das suas necessidades (na forma de casos de uso de negócio). Dessa forma, assegurar-se-ia que o *Project Manager* conhece detalhadamente o âmbito do projecto, o que lhe permite lidar adequadamente com os pedidos de alteração (e da dimensão comercial que lhe está inerente) e efectuar um eficaz acompanhamento da sua execução.

¹²⁸ Através do usualmente designado ambiente de produção.

Business Designer* ⇔ *System Analyst

Tomando por base as necessidades de negócio identificadas pelo *Business-Process Analyst*, compete ao *Business Designer* detalhar a especificação de negócio, produzindo artefactos que caracterizem as entidades de negócio envolvidas, suas expectativas e interações. Contudo, apesar de focada no negócio, a actividade deste papel encontra-se próxima do *System Analyst*, responsável pela identificação e documentação dos requisitos do projecto. Por isso, considerando que o *System Analyst* também deve conhecer a área de negócio em causa e dada a necessidade de simplificação do processo existente, recomenda-se que este acumule o papel de *Business Designer*.

Requirements Reviewer* ⇔ *Project Reviewer

A metodologia atribui a este papel a responsabilidade de rever formalmente os requisitos identificados e incorporados no modelo de casos de uso pelo *Requirements Specifier*. Para isso, é essencial que este interveniente seja conhecedor da área de negócio em causa. Adicionalmente, deve também encontrar-se familiarizado com as técnicas de modelação¹²⁹ utilizadas na descrição dos requisitos, o que, caso apenas se consubstancie em diagramas de casos de uso e textos em linguagem natural, não é difícil de alcançar. Pelo exposto, e à semelhança do que se defende para o *Business Reviewer*, considera-se que este papel pode ser acumulado pelo *Project Reviewer* sem grande prejuízo na eficácia do seu desempenho.

Requirements Specifier* ⇔ *Project Manager

Compete ao *Project Manager* assegurar a gestão do âmbito do projecto, garantindo o cumprimento do âmbito contratado e minimizando as extensões a este que não aportem valor para a organização que representa. Contudo, a eficácia da sua intervenção poderá sofrer caso não possua (por falta de conhecimentos técnicos ou por excessiva delegação em terceiros) um conhecimento aprofundado dos objectivos acordados entre as partes. Pelo exposto, ao chamar a si as funções de *Requirements Specifier*, o *Project Manager* potencia o seu controlo sobre o projecto, responsabilizando-se por elencar e caracterizar cada um dos requisitos (principalmente funcionais) apresentados pelo cliente do projecto. Desse modo, terá mais facilidade não só em gerir adequadamente as expectativas do cliente (clarificando o que está e não está no âmbito do projecto), como também em assegurar que estas são concretizadas pela equipa de projecto.

¹²⁹ Por exemplo, a notação UML.

Use Case Specifier* ⇔ *System Analyst

O papel do *Use Case Specifier* é definido pela metodologia como alguém que, interagindo de perto com os utilizadores finais, colabora com o *System Analyst* na descrição dos casos de uso que dão corpo aos requisitos identificados. Ou seja, este interveniente não é definido como tendo incumbências próprias mas sim como um auxiliar de outro interveniente. Logo, no âmbito de um esforço de simplificação do elenco processual, este papel deve ser acumulado pelo *System Analyst*.

Architecture Reviewer* ⇔ *Process Engineer

Ao contrário do anterior, este papel tem um perfil marcadamente técnico, cabendo-lhe a revisão formal da arquitectura applicacional concebida pelo *Software Architect*, no sentido de validar as suas escolhas. Ainda segundo a metodologia, é importante que este interveniente possua a legitimidade necessária para apontar erros ou lacunas que, apesar de impopulares ou eventualmente incompatíveis com o planeamento do projecto, devam ser identificados no melhor interesse do sucesso deste. Para tal, para além das óbvias competências técnicas, é importante possuir boas competências de comunicação, que lhe permitam gerir eventuais conflitos com a sensibilidade e tacto necessários. Vale a pena considerar ainda que:

- o *Project Manager* não possui os conhecimentos técnicos necessários ao desempenho deste papel;
- o *Integrator* provavelmente não usufruirá de um estatuto no interior da organização que lhe permita total liberdade na avaliação do trabalho desenvolvido pelo *Software Architect*;
- o *Process Engineer*, pela natureza da sua função, possui a legitimidade necessária para avaliar o desempenho de todos os intervenientes no processo de desenvolvimento de software, sobre o qual deverá possuir uma vasta experiência.

Dito isto, defende-se que o *Process Engineer* será o interveniente em melhores condições para acumular as responsabilidades do *Architecture Reviewer*.

***Capsule Designer* ⇔ *Integrator* (ou *System Integrator*)**

A metodologia diz-nos que o *Capsule Designer* possui um perfil semelhante ao *Designer* mas se encontra mais focado em assegurar que o(s) sistema(s) a implementar satisfaz(em) adequadamente os requisitos de desempenho e concorrência típicos de aplicações que mantêm estado e respondem a eventos. Obviamente, a redução do leque de

participantes no processo de desenvolvimento de software resultará inevitavelmente na necessidade de melhoria das competências e desempenhos individuais já que, em virtude das acumulações de papéis, cada novo interveniente necessita de fazer mais e melhor. Assim, e dada a semelhança dos seus perfis, parece natural que seja o *Integrator* a assumir este papel (à semelhança do que se recomenda para o *Designer*).

***Code Reviewer* ⇒ *Integrator* (ou *System Integrator*)**

Dada a função de coordenação de uma linha de desenvolvimento que um determinado *Integrator* desempenha, e para a qual é essencial a existência de um bom nível de competências técnicas, considera-se natural que recaia sobre ele a responsabilidade de proceder à revisão e auditoria do código fonte produzido pelos *Implementers* envolvidos. Dessa forma, poderá verificar se cada componente foi implementado de acordo com as suas instruções e detectar potenciais problemas (por exemplo, de concorrência).

Poder-se-á questionar se não faria mais sentido que esta função fosse assumida pelo *Test Manager/System Tester*, responsáveis por garantir níveis adequados de qualidade dos artefactos produzidos. Contudo, após pesar os prós e contras de tal situação, optou-se por não o fazer por se considerar que para além de já ser difícil para uma PME disponibilizar os recursos necessários para estes papéis, nem sempre os envolvidos nestas actividades são suficientemente experientes para poderem efectuar um trabalho de auditoria com a profundidade desejada.

Component Engineer* ⇒ *Implementer

Considerando que as interfaces entre os vários sub-sistemas são gizadas pelo *Designer*, a intervenção do *Component Engineer* encontrar-se-á focada na estruturação interna de cada sub-sistema, nomeadamente no que diz respeito às operações, métodos, atributos, relações e requisitos de cada uma das classes de concepção respectivas. Como quando há menos envolvidos numa tarefa, cada um necessita de fazer mais, defende-se por isso que as incumbências do *Component Engineer* devem ser passadas para os *Implementers* a trabalhar no projecto que, contando com o acompanhamento necessário por parte do mais experiente *Designer/Integrator*, deixam de ser meros executantes para dessa forma participar no esforço de concepção da realidade a concretizar.

***Designer* ⇒ *Integrator* (ou *System Integrator*)**

É função do *Designer* traduzir a arquitectura concebida pelo *Software Architect* numa concepção coerente dos componentes/módulos/sub-sistemas a implementar, detalhando

responsabilidades, operações e relações entre estes. Logo, é possível concluir que tal actividade lhe dará uma visão abrangente de todo o sistema e um conhecimento aprofundado da forma como as partes interagem entre si. Assim, considerando que compete ao *Integrator* assegurar uma integração bem sucedida dos vários componentes existentes, faz sentido que ambos os papéis sejam desempenhados pela mesma pessoa, rentabilizando, dessa forma, o conhecimento gerado na fase de concepção.

Design Reviewer* ⇒ *Software Architect

A metodologia denota uma forte defesa da existência de papéis especialmente dedicados à revisão de artefactos criados por terceiros. Por isso, não surpreende a existência deste interveniente, responsável pela revisão da concepção do(s) sistema(s) a desenvolver, elaborada pelo *Designer*. Assim, no sentido de manter a independência entre autor e avaliador dos artefactos produzidos, propõe-se que este papel seja acumulado pelo *Software Architect* que, pela natureza das suas funções e pela experiência acumulada que expectavelmente possuirá, deverá ter todas as condições para o seu bom exercício.

Use Case Engineer* ⇒ *System Analyst

Tal como se encontra definido pela metodologia, este interveniente é responsável por garantir que uma ou mais concretizações de casos de uso representam, de forma coerente e integral, os requisitos estabelecidos para o projecto. Por isso, e dado o papel que o *System Analyst* desempenha no levantamento destes requisitos, considera-se que a acumulação por parte deste das actividades do *Use Case Engineer* será como que uma extensão natural do seu trabalho.

Change Control Manager* ⇒ *Project Manager

A metodologia indica que, em projectos de reduzida dimensão, é normal que este papel seja assumido pelo *Project Manager* ou pelo *Software Architect*. Contudo, apesar de ser obviamente essencial que o *Software Architect* possua um conhecimento profundo e actualizado sobre o âmbito do projecto, considera-se ser ainda mais importante que o *Project Manager* possa assegurar um eficaz controlo sobre o mesmo, sob pena de se criarem no cliente expectativas irrealistas ou de se proceder a desenvolvimentos sem retorno comercial. Note-se que, embora não seja obrigatório que todos os aumentos de âmbito sejam total ou parcialmente custeados pelo cliente, é essencial que a realidade resulte de uma decisão

consciente do *Project Manager* e não seja condicionada por uma inabilidade¹³⁰ comercial na condução do processo, mais propícia quando o diálogo com o cliente é conduzido por elementos com um perfil mais técnico. Pelo exposto, e à semelhança do que foi proposto anteriormente para o *Business-Process Analyst*, deverá ser o *Project Manager* a acumular este papel, recorrendo sempre que necessário ao apoio mais tecnológico do *Software Architect*.

Configuration Manager* ⇒ *System Administrator

Propõe-se adicionar as responsabilidades deste papel às incumbências próprias do *System Administrator*, já que parece uma extensão natural da intervenção deste. Ou seja, se o *System Administrator* é já responsável pela gestão e disponibilização da infra-estrutura utilizada pela equipa de projecto, então faz sentido que também assegure as necessidades específicas da gestão de configurações. Adicionalmente, o facto de o *Configuration Manager* se encontrar também incumbido de reportar sobre a evolução dos pedidos de alteração reforça a racionalidade desta acumulação, já que compete ao *System Administrator* gerir os ambientes de qualidade e produção e, conseqüentemente, fazer repercutir nestes as alterações solicitadas.

Deployment Manager* ⇒ *Project Manager

Espera-se deste interveniente que planeie e gira de forma eficaz a transição para a comunidade de utilizadores dos produtos resultantes do esforço de desenvolvimento. O sucesso deste tipo de actividade irá depender em grande medida do diálogo mantido com o cliente e das competências de planeamento e comunicação do indivíduo envolvido. Para além disso, e segundo a própria metodologia, este interveniente deve trabalhar em grande proximidade com o *Project Manager* ou até eventualmente ambos os papéis serem executados pela mesma pessoa. Pelo exposto, dado que a abordagem seguida pretende potenciar a centralidade e controlo do *Project Manager*, defende-se que se vá um pouco mais longe do que a metodologia propõe, fundindo num só estes dois papéis.

Management Reviewer* ⇒ *Project Reviewer

Apesar do seu perfil mais ligado ao negócio e à gestão, a linha que separa o *Management Reviewer* do *Project Reviewer* é muito ténue, já que ambos são definidos como responsáveis pela revisão e a avaliação dos artefactos produzidos em determinados

¹³⁰ Por exemplo, não esclarecendo à partida que determinada funcionalidade não se encontrava incluída no âmbito inicial do projecto, pelo que poderia implicar encargos adicionais.

momentos-chave. Assim, e assumindo como já foi referido anteriormente que o *Project Reviewer* será alguém com alguns conhecimentos nesta área, considera-se que esta separação não se justifica, pelo que se propõe a sua junção.

***Integration Tester* ⇒ *Integrator* (ou *System Integrator*)**

Considerando que a principal responsabilidade deste papel consiste na realização dos testes de integração, indispensáveis à verificação que os vários componentes que compõem a versão funcionam correctamente em conjunto, a sua actividade pode considerar-se como uma extensão natural da levada a cabo pelo *Integrator*. Para além disso, é a própria metodologia que, no sentido de minimizar a necessidade de replicar conhecimento comum por vários intervenientes, sugere a possibilidade deste papel ser desempenhado pelo *Integrator*. De facto, essa sugestão faz sentido não só pela redução de complexidade obtida, como também por permitir ao *Integrator* possuir um maior grau de confiança nas versões que entrega para testes de sistema, devido à possibilidade que tem de detectar prematuramente possíveis erros de integração.

Test Analyst* ⇒ *Project Manager

Para que possa ser verdadeiramente eficaz, o esforço de teste necessita de estar alinhado com o contexto do projecto e com as necessidades dos utilizadores finais do mesmo. Assim, dado que o *Project Manager* é o interveniente com maior conhecimento sobre o âmbito do projecto e sobre a forma como os artefactos irão ser usados, deverá ser ele a definir (em diálogo com o *Integrator* e o *Test Designer*) os cenários que deverão ser alvo de teste. Para além disso, também parece fazer todo o sentido que seja ele o responsável por:

- monitorizar o desenrolar do processo de teste, assegurando a sua evolução ao ritmo necessário;
- analisar os resultados do esforço de teste produzidos pelo *Test Designer* e agir em conformidade, certificando-se que são reportados aos respectivos *Integrators* e que estes asseguram a sua atempada correcção;
- avaliar a eficácia do processo de teste através da qualidade percebida e reportada pelos utilizadores finais;
- ser o elemento facilitador da comunicação entre o *Test Designer* e o *Integrator*.

Test Designer* ⇒ *Test Manager

Da actividade levada a cabo pelo *Test Analyst* emanam um conjunto de cenários a testar, constituindo o ponto de partida para o trabalho do *Test Designer*, incumbido de coordenar o planeamento, concepção e implementação dos testes necessários. Contudo, estas actividades constituem um corolário das responsabilidades que a literatura atribui ao *Test Manager*, descrito como o responsável geral pelo sucesso do esforço de testes. Assim sendo, deve ser este interveniente a assumir as funções de *Test Designer*.

Graphic Artist* ⇒ *User-Interface Designer

O desempenho deste papel beneficia de uma apurada sensibilidade estética e de alguma experiência na utilização de aplicações de manipulação de imagem. Contudo, dado ser vulgar que estas características também se encontrem presentes no *User-Interface Designer*, poderá ser ele o responsável pela satisfação das necessidades de imagem e comunicação gráfica inerentes ao projecto.

Review Coordinator* ⇒ *Project Manager

Apesar de existirem responsáveis pela produção da documentação e pela respectiva revisão, é ainda desejável a participação de um interveniente com a incumbência de assegurar que as revisões são efectuadas atempadamente. Assim, e dado os papéis de revisão que foram alvo de simplificação (*Requirements Reviewer*, *Architecture Reviewer*, *Code Reviewer*, *Design Reviewer* e *Management Reviewer*) foram atribuídas ao *Project Manager*, considera-se ser ele o melhor colocado para assegurar a coordenação e controlo inerentes a este papel.

Technical Writer* ⇒ *Course Developer

A existência deste papel na metodologia parece traduzir o facto de ser desejável que a produção dos conteúdos de suporte (manuais de utilizador, etc.) possa ser executada por elementos distintos (eventualmente com conhecimentos técnicos mais limitados) dos que se encontram envolvidos na implementação do projecto. Dessa forma, seria possível concentrar o esforço dos *Implementers* nas actividades de desenvolvimento e possivelmente reduzir o custo global de desenvolvimento, ao utilizar recursos com um menor custo para as usualmente morosas actividades de documentação. Contudo, dadas as restrições (já referidas anteriormente) habitualmente verificadas em organizações de menor dimensão, será difícil encontrar nas organizações alguém que desempenhe este papel de forma exclusiva. Contudo, dado que as actividades do *Course Developer* podem ser consideradas como complementares

às do *Technical Writer* (o material de formação que produz é também direccionado para utilizadores finais), recomenda-se a fusão destes papéis.

Tool Specialist* ⇔ *Process Engineer

O *Process Engineer* é um interveniente de suporte à metodologia, responsável por monitorizar a sua aplicação e efectuar os ajustes necessários à optimização da sua eficácia. Assim, considera-se que faz sentido estender a sua intervenção para também incluir a identificação das necessidades dos vários intervenientes em relação a ferramentas que auxiliem/facilitem o seu trabalho, seleccionando as aplicações mais apropriadas para as satisfazer e assegurando que são atempadamente colocadas à disposição destes. Findo o projecto, essa acumulação de funções permitirá ao *Process Engineer* avaliar em que medida uma possível ineficácia da aplicação da metodologia se deve ou não à ausência das ferramentas apropriadas para suportar a actividade dos intervenientes, no sentido de evitar chegar a conclusões erradas.

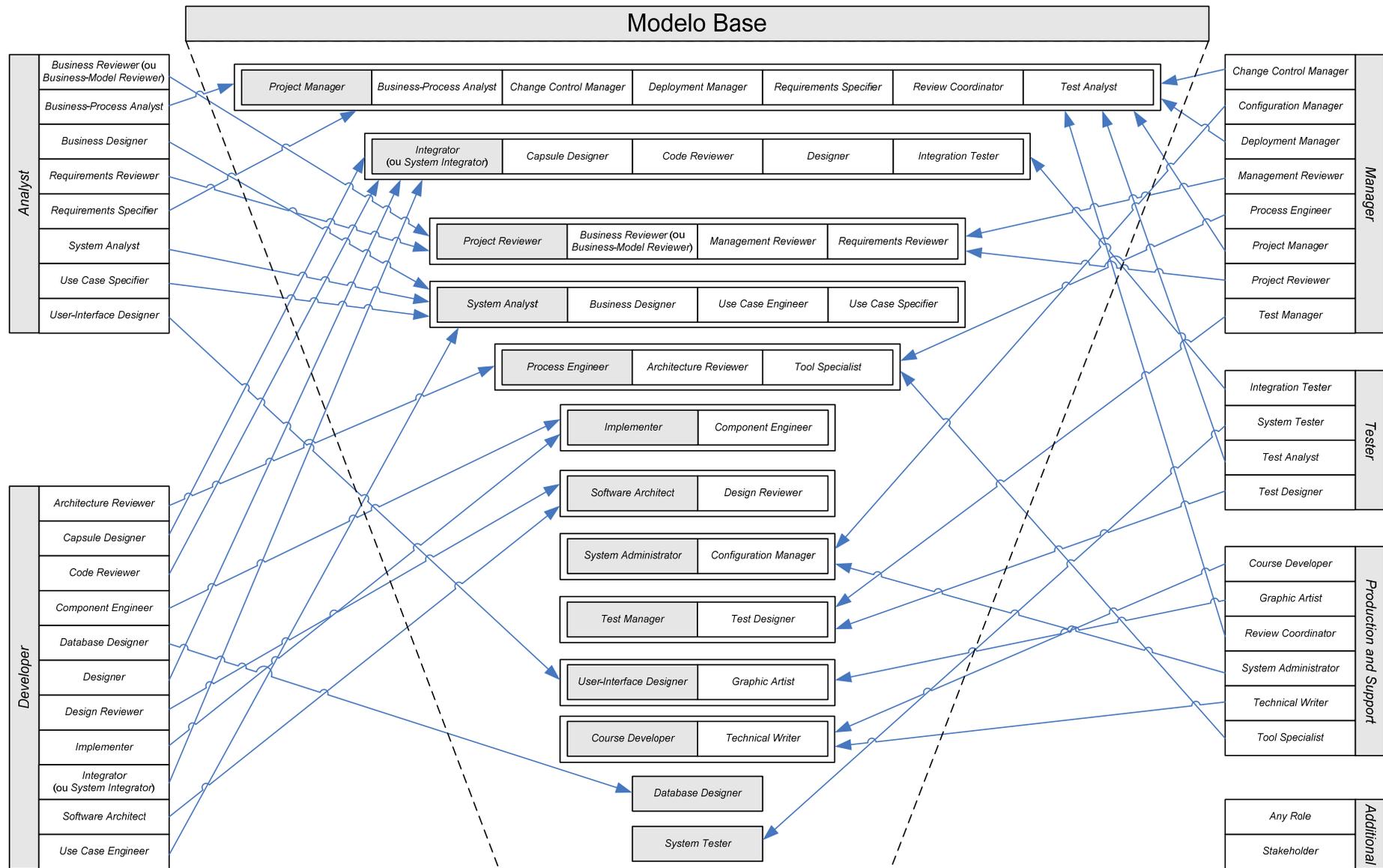


Figura 3.1 – Mapeamentos do Modelo Base

Encontra-se assim apresentado o modelo base proposto (ver Figura 3.1), que assenta na participação de treze intervenientes distintos. Porém, para facilitar ainda mais a sua adopção por uma organização PME nacional é aconselhável proceder a adaptação das denominações dos papéis essenciais resultantes já que:

- As designações originais de alguns papéis (por exemplo, o *Process Engineer*) são algo estranhas para alguém que não se encontre familiarizado com o RUP;
- Após a concretização das acumulações de funções propostas, a designação de alguns papéis considerados essenciais (por exemplo, o *Integrator*) deixa de fazer sentido por já não reflectir correcta e totalmente o âmbito da sua intervenção;
- A caracterização mais aprofundada de cada papel que irá explanar na secção 4.2 não se irá restringir às incumbências previstas pelo RUP, dado este não mencionar algumas funções que também parecem importantes (por exemplo, preparação de actividades de suporte, focalização na qualidade de todos os artefactos e não apenas nos testes aplicacionais, desenvolvimento de estratégias de influência externa, etc.). Logo, as denominações encontradas deverão ser suficientemente abrangentes para permitir a incorporação das funções adicionais relevantes;
- A implementação de um processo de desenvolvimento de software não envolve apenas os elementos da equipa técnica, devendo comprometer intervenientes de vários quadrantes da organização, alguns em papéis de grande relevância (por exemplo: *Project Manager*, *Project Reviewer*, etc.). Assim, dado que o domínio destes sobre a língua inglesa poderá não ser tão aprofundado quanto o usualmente verificado pelos profissionais da área técnica¹³¹, é importante fazer um esforço para encontrar denominações apropriadas em português.

Pelo exposto, propõe-se a adopção da terminologia apresentada na Tabela 3.1.

¹³¹ Dadas as suas habituais necessidades de consultar informação e documentação nessa língua.

Tabela 3.1 – Denominações do Elenco Processual Proposto

Papel Essencial no Modelo Base	Denominação Portuguesa	Sigla Unívoca
<i>Project Manager</i>	Gestor de Projecto	GPI
<i>Integrator (ou System Integrator)</i>	Chefe de Equipa	CEI
<i>Project Reviewer</i>	Patrocinador de Projecto	PPI
<i>System Analyst</i>	Analista	ANI
<i>Process Engineer</i>	Coordenador de Desenvolvimento	CDI
<i>Implementer</i>	Programador	PRI
<i>Software Architect</i>	Arquitecto de Software	ASI
<i>System Administrator</i>	Coordenador de Infra-Estrutura	CII
<i>Test Manager</i>	Coordenador de Qualidade	CQI
<i>User-Interface Designer</i>	Coordenador de Comunicação e Imagem	CCI
<i>Course Developer</i>	Coordenador de Suporte	CSI
<i>Database Designer</i>	Administrador de Bases de Dados	ABI
<i>System Tester</i>	Auditor de Qualidade	AQI

A tabela anterior propõem não só novas designações para cada um dos papéis essenciais, como também siglas que poderão ser utilizadas para referenciar de forma simples e inequívoca cada um deles. Note-se que o facto de cada uma dessas siglas terminar com o carácter “I” se deve ao facto da sua intervenção decorrer no interior da organização, já que na secção 4.2 se irão também abordar os fluxos de comunicação existentes entre os intervenientes internos e os seus homólogos externos. É também conveniente referir que o facto de se apelidarem alguns dos intervenientes de “*Coordenador*” não significa que seja expectável (à excepção do recomendado para o CDI na secção 3.4) que o indivíduo nomeado para o papel corresponda ao responsável funcional (caso exista) pela área em questão. Em vez disso, esse termo pretende transmitir que esses intervenientes serão responsáveis por assegurar, pessoalmente ou envolvendo terceiros, a satisfação das necessidades do projecto em relação às respectivas áreas funcionais, agindo como elos de ligação privilegiados com as mesmas.

Conclui-se assim a apresentação do modelo base, considerando-se que a redução de complexidade atingida na sequência da diminuição do número de intervenientes no processo

de desenvolvimento de software é suficiente para assegurar a aplicabilidade deste modelo a uma organização de média dimensão. Contudo, apesar do progresso conseguido, é possível que este modelo ainda possa ser considerado como demasiadamente complexo para ser passível de aplicação em instituições de pequena dimensão, motivo pelo qual se irá apresentar um modelo alternativo na secção que se segue.

3.2.2. Modelo Reduzido

Apesar do esforço de simplificação do elenco processual já realizado, é provável que o modelo base ainda se afigure como de difícil aplicação a organizações de micro ou pequena¹³² dimensão, pelo que se considera pertinente procurar um modelo que envolva um menor número de intervenientes, abdicando de alguma especialização dos mesmos e promovendo a sua polivalência. Contudo, para o conseguir não basta eliminar alguns papéis, distribuindo as suas responsabilidades pelos intervenientes remanescentes, dado que isso poderia colocar em causa o equilíbrio conseguido no modelo base, fazendo com que a aplicação falhasse devido à incapacidade de um ou mais intervenientes desempenharem com eficácia as suas incumbências. Em vez disso, propõe-se dar continuidade ao processo de simplificação de acordo com as seguintes orientações: (1) identificando quais dos papéis anteriormente considerados como essenciais podem revestir-se de menor proeminência quando comparados com os restantes; (2) identificando o papel em melhores condições para assumir as responsabilidades de cada interveniente excluído, considerando o seu perfil; (3) validando se os mapeamentos propostos não aumentam de forma exagerada a área de intervenção dos papéis destino, no sentido de assegurar que estes possuem condições efectivas para assumir responsabilmente as responsabilidades inerentes às várias funções que devem desempenhar.

Contudo, o leitor deverá estar consciente de que a eventual maior facilidade de aplicação do modelo reduzido, quando comparado com o modelo base, é usualmente custeada por uma redução de qualidade dos artefactos produzidos e/ou por um maior custo incorrido na sua produção (em virtude da menor experiência/especialização dos envolvidos). Contudo, estas desvantagens poderão ser consideradas como um mal menor em contextos organizacionais em que a única alternativa seria a utilização de um processo muito mais *ad-hoc*, sem formalização de papéis nem responsabilidades que resulta frequentemente num maior desperdício de recursos e incongruência de acções.

Dito isto, passemos à análise detalhada dos papéis que, apesar de considerados como

¹³² Caso, segundo [IAPMEI, 2007], possua menos de 10 ou 50 efectivos, respectivamente.

essenciais no modelo base, poderão ser suprimidos no sentido de reduzir a complexidade e minimizar o número de intervenientes no processo de desenvolvimento.

System Analyst* ⇔ *Project Manager

De acordo com a metodologia, compete ao *System Analyst* a responsabilidade específica de coordenar o processo de levantamento de requisitos, de modo a delimitar o âmbito do projecto de forma tão inequívoca quanto possível. Contudo, a sua intervenção estaria sujeita ao acompanhamento e coordenação do *Project Manager* que, enquanto o interveniente mais próximo do cliente, necessita de estar constantemente informado sobre o decorrer das trabalhos. Para além disso, em contextos de PME's é vulgar que o papel de *Project Manager* seja entregue a um indivíduo com formação em eng^a de software (e possivelmente até em eng^a de requisitos) a quem, fruto da sua experiência, são dadas oportunidades de evoluir profissionalmente para posições de maior responsabilidade. Assim, considera-se que o *System Analyst* poderá ser suprimido à custa de um maior envolvimento do *Project Manager* no processo de levantamento de requisitos, assumindo que neste podem ser encontradas as valências técnicas mínimas para o desempenho desta função.

User-Interface Designer* ⇔ *Implementer

Apesar de indiscutivelmente muito importantes para o sucesso de um projecto, a atractividade e usabilidade das interfaces de utilizador implementadas são questões que, quando comparadas com outras¹³³, se revestem de uma menor prioridade, razão pela qual este papel é um candidato natural à supressão. Assim sendo, propõe-se que seja cada *Implementer* a assumir este papel, ficando obviamente o resultado final bastante mais dependente da sensibilidade para estas questões do(s) indivíduo(s) em causa, do que uma formação sólida destes sobre estas temáticas.

Software Architect* ⇔ *Integrator

Apesar de cíclico, o envolvimento deste interveniente no processo de desenvolvimento é bastante mais significativo na sua parte inicial, nomeadamente nas fases de esboço e detalhe o que, em organizações de menor dimensão, dificulta a afirmação deste tipo de profissionais (pelo menos em exclusividade de funções) devido a serem difíceis de rentabilizar¹³⁴ em fases

¹³³ Gestão de âmbito, gestão de tempo, gestão de risco, etc.

¹³⁴ Usualmente, uma PME não pode, por falta de recursos, aceitar muitos projectos em simultâneo. Assim, mesmo que a organização tivesse *Software Architects* disponíveis para alocar a um novo projecto, poderia não ter os restantes intervenientes (*Integrators*, *Implementers*, etc), conduzindo a um desperdício da sua capacidade instalada.

subsequentes do projecto. Para além disso, não só usualmente a dimensão e/ou complexidade dos projectos em que este tipo de organizações está envolvida também não o justificam, como também não é possível proporcionar-lhes os recursos necessários (tempo para investigação, formação, infra-estrutura, etc.) para que estes possam efectivamente acompanhar o surgimento das evoluções tecnológicas relevantes. Assim sendo, considera-se que a manutenção deste papel seria artificial, traduzindo-se inevitavelmente no seu descuro aquando da operacionalização da metodologia. Pelo exposto, propõe-se que seja o *Integrator* a assumir este papel sempre que necessário, não só porque também ele partilha da necessidade de actualização¹³⁵ tecnológica permanente, mas também porque provavelmente os indivíduos escolhidos para esta função serão dos tecnicamente mais experientes da organização. Assim, cada *Integrator* envolvido no projecto ficaria encarregue de architecturar as aplicações sob a sua responsabilidade. Contudo, e como não poderia deixar de ser, há vantagens associadas a esta simplificação, como sejam:

- possível perda de coerência das actividades de arquitectura de software conduzidas quer em projectos distintos, quer por diferentes *Integrators* dentro do mesmo projecto, devido à ausência de um interveniente exterior à equipa de projecto que sirva de referencial e auxilie cada *Integrator* a encontrar as melhores soluções para os problemas tecnológicos que necessita de endereçar;
- possível redução da capacidade de aprendizagem organizacional e, consequentemente, do potencial inovador da organização.

Database Designer* ⇔ *Integrator

A generalidade das licenciaturas nacionais na área da engenharia informática contemplam a formação (mais ou menos avançada) de índole técnica sobre a concepção e representação de modelos de base de dados, apesar de usualmente alguns conceitos mais avançados (*triggers*, *stored procedures*, etc.) apenas serem abordados no contexto de um ou dois motores de base de dados específicos. Todavia, a formação universitária habitualmente não aborda tópicos mais avançados (como sejam a administração, estratégias de *backup* e recuperação de dados, optimização do motor de base de dados) pelo que as organizações que pretendem possuir internamente este tipo de conhecimento se vêm usualmente na necessidade de recorrer a formação especializada e normalmente associada a um determinado

¹³⁵ Neste tipo de organizações a actualização de conhecimentos resulta normalmente mais de processos de investigação

motor de base de dados específico. Contudo, considera-se que a generalidade dos profissionais dessa área do conhecimento possuirão os conhecimentos mínimos necessários para desempenhar esta função, pelo que, no âmbito do esforço de simplificação em curso, este papel poderia ser assumido pelo *Integrator* ou pelo *Implementer*. No entanto, o facto de o *Integrator* possuir uma visão mais abrangente do projecto quando comparado com um *Implementer* encarregue de uma tarefa específica, poderá proporcionar-lhe a sensibilidade necessária para desenhar um modelo de dados que contemple não só a totalidade das necessidades actuais como também as evoluções mais prováveis num futuro próximo. Assim sendo, propõe-se que seja ele a acumular as incumbências do *Database Designer*, de cuja autonomia se prescindir neste modelo reduzido.

Course Developer* ⇒ *Integrator

À semelhança do que já foi referido no papel anterior, não restam dúvidas de que a qualidade dos materiais de apoio à formação de utilizadores é extremamente importante como elemento facilitador do processo de mudança inerente à adopção de uma nova aplicação informática. Porém, antes disso é necessário assegurar que a aplicação em causa é desenvolvida atempadamente e de acordo com os requisitos estipulados, objectivos nos quais têm de ser concentradas a maior das atenções. Assim sendo, considera-se que dado o pragmatismo que tem que imperar em organizações de pequena dimensão, não se justifica a manutenção autónoma deste papel já que, apesar de provavelmente não possuir formação nas áreas didáctica e pedagógica, o *Integrator* deverá possuir todas as condições para produzir conteúdos que, do ponto de vista técnico, permitam elucidar os utilizadores finais sobre o funcionamento e utilização da(s) nova(s) aplicação(ões). Para além disso, considerando que os conteúdos produzidos deverão ser sujeitos ao escrutínio e aprovação de um outro interveniente, seria possível colmatar as possíveis lacunas comunicacionais antes da exposição dos mesmos aos utilizadores finais. Contudo, e à semelhança do que também foi referido sobre o papel precedente, o leitor deverá ter em mente que, se não forem criados os mecanismos de revisão apropriados, poderão ser observadas variâncias qualitativas bastante significativas nos resultados, motivo pelo qual esse aspecto não deve ser negligenciado.

Desta forma, e de acordo com o exposto na tabela anterior, seria possível reduzir ainda mais a dimensão do elenco processual, reduzindo-o a apenas oito intervenientes distintos. Porém, como foi visto na secção 3.2.1, o modelo base propunha a associação de alguns dos

papéis suprimidos aos papéis essenciais de que agora prescindimos, pelo que se impõe a definição de um novo mapeamento para cada um deles. Para além disso, é necessário não só validar se os novos mapeamentos não sobrecarregam em demasia nenhum dos intervenientes remanescentes, como também assegurar que não colocam em causa a independência que deve existir entre os titulares de alguns papéis. Nos casos em que tal se verificar, serão propostos um ou mais novos mapeamentos destinados a equilibrar as responsabilidades e volume de trabalho associado a cada um dos papéis essenciais.

Assim sendo, defende-se a concretização dos mapeamentos adicionais seguidamente apresentados.

Business Designer* ⇔ *Project Manager

A intervenção do *Business Designer* destina-se a dar alguma profundidade ao trabalho do *Business-Process Analyst*, no sentido de caracterizar adequada e exhaustivamente uma parte da organização cliente, podendo por isso ser considerado um papel de apoio à actividade deste. Pelo exposto, e dado o esforço de redução do elenco processual em curso, considera-se que será aceitável anexar este papel ao *Project Manager*, estendendo de forma natural a sua intervenção uma vez que este também se encontra responsável pela função de *Business-Process Analyst*.

Use Case Specifier* ⇔ *Project Manager

Por motivos semelhantes aos identificados na sub-secção “*Use Case Specifier* ⇔ *System Analyst*” na pág. 81, este papel deverá acompanhar o *System Analyst* no mapeamento para o *Project Manager*.

Designer* ⇔ *Implementer

Da atribuição do papel de *Design Reviewer* ao *Integrator* resulta uma incompatibilidade de funções, em virtude de este já se encontrar responsável pelo papel de *Designer*. Obviamente, esta situação é altamente indesejável, já que concentraria no mesmo interveniente as funções de concepção do(s) sistema(s) a implementar e de avaliação da correcção da mesma, ferindo a independência do processo. Por outro lado, o facto de o modelo reduzido associar ao *Integrator* vários papéis adicionais, faz com que exista uma tendência para a degradação da sua eficácia operacional, como consequência do seu vastíssimo leque de incumbências. Pelo exposto, de forma a sanar/minimizar estes problemas, propõe-se que este papel passe a estar atribuído ao *Implementer*, que deixa de ser mero

executante para participar também na concepção da parte da aplicação pela qual se encontra responsável. Obviamente, como não podia deixar de ser, para que o resultado final não seja significativamente afectado é desejável que os *Implementers* envolvidos possuam a experiência necessária para se sentirem confortáveis com este aumento da sua área de intervenção.

Design Reviewer* ⇒ *Integrator

Com a supressão do *Software Architect*, papel ao qual o *Design Reviewer* se encontrava associado, torna-se necessário encontrar um outro interveniente capaz de assumir esta função. Nesse sentido, considera-se que a solução mais racional passa pelo mapeamento desta para o *Integrator*, à semelhança do que foi proposto para o *Software Architect* e seguindo uma lógica semelhante.

Use Case Engineer* ⇒ *Test Manager

Caso aplicasse a este papel a mesma lógica utilizada no mapeamento do *System Analyst* (ao qual tinha sido anexado no modelo base), deveria ser o *Project Manager* a assumi-lo neste modelo reduzido. Contudo, são estes os motivos que levam a propor que seja antes o *Test Manager* o destinatário deste mapeamento:

- o *Test Manager* deverá reunir todas as condições (técnicas e não só) necessárias ao bom desempenho desta função;
- como consequência do processo de simplificação implícito ao modelo reduzido o *Project Manager* ficará responsável por dez papéis distintos, o que, aliado ao facto de ser o responsável máximo pelo projecto, resulta numa enorme carga. Logo, será mais realista considerar que o *Test Manager* terá melhores condições para desempenhar este papel de uma forma eficaz;
- potencia o conhecimento exaustivo dos requisitos por parte do *Test Manager*, o que não só lhe permitirá preparar mais facilmente o plano de testes, como também possibilitará que auxilie o *Project Manager* no controlo da sua concretização.

Graphic Artist* ⇒ *Implementer

Por motivos semelhantes aos identificados na sub-secção “*Graphic Artist* ⇒ *User-Interface Designer*” na pág. 86, este papel deverá acompanhar o *User-Interface Designer* no mapeamento para o *Implementer*.

Technical Writer* ⇒ *Implementer

Com a supressão do *Course Developer* como papel essencial deixa de beneficiar da possibilidade de delegar neste a produção do material de suporte, manuais de utilizador, etc., com o objectivo de libertar os *Implementers* para as actividades de desenvolvimento. Porém, em equipas de reduzida dimensão, é essencial rentabilizar o conhecimento de todos os intervenientes. Por isso, dado que os conteúdos a produzir pelo *Technical Writer* emanam essencialmente dos pormenores de implementação que os *Implementers* dominam, pelo que se recomenda que sejam estes a assumir este papel.

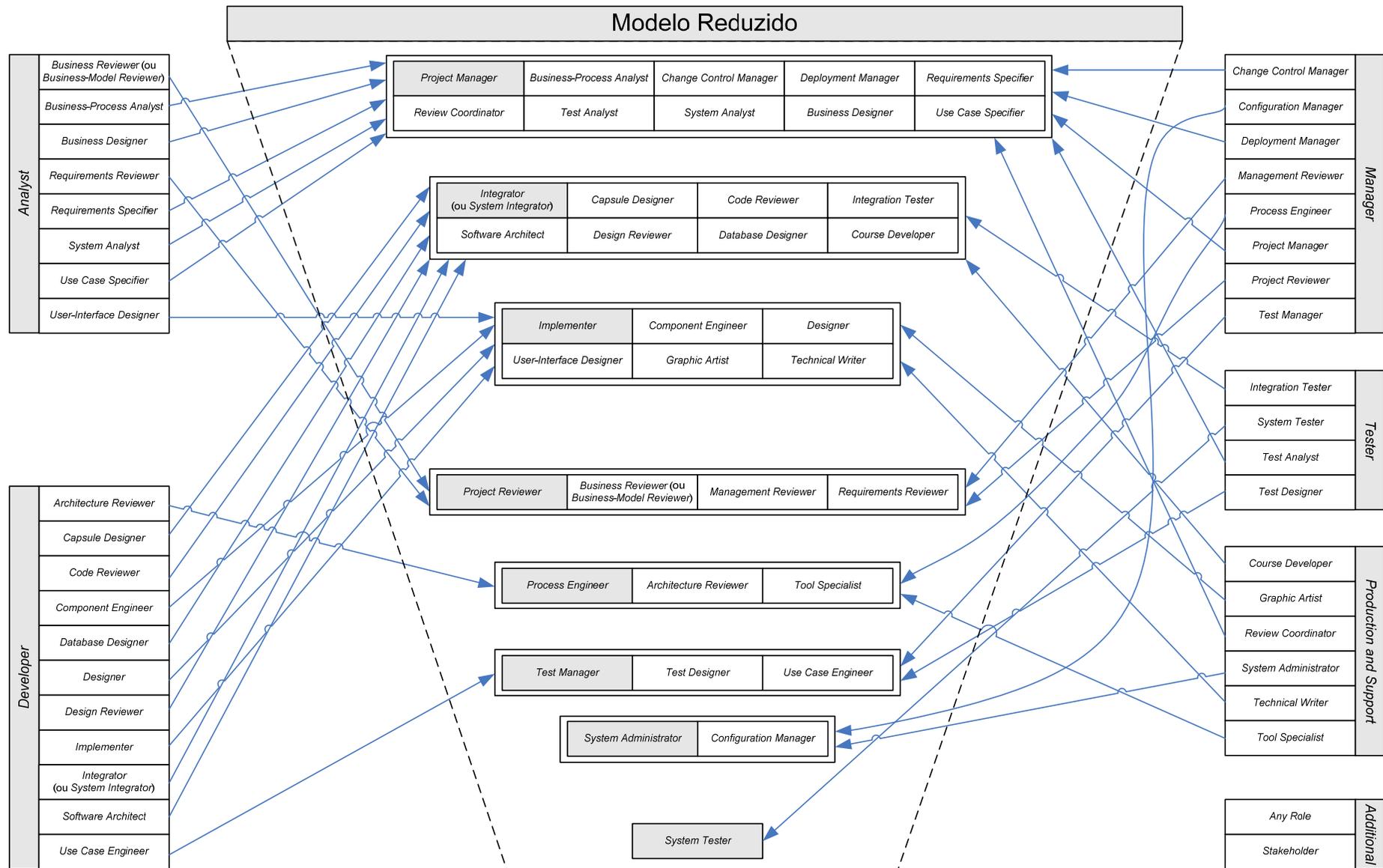


Figura 3.2 – Mapeamentos do Modelo Reduzido

	Modelo Base	Modelo Reduzido																				
Gestor de Projecto	<table border="1"> <tr> <td>Project Manager</td> <td>Business-Process Analyst</td> <td>Change Control Manager</td> <td>Deployment Manager</td> </tr> <tr> <td>Requirements Specifier</td> <td>Review Coordinator</td> <td>Test Analyst</td> <td></td> </tr> </table>	Project Manager	Business-Process Analyst	Change Control Manager	Deployment Manager	Requirements Specifier	Review Coordinator	Test Analyst		<table border="1"> <tr> <td>Project Manager</td> <td>Business-Process Analyst</td> <td>Change Control Manager</td> <td>Deployment Manager</td> </tr> <tr> <td>Requirements Specifier</td> <td>Review Coordinator</td> <td>Test Analyst</td> <td>System Analyst</td> </tr> <tr> <td>Business Designer</td> <td>Use Case Specifier</td> <td></td> <td></td> </tr> </table>	Project Manager	Business-Process Analyst	Change Control Manager	Deployment Manager	Requirements Specifier	Review Coordinator	Test Analyst	System Analyst	Business Designer	Use Case Specifier		
Project Manager	Business-Process Analyst	Change Control Manager	Deployment Manager																			
Requirements Specifier	Review Coordinator	Test Analyst																				
Project Manager	Business-Process Analyst	Change Control Manager	Deployment Manager																			
Requirements Specifier	Review Coordinator	Test Analyst	System Analyst																			
Business Designer	Use Case Specifier																					
Chefe de Equipa	<table border="1"> <tr> <td>Integrator (ou System Integrator)</td> <td>Capsule Designer</td> <td>Code Reviewer</td> <td>Designer</td> </tr> <tr> <td>Integration Tester</td> <td></td> <td></td> <td></td> </tr> </table>	Integrator (ou System Integrator)	Capsule Designer	Code Reviewer	Designer	Integration Tester				<table border="1"> <tr> <td>Integrator (ou System Integrator)</td> <td>Capsule Designer</td> <td>Code Reviewer</td> <td>Integration Tester</td> </tr> <tr> <td>Software Architect</td> <td>Design Reviewer</td> <td>Database Designer</td> <td>Course Developer</td> </tr> </table>	Integrator (ou System Integrator)	Capsule Designer	Code Reviewer	Integration Tester	Software Architect	Design Reviewer	Database Designer	Course Developer				
Integrator (ou System Integrator)	Capsule Designer	Code Reviewer	Designer																			
Integration Tester																						
Integrator (ou System Integrator)	Capsule Designer	Code Reviewer	Integration Tester																			
Software Architect	Design Reviewer	Database Designer	Course Developer																			
Patrocinador de Projecto	<table border="1"> <tr> <td>Project Reviewer</td> <td>Business Reviewer (ou Business-Model Reviewer)</td> <td>Management Reviewer</td> <td>Requirements Reviewer</td> </tr> </table>	Project Reviewer	Business Reviewer (ou Business-Model Reviewer)	Management Reviewer	Requirements Reviewer	<table border="1"> <tr> <td>Project Reviewer</td> <td>Business Reviewer (ou Business-Model Reviewer)</td> <td>Management Reviewer</td> <td>Requirements Reviewer</td> </tr> </table>	Project Reviewer	Business Reviewer (ou Business-Model Reviewer)	Management Reviewer	Requirements Reviewer												
Project Reviewer	Business Reviewer (ou Business-Model Reviewer)	Management Reviewer	Requirements Reviewer																			
Project Reviewer	Business Reviewer (ou Business-Model Reviewer)	Management Reviewer	Requirements Reviewer																			
Analista	<table border="1"> <tr> <td>System Analyst</td> <td>Business Designer</td> <td>Use Case Engineer</td> <td>Use Case Specifier</td> </tr> </table>	System Analyst	Business Designer	Use Case Engineer	Use Case Specifier																	
System Analyst	Business Designer	Use Case Engineer	Use Case Specifier																			
Coordenador de Desenvolvimento	<table border="1"> <tr> <td>Process Engineer</td> <td>Architecture Reviewer</td> <td>Tool Specialist</td> </tr> </table>	Process Engineer	Architecture Reviewer	Tool Specialist	<table border="1"> <tr> <td>Process Engineer</td> <td>Architecture Reviewer</td> <td>Tool Specialist</td> </tr> </table>	Process Engineer	Architecture Reviewer	Tool Specialist														
Process Engineer	Architecture Reviewer	Tool Specialist																				
Process Engineer	Architecture Reviewer	Tool Specialist																				
Programador	<table border="1"> <tr> <td>Implementer</td> <td>Component Engineer</td> </tr> </table>	Implementer	Component Engineer	<table border="1"> <tr> <td>Implementer</td> <td>Component Engineer</td> <td>Designer</td> <td>User-Interface Designer</td> </tr> <tr> <td>Graphic Artist</td> <td>Technical Writer</td> <td></td> <td></td> </tr> </table>	Implementer	Component Engineer	Designer	User-Interface Designer	Graphic Artist	Technical Writer												
Implementer	Component Engineer																					
Implementer	Component Engineer	Designer	User-Interface Designer																			
Graphic Artist	Technical Writer																					
Arquitecto de Software	<table border="1"> <tr> <td>Software Architect</td> <td>Design Reviewer</td> </tr> </table>	Software Architect	Design Reviewer																			
Software Architect	Design Reviewer																					
Coordenador de Infra-Estrutura	<table border="1"> <tr> <td>System Administrator</td> <td>Configuration Manager</td> </tr> </table>	System Administrator	Configuration Manager	<table border="1"> <tr> <td>System Administrator</td> <td>Configuration Manager</td> </tr> </table>	System Administrator	Configuration Manager																
System Administrator	Configuration Manager																					
System Administrator	Configuration Manager																					
Coordenador de Qualidade	<table border="1"> <tr> <td>Test Manager</td> <td>Test Designer</td> </tr> </table>	Test Manager	Test Designer	<table border="1"> <tr> <td>Test Manager</td> <td>Test Designer</td> <td>Use Case Engineer</td> </tr> </table>	Test Manager	Test Designer	Use Case Engineer															
Test Manager	Test Designer																					
Test Manager	Test Designer	Use Case Engineer																				
Coordenador de Comunicação e Imagem	<table border="1"> <tr> <td>User-Interface Designer</td> <td>Graphic Artist</td> </tr> </table>	User-Interface Designer	Graphic Artist																			
User-Interface Designer	Graphic Artist																					
Coordenador de Suporte	<table border="1"> <tr> <td>Course Developer</td> <td>Technical Writer</td> </tr> </table>	Course Developer	Technical Writer																			
Course Developer	Technical Writer																					
Administrador de Bases de Dados	<table border="1"> <tr> <td>Database Designer</td> </tr> </table>	Database Designer																				
Database Designer																						
Auditor de Qualidade	<table border="1"> <tr> <td>System Tester</td> </tr> </table>	System Tester	<table border="1"> <tr> <td>System Tester</td> </tr> </table>	System Tester																		
System Tester																						
System Tester																						

Figura 3.3 – Mapa Comparativo entre o Modelo Base e o Modelo Reduzido

Encontra-se assim apresentado o modelo reduzido proposto, sintetizado na Figura 3.2¹³⁶. Contudo, a simplificação do elenco processual que conduz ao modelo reduzido alicerça-o em três intervenientes primordiais: o *Project Manager*, o *Integrator* e o *Implementer*. Assim, e obviamente sem menosprezo para a relevância dos restantes, será essencialmente do desempenho destes que dependerá o sucesso do projecto, por assegurarem em conjunto um leque de papéis alargado e de especial importância. Por isso, as organizações que considerarem a adaptação deste modelo deverão estar cientes de que os indivíduos chamados a assumir cada uma dessas funções necessitam de ser suficientemente experientes para lidar não só com a complexidade do que deles se espera, como também com a pressão adicional que daí resulta. Adicionalmente, e no sentido de facilitar a comparação entre ambos os modelos apresentados, a Figura 3.3 sumariza a agregação de papéis resultante de cada um.

3.3. Acumulação de Responsabilidades

Apresentado que está o elenco processual proposto, considera-se pertinente reflectir sobre se é ou não recomendável o estabelecimento de restrições à acumulação de vários papéis em simultâneo por um mesmo indivíduo. Se, por um lado, tal recomendação pode implicar a necessidade de mobilização de um maior número de recursos, também é um facto que isso poderá contribuir para um melhor desempenho de cada um dos papéis, ao evitar uma dispersão improdutiva e/ou incompatibilidades de natureza ética.

Pelo exposto, é importante encontrar uma solução equilibrada que passa pela definição das limitações identificadas na Tabela 3.2.

¹³⁶ A Figura 3.3 do mesmo anexo permite uma mais fácil comparação de ambos os modelos.

Tabela 3.2 – Restrições à Acumulação de Papéis (no mesmo projecto)

Caso seja ...	Não deve acumular com ...	Motivação
Gestor de Projecto [GPI]	Chefe de Equipa [CEI]	Ambos estes papéis são, em qualquer dos modelos, de elevada importância e desgaste. Porém, a acumulação destes papéis é desaconselhada, já que se espera do GPI que isole o CEI das pressões e solicitações exteriores, de forma a que este se possa concentrar nas suas funções e assegurar a concretização atempada dos objectivos em jogo. Por outro lado, a acumulação poderia também fazer com que, na ânsia de assegurar o decorrer dos trabalhos, a interacção com o cliente fosse negligenciada, potenciando a insatisfação do mesmo e problemas futuros de gestão de âmbito. Por último, convém lembrar que o GPI exerce uma função de supervisão e coordenação do trabalho do(s) CEI, pelo que também por isso é desejável preservar esta segregação de funções.
Gestor de Projecto [GPI]	Patrocinador de Projecto [PPI]	Para além das suas possíveis funções de influência externa, o(s) PPI encontram-se incumbidos da revisão de vários artefactos produzidos pelo GPI, nomeadamente da arquitectura de negócio e do(s) planeamento(s). Adicionalmente, e dado que o GPI é considerado o responsável máximo pelo sucesso do projecto, considera-se extremamente importante a existência de um (ou mais) PPI, no sentido de ir escrutinando a eficácia do desempenho do GPI e, em caso de necessidade, accionar os mecanismos correctivos apropriados.
Chefe de Equipa [CEI]	Coordenador de Qualidade [CQI]	O CQI terá um papel-chave na avaliação da qualidade dos artefactos (aplicações e não só) produzidos sobre a supervisão do(s) CEI, pelo que há óbvios impedimentos éticos a esta acumulação.
Chefe de Equipa [CEI]	Auditor de Qualidade [AQI]	Esta acumulação apenas deverá ser permitida quando as funções de AQI a que o CEI for chamado não se encontrarem em nada relacionadas com a linha de desenvolvimento que coordena, no sentido de evitar conflitos de interesses.
Coordenador de Desenvolvimento [CDI]	Coordenador de Qualidade [CQI]	Apesar de o CDI não ter participação directa na produção dos artefactos sujeitos à avaliação do CQI, o facto de trabalhar em grande proximidade com o(s) CEI poderá torná-lo parcial na avaliação da qualidade das soluções desenvolvidas, pelo que se desaconselha esta acumulação.
Programador [PRI]	Coordenador de Qualidade [CQI]	O CQI terá um papel-chave na avaliação da qualidade dos artefactos (aplicações e não só) produzidos pelo(s) PRI, pelo que há óbvios impedimentos éticos a esta acumulação.
Programador [PRI]	Auditor de Qualidade [AQI]	Esta acumulação apenas deverá ser permitida quando as funções de AQI a que o PRI for chamado não se encontrarem em nada relacionadas com a linha de desenvolvimento em que participa/ou, no sentido de evitar conflitos de interesses.
Arquitecto de Software [ASI]	Coordenador de Qualidade [CQI]	O ASI irá ter voz activa na definição da arquitectura técnica que estará sob o escrutínio do CQI, pelo que há óbvios impedimentos éticos a esta acumulação.
Arquitecto de Software [ASI]	Auditor de Qualidade [AQI]	Esta acumulação apenas deverá ser permitida quando as funções de AQI a que o ASI for chamado não se encontrarem em nada relacionadas com a sua intervenção no projecto (solicitando-lhe por exemplo a revisão de manuais de utilizador), no sentido de evitar conflitos de interesses.
Coordenador de Qualidade [CQI]	Coordenador de Comunicação e Imagem [CCI]	O CQI terá um papel-chave na avaliação da qualidade dos artefactos (aplicações e não só) produzidos no âmbito do projecto, cuja componente de imagem será concebida e gerida pelo CCI. Assim, também se desaconselha esta acumulação por motivos éticos.
Coordenador de Qualidade [CQI]	Administrador de Bases de Dados [ABI]	O CQI terá um papel-chave na avaliação do desempenho do(s) sistema(s) implementado(s), para o qual contribui em boa medida a eficácia da administração da(s) base(s) de dados levada a cabo pelo ABI, pelo que há impedimentos éticos a esta acumulação.
Coordenador de Comunicação e Imagem [CCI]	Auditor de Qualidade [AQI]	Esta acumulação apenas deverá ser permitida quando as funções de AQI a que o CCI for chamado não se encontrarem em nada relacionadas com a sua intervenção no projecto (solicitando-lhe por exemplo a revisão de manuais de utilizador), no sentido de evitar conflitos de interesses.
Administrador de Bases de Dados [ABI]	Auditor de Qualidade [AQI]	Esta acumulação apenas deverá ser permitida quando as funções de AQI a que o ABI for chamado não se encontrarem em nada relacionadas com a sua intervenção no projecto (solicitando-lhe por exemplo a revisão de manuais de utilizador), no sentido de evitar conflitos de interesses.

A Tabela 3.3 resume as restrições propostas de uma forma mais facilmente legível (onde **x** representa uma restrição absoluta e **?** representa uma restrição condicional).

Tabela 3.3 – Quadro-Resumo de Restrições à Acumulação de Papéis (no mesmo projecto)

Caso seja ...	Gestor de Projecto	Chefe de Equipa	Patrocinador de Projecto	Analista	Coordenador de Desenvolvimento	Programador	Arquitecto de Software	Coordenador de Infra-Estrutura	Coordenador de Qualidade	Coordenador de Comunicação e Imagem	Coordenador de Suporte	Administrador de Bases de Dados	Auditor de Qualidade
Gestor de Projecto		x	x										
Chefe de Equipa	x								x				?
Patrocinador de Projecto	x												
Analista													
Coordenador de Desenvolvimento									x				
Programador									x				?
Arquitecto de Software									x				?
Coordenador de Infra-Estrutura													
Coordenador de Qualidade		x			x	x	x			x		x	
Coordenador de Comunicação e									x				?
Coordenador de Suporte													
Administrador de Bases de Dados									x				?
Auditor de Qualidade		?				?	?			?		?	

Contudo, mesmo em organizações de dimensão muito reduzida é usual verificar-se a existência de mais do que um projecto a decorrer em simultâneo, que “competem” entre si pela obtenção dos recursos (entre os quais os humanos) necessários à prossecução dos seus objectivos. Assim, dada a exiguidade de recursos humanos que caracteriza as organizações PME, é natural que cada indivíduo seja chamado a desempenhar, num mesmo período, funções em vários projectos, esperando-se que tal não o impeça de corresponder adequadamente a todas elas. Pelo exposto, é também importante analisar em que medida poderá ser contraproducente acumular papéis (semelhantes ou distintos) em diferentes projectos, no sentido de identificar as situações a evitar pelas organizações adoptantes. Assim sendo, a Tabela 3.4 apresenta as restrições recomendadas.

Tabela 3.4 – Restrições à Acumulação de Papéis (entre projectos)

Caso seja ... num projecto	Não deve acumular com ... noutro projecto	Motivação
Gestor de Projecto	Gestor de Projecto	Dada a elevada criticidade e ocupação temporal usualmente inerente a ambos os papéis, esta acumulação é desaconselhada, só devendo ser permitida se, após análise concreta do caso, se considerar que não lesa (pelo menos significativamente) a eficácia do desempenho de qualquer uma das funções.
	Chefe de Equipa	
	Programador	
	Coordenador de Qualidade	
Chefe de Equipa	Chefe de Equipa	Dada a elevada criticidade e ocupação temporal usualmente inerente a ambos os papéis, esta acumulação é desaconselhada, só devendo ser permitida se, após análise concreta do caso, se considerar que não lesa (pelo menos significativamente) a eficácia do desempenho de qualquer uma das funções.
	Programador	
	Coordenador de Qualidade	
Coordenador de Desenvolvimento	Coordenador de Qualidade	Dada a intervenção do CDI ser transversal aos vários projectos, a incompatibilidade das suas funções com as do CQI é semelhante à verificada no interior do mesmo projecto.
Programador	Programador	Os recursos partilhados são, mesmo em equipas habituadas a trabalhar em conjunto, bastante difíceis de gerir, conduzindo frequentemente a uma de três situações indesejáveis: desperdício de recursos ¹³⁷ , desgaste excessivo de recursos ¹³⁸ ou quebra de prazos ¹³⁹ . Pelo exposto, dever-se-á zelar pela estabilidade das equipas de desenvolvimento, fazendo valer esta incompatibilidade sempre que possível.

Assim, no sentido de facilitar a observação das restrições identificadas na Tabela 3.4, procedeu-se à sua congregação no quadro-resumo da Tabela 3.5 (onde ✕ representa uma restrição absoluta e ? representa uma restrição condicional). Note-se que, dado que a natureza da intervenção do CDI, a acumulação dessa função entre projectos se encontra obviamente implícita.

¹³⁷ Quando, por falta de comunicação ou qualquer outro motivo, um dos CEI lhe pede menos do que o PRI teria disponibilidade para fazer.

¹³⁸ Quando, por falta de comunicação ou por qualquer outro motivo, um dos CEI lhe pede mais do que o PRI teria disponibilidade para fazer e este, em virtude de um esforço adicional, consegue corresponder de forma adequada.

¹³⁹ Quando, por falta de comunicação ou por qualquer outro motivo, um dos CEI lhe pede mais do que o PRI teria disponibilidade para fazer e este não consegue corresponder de forma adequada.

Tabela 3.5 – Quadro-Resumo de Restrições à Acumulação de Papéis (entre projectos)

Caso seja ... num projecto	Gestor de Projecto	Chefe de Equipa	Patrocinador de Projecto	Analista	Coordenador de Desenvolvimento	Programador	Arquitecto de Software	Coordenador de Infra-Estrutura	Coordenador de Qualidade	Coordenador de Comunicação e Imagem	Coordenador de Suporte	Administrador de Bases de Dados	Auditor de Qualidade
Gestor de Projecto	?	?				?			?				
Chefe de Equipa	?	?				?			?				
Patrocinador de Projecto													
Analista													
Coordenador de Desenvolvimento									x				
Programador	?	?				?							
Arquitecto de Software													
Coordenador de Infra-Estrutura													
Coordenador de Qualidade	?	?			x								
Coordenador de Comunicação e													
Coordenador de Suporte													
Administrador de Bases de Dados													
Auditor de Qualidade													

3.4. Enquadramento Organizacional do Elenco Processual Proposto

Para além de permitir a redução da complexidade do elenco processual (com todas as vantagens que já foram apresentadas), o processo de simplificação descrito anteriormente permite também facilitar a tarefa de selecção dos intervenientes, ao diminuir o número de nomeações que necessitam de ser efectuadas. Porém, se a organização adoptante não possuir uma estrutura orientada à execução de projectos (baseada em equipas de projecto, em matriz, etc.) mas sim uma mais clássica estrutura funcional, poderá encontrar mais dificuldades na identificação dos elementos certos para cada um dos papéis existentes. E, embora essa escolha também dependa em grande medida do perfil desejável¹⁴⁰ para cada tipo de interveniente, considerou-se relevante propor um mapeamento entre os papéis definidos e as funções usualmente encontradas numa PME do sector de desenvolvimento de software,

¹⁴⁰ Que irá ser detalhado no próximo capítulo.

apresentado na Tabela 3.6.

Tabela 3.6 – Enquadramento Organizacional do Elenco Proposto

Papel	Solução Ideal	Solução Alternativa
Gestor de Projecto	Membro Sénior do Dep. de Vendas	Director do Dep. de Vendas, Quadro Directivo, Membro Sénior do Dep. de Desenvolvimento, Membro Sénior do Dep. de Produção/TI
Chefe de Equipa	Membro Sénior do Dep. de Desenvolvimento	Director do Dep. de Desenvolvimento
Patrocinador de Projecto	Director-Geral ¹⁴¹	Quadro Directivo, Indivíduo com ligação privilegiada a cliente/fornecedor
Analista	Membro do Dep. de Qualidade	Membro do Dep. de Vendas, Membro do Dep. de Desenvolvimento
Coordenador de Desenvolvimento	Director do Dep. de Desenvolvimento	
Programador	Membro do Dep. de Desenvolvimento	
Arquitecto de Software	Membro Sénior do Dep. de Desenvolvimento	Director do Dep. de Desenvolvimento
Coordenador de Infra-Estrutura	Membro do Dep. de Produção/TI	Membro do Dep. de Desenvolvimento
Coordenador de Qualidade	Membro do Dep. de Qualidade	Membro do Dep. de Desenvolvimento
Coordenador de Comunicação e Imagem	Membro do Dep. de Marketing e Comunicação	Membro do Dep. de Desenvolvimento , Função subcontratada a terceiros
Coordenador de Suporte	Responsável pela Área de Suporte	Membro do Dep. de Produção/TI, Membro do Dep. de Desenvolvimento
Administrador de Bases de Dados	DBA ¹⁴²	Membro do Dep. de Produção/TI, Função subcontratada a terceiros
Auditor de Qualidade	Nenhuma função em particular ¹⁴³	

Contudo, deve notar-se que esta proposta não se encontra associada a nenhuma estrutura organizacional considerada ideal ou recomendada, dado esta questão não ter sido alvo de investigação no âmbito desta dissertação. Assim, este mapeamento deve apenas ser encarado como uma referência, dadas as variadíssimas molduras organizacionais que poderão ser encontradas neste sector. Para além disso, se o enquadramento apresentado na Tabela 3.6 não for suficiente ou adequado para facilitar a identificação do(s) candidato(s) em melhores condições para cada um dos lugares a preencher, a escolha deverá tomar por base os perfis traçados nas fichas individuais incluídas na secção 4.2, para além da crença existente na

¹⁴¹ Poderá parecer um pouco exagerado mas, em contextos PME, alguns projectos assumem tal importância relativa para a organização que é frequente que o Director-Geral se envolva activamente nos mesmos.

¹⁴² *Database Administrator*, usualmente certificado em pelo menos um motor de base de dados específico.

¹⁴³ Dado que os indivíduos nomeados para este papel poderão possuir incumbências muito distintas entre si (testes aplicativos, revisão de documentação, etc), pelo que não é possível definir um perfil comum.

capacidade de cada indivíduo para assumir responsabilmente as incumbências que lá se encontram descritas. Por isso, as organizações adoptantes deverão analisar se as sugestões efectuadas fazem sentido à luz da sua realidade intrínseca e, se necessário, proceder aos ajustes necessários.

3.5. Conclusões

Ao longo deste capítulo demonstrou-se que, seguindo os raciocínios apresentados, é possível configurar o elenco processual do RUP de modo a reduzir significativamente a sua dimensão e, dessa forma, potenciar a sua utilização por parte de organizações PME. Nesse sentido, procedeu-se a um esforço de redução da complexidade, consubstanciado na concentração dos vários papéis definidos pelo RUP em torno de um conjunto de intervenientes que, pelos motivos apresentados, deveriam ser considerados como essenciais. Desse esforço resultaram dois modelos: (1) um modelo base com treze papéis distintos, destinado a organizações de dimensão média que pretendam adoptar um processo de desenvolvimento de software que, apesar das acumulações de funções, não deixe de rentabilizar a formação e conhecimentos específicos dos vários intervenientes; (2) um modelo reduzido mais pragmático que, com os seus oito papéis distintos, visa permitir a uma organização de micro/pequena dimensão controlar eficazmente o desenrolar dos seus projectos e evitar a sobreposição e/ou indefinição das áreas de intervenção de cada interveniente.

Contudo, os modelos propostos devem ser considerados como um ponto de partida e não como uma panaceia capaz de resolver *de per se* as deficiências do seu processo de desenvolvimento de software. Em vez disso, estes modelos podem e devem ser adaptados às organizações em causa já que cada uma possui condicionalismos próprios que podem aconselhar uma abordagem ligeiramente diferente. Assim, antes da adopção de qualquer um deles, a organização deverá verificar se possui os recursos humanos necessários (preferencialmente respeitando as incompatibilidades recomendadas), estimar os custos de mudança (resultantes de formação, reconversão de processos, etc.) em que terá de incorrer para assegurar a sua implementação e validar se é possível assegurar a sua compatibilização com seu o *modus operandi*.

No capítulo seguinte, será possível encontrar uma caracterização detalhada do elenco processual resultante deste esforço de síntese, visando não só facilitar a escolha dos seus titulares, como também auxiliar no seu desempenho.

4. Caracterização do Elenco Processual Proposto

Este capítulo procede a uma minuciosa caracterização dos vários intervenientes nos modelos anteriormente propostos, visando contribuir dessa forma para potenciar a sua eficácia e facilitar a sua operacionalização. Nesse sentido, as relações entre os vários papéis são representadas sob a forma dos fluxos comunicacionais (internos ou externos) que as suportam e que traduzem as principais interações que têm lugar entre eles. Para que seja possível à organização seleccionar os indivíduos mais apropriados para cada papel, para além de assegurar que cada um deles sabe exactamente as funções que é chamado a assumir, são ainda apresentadas fichas individuais que definem as acções a encetar por cada interveniente em determinados momentos-chave do projecto (início de projecto, início de iteração, durante a iteração, fim de iteração e fim de projecto).

4.1. Introdução

No capítulo anterior levou-se a cabo o exercício de, tomando o elenco processual do RUP como base, apresentar uma solução no sentido de reduzir a cardinalidade deste de forma a tornar verosímil a aplicação do processo num contexto de PMEs. Para isso, propôs-se o mapeamento dos papéis originalmente definidos pelo RUP num conjunto mais restrito de treze (ou oito, no caso do modelo reduzido) papéis distintos, que foram descritos em traços gerais.

Porém, a operacionalização deste elenco sintetizado carece de uma definição minuciosa das respectivas incumbências, que permita delimitar claramente a área de intervenção de cada um dos intervenientes. Assim, e apresentados que estão os dois modelos propostos para o elenco de participantes no processo de desenvolvimento de software de uma organização PME, é chegada a altura de caracterizar com maior profundidade os seus elementos. Dessa forma, não só será mais fácil para as organizações escolherem o indivíduo com o perfil mais apropriado a cada papel, como também para cada um destes saber o que se espera da sua intervenção. Contudo, dado que algumas das organizações interessadas em adoptar este conjunto de papéis poderão não se encontrar familiarizadas com o RUP, este será apresentado

de uma forma independente da metodologia que esteve na sua génese.

Nesse sentido, e de forma a contextualizar as interacções que têm lugar entre os vários intervenientes, a Figura 4.1 identifica os fluxos comunicacionais mais frequentes embora, como será descrito na secção 4.2, existam obviamente outros que, embora ocasionais, não devem ser menosprezados.

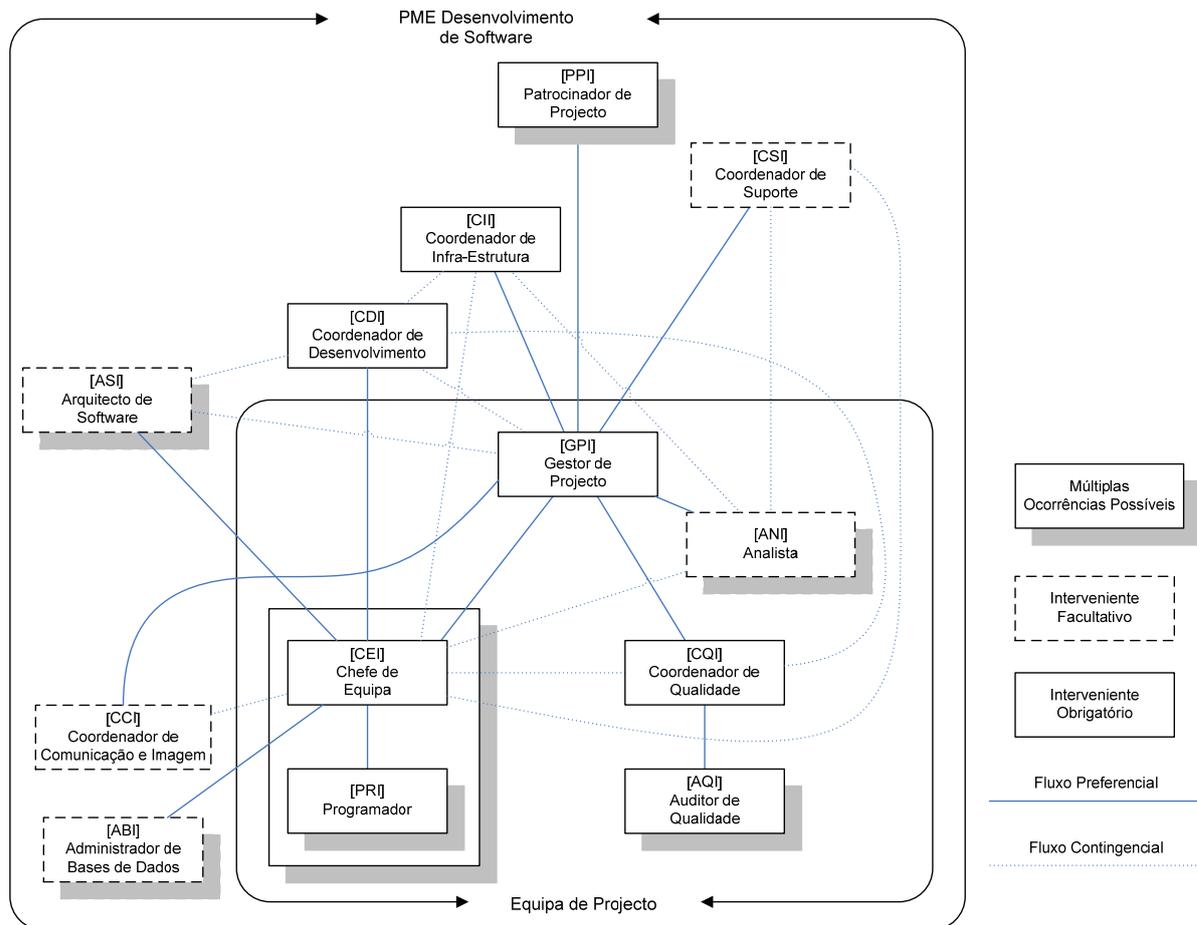


Figura 4.1 – Fluxos de Comunicação (Internos)

As motivações subjacentes a cada um dos fluxos identificados serão discutidas em pormenor aquando da caracterização de cada um dos intervenientes. Contudo, da Figura 4.1 ressaltam algumas questões que carecem de clarificação imediata:

1. a dimensão vertical da Figura 4.1 não possui qualquer conotação hierárquica, seja esta no âmbito do projecto ou organização como um todo;
2. no que concerne à divisão proposta entre o interior e o exterior da equipa de projecto, o facto de não ter considerado que um determinado interveniente não se encontra englobado na referida equipa não significa nem que o seu desempenho

não seja relevante, nem que este não seja formalmente nomeado para executar esse papel no âmbito desse projecto. Em vez disso, essa linha pretende separar os intervenientes que têm um grande envolvimento no projecto (eventualmente diário ou a tempo inteiro) dos que apenas têm uma participação esporádica no mesmo (de acompanhamento, coordenação, etc.);

3. o diagrama deixa transparecer que, no âmbito de um mesmo projecto, poderão existir várias linhas de desenvolvimento que, apesar de possivelmente relacionadas entre si, evoluem de forma independente. Nesses casos, para além de cada CEI poder ser responsável pela coordenação de vários PRI, é plausível que existam vários conjuntos distintos¹⁴⁴ de { CEI, PRI₁, ..., PRI_n }, cada um associado a uma determinada linha de desenvolvimento;
4. a dimensão do projecto e o número de linhas de desenvolvimento existentes poderá implicar a alocação à equipa de projecto de mais do que um AQI, em virtude da extensão das actividades de gestão da qualidade (testes aplicacionais, revisão de documentos, etc.) a realizar;
5. o âmbito do projecto poderá ser de tal forma vasto (horizontal ou verticalmente) que a participação de um ANI não seja suficiente para proceder a um rigoroso levantamento de requisitos, pelo que também este papel poderá ser simultaneamente desempenhado por vários indivíduos;
6. como irá ser referido em maior detalhe quando se proceder à sua caracterização, a função de PPI não se restringe a meras actividades de revisão (da arquitectura de negócio, de planeamento, etc.), podendo tirar partido dos seus contactos para desempenhar um importante papel no desenvolvimento de estratégias de influência¹⁴⁵ externa¹⁴⁶ destinadas a potenciar o sucesso do projecto. Porém, caso o projecto envolva várias entidades externas (clientes, parceiros, fornecedores, etc.), poderá surgir a necessidade de contar com mais do que um PPI, já que é possível que apenas um indivíduo não consiga agir eficazmente junto de todas essas entidades.

¹⁴⁴ Apesar de logicamente distintos, nada obriga que estes conjuntos sejam exclusivos entre si, já que em contextos PME é frequente que um determinado elemento participe simultaneamente em diferentes actividades.

¹⁴⁵ Por exemplos, contactos ao mais alto nível com a entidade cliente no sentido de resolver diferendos que surjam no âmbito do projecto ou de negociar aspectos relevantes.

¹⁴⁶ A necessidade de influência interna não parece fazer muito sentido em organizações de dimensão pequena e média.

Contudo, não é possível concretizar qualquer projecto sem que exista, a vários níveis, uma considerável interacção entre a entidade fornecedora (neste caso, uma PME de desenvolvimento de software) e a entidade cliente, consubstanciando vários fluxos comunicacionais, identificados na Figura 4.2 (incluída no Anexo II em maiores dimensões).

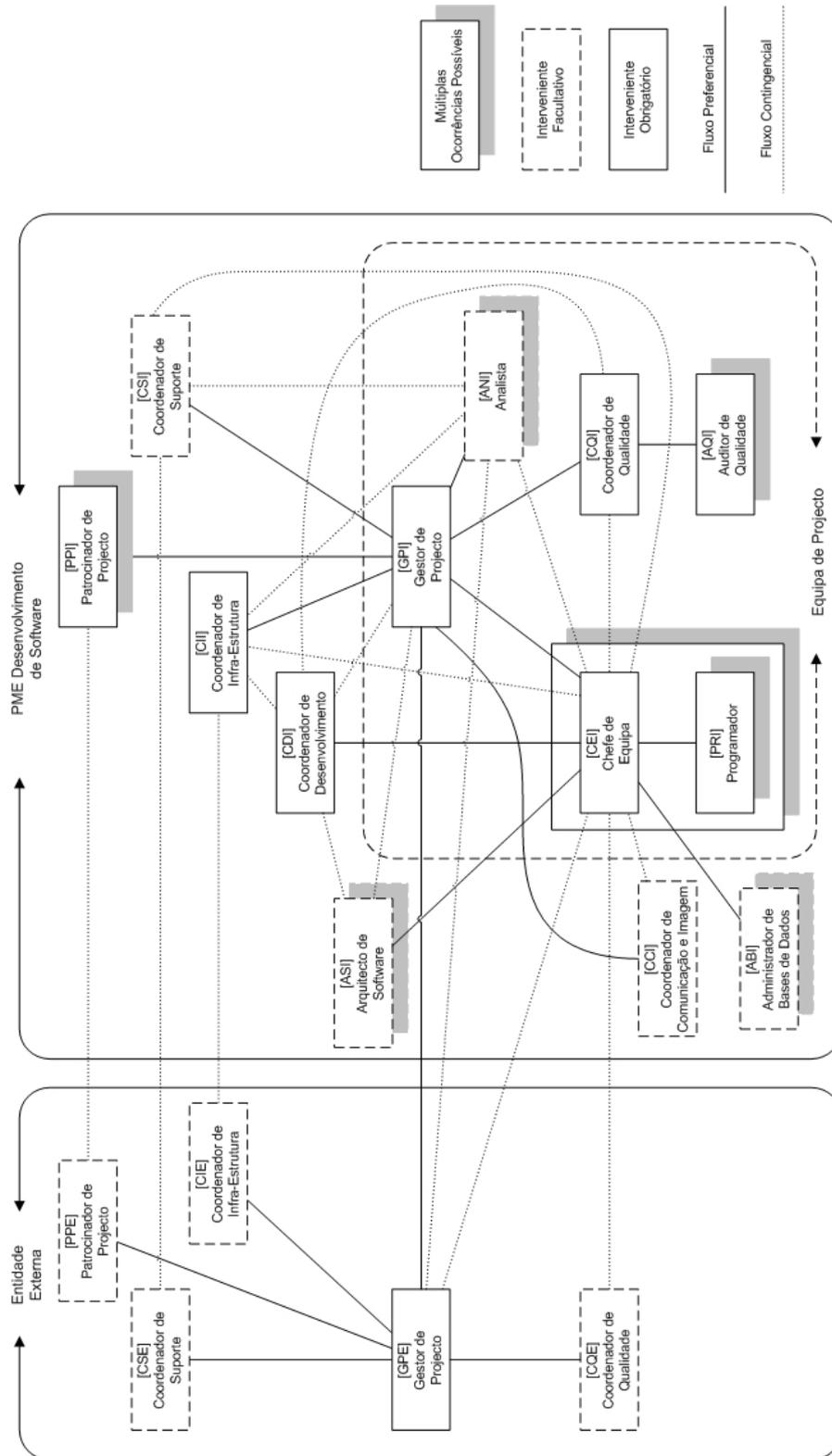


Figura 4.2 – Fluxos de Comunicação (Internos e Externos)

Como é dado observar na Figura 4.2, não obstante a existência de várias comunicações contingenciais¹⁴⁷ entre os mais variados interlocutores internos e externos, o principal elo de ligação entre o interior e o exterior da organização deverá ser o GPI, já que só assim se poderá assegurar o seu controlo efectivo sobre o desenrolar do projecto. Quando tal não acontece, é vulgar constatar a ocorrência de situações indesejáveis, como por exemplo:

- quando se permite que o contacto (sem acompanhamento pelo GPI) entre o CEI e o exterior da organização se torne habitual, potencia-se o risco de aumento gradual do âmbito do projecto decorrente de pedidos directos do cliente ao CEI sem que o GPI os possa enquadrar comercialmente e, em alguns casos, sem que sequer saiba que tiveram lugar (pelo menos até que seja tarde demais);
- quando, da comunicação entre um interveniente interno e outro externo, surgem factos que não são dados a saber (imediatamente ou a muito curto prazo) ao GPI, potencia-se o risco do exterior da organização ter mais informação sobre o projecto do que o GPI, o que obviamente limita a sua capacidade de acção e é lesivo para a imagem da organização;
- quando a intervenção externa de interveniente interno não é concertada com o GPI, potencia-se o risco de veicular informações incorrectas ou desactualizadas, dado que o GPI acompanha de perto o desenrolar do projecto e deverá ser o indivíduo com a visão mais abrangente sobre o estado do mesmo;
- quando a intervenção de um PPI não é concertada com o GPI, potencia-se o risco de assumir compromissos irrealistas (pelo menos sem consequências negativas para a entidade fornecedora) que, por poderem ter lugar ao mais alto nível, usualmente não podem ser desfeitos.

Convém ainda referir que, no sentido de facilitar a identificação dos fluxos comunicacionais que ultrapassam as fronteiras organizacionais, as siglas que identificam cada interveniente terminam com ‘I’ ou com ‘E’, consoante este se encontra no interior ou exterior da organização produtora de software.

¹⁴⁷ Em caso de necessidade premente (por exemplo, a detecção pelo cliente de um problema técnico que, dadas as suas particularidades, o GPI não se encontre em condições de analisar).

Na secção que se segue serão apresentadas as fichas descritivas correspondentes a cada um dos papéis identificados na Figura 4.1.

4.2. Caracterização do Elenco Processual Proposto

Nas subsecções que se seguem irão ser explanadas as especificidades dos vários elementos que compõem o elenco processual proposto (apresentados por ordem alfabética crescente das respectivas siglas), detalhando as suas incumbências e participação nos fluxos de comunicação identificados.

4.2.1. [ABI] Administrador de Bases de Dados

Em projectos em que os requisitos de desempenho identificados careçam de maior atenção ou no âmbito dos quais seja utilizado um motor de base de dados com o qual o CII não se encontre particularmente familiarizado, é de todo recomendável a participação deste interveniente que, fruto da sua experiência e formação específica, poderá contribuir de forma decisiva para a concepção de modelos de dados eficientes e eficazes. Para além disso, a disponibilidade de um elemento com este perfil permitirá também encontrar as melhores soluções para a configuração dos motores de base de dados utilizados, complementadas com os adequados índices, *views*, *constraints*, *triggers* e *stored procedures*.

Responsável por ...

- Configurar e otimizar o funcionamento do(s) motor(es) de base de dados pelo(s) qual(is) é responsável.
- Conceber, em diálogo com o(s) respectivo(s) CEI, o(s) modelo(s) de dados a suportar pelo(s) motor(es) de base de dados pelo(s) qual(is) é responsável.
- Conceber, em diálogo com o(s) respectivo(s) CEI, a política de cópias de segurança mais apropriada para cada um do(s) motor(es) de base de dados pelo(s) qual(is) é responsável.
- Preparar, em diálogo com o(s) respectivo(s) CEI, os índices, *views*, *constraints*, *triggers* e *stored procedures* necessários para otimizar a utilização do(s) modelo(s) de dados suportado(s) pelo(s) motor(es) de base de dados pelo(s) qual(is) é responsável.
- Notificar o GPI sempre que pretender proceder a uma alteração de férias.

Cardinalidade

- Este papel tem um cariz facultativo mas, quando adoptado, poderá ser desempenhado em simultâneo por vários indivíduos num mesmo projecto, caso o mesmo envolva a utilização de distintos motores de base de dados.

Perfil

- Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil:
 - Formação de 2º ciclo de Bolonha (ou equivalente) em informática (nomeadamente em concepção e normalização de modelos de dados);
 - Formação sobre a configuração e administração do(s) motor(es) de base de dados pelo(s) qual(is) é responsável;
 - Pelo menos 2 (dois) anos de experiência na instalação, configuração e administração do(s) motor(es) de base de dados pelo(s) qual(is) é responsável.

Participação em fluxos de informação internos

- [Preferencial] CEI: No sentido de obter as informações necessárias sobre o(s) modelo(s) de dados, índices, *views*, *constraints*, *triggers*, *stored procedures* e política de cópias de segurança a conceber.

Participação em fluxos de informação externos

- (não é expectável a sua participação em fluxos de informação externos)

O seu desempenho é crítico para ...

- Assegurar a optimização do funcionamento do(s) motor(es) de base de dados pelo(s) qual(is) é responsável.
- Assegurar a normalização do(s) modelo(s) de dados suportado(s) pelo(s) motor(es) de base de dados pelo(s) qual(is) é responsável.

Deve evitar ...

- A utilização de tipos de dados ou funcionalidades proprietárias de um determinado motor de base de dados, dado que poderão dificultar uma possível futura substituição do mesmo, desde que tal não resulte numa significativa degradação de desempenho do mesmo.

No início do projecto deve ...

- Conceber, em diálogo com o(s) respectivo(s) CEI, o(s) modelo(s) de dados a suportar pelo(s) motor(es) de base de dados pelo(s) qual(is) é responsável.
- Conceber, em diálogo com o(s) respectivo(s) CEI, a política de cópias de segurança mais apropriada para cada um do(s) motor(es) de base de dados pelo(s) qual(is) é responsável.
- Publicar, em local definido pela organização, a identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.
- Verificar se o seu calendário de férias se encontra actualizado e, caso não esteja, comunicar esse facto ao GPI.

No início de cada iteração deve ...

- Rever, em conjunto com o(s) respectivo(s) CEI, o(s) modelo(s) de dados a suportar pelo(s) motor(es) de base de dados pelo(s) qual(is) é responsável no sentido de identificar possíveis alterações necessárias durante a iteração que se inicia.
- Verificar que a política de cópias de segurança que definiu para cada um do(s) motor(es) de base de dados pelo(s) qual(is) é responsável se encontra a funcionar correctamente e/ou se é pertinente a concretização de alguma alteração à mesma.

Durante cada iteração deve ...

- Configurar e otimizar o funcionamento do(s) motor(es) de base de dados pelo(s) qual(is) é responsável.
- Preparar, em diálogo com o(s) respectivo(s) CEI, os índices, *views*, *constraints*, *triggers* e *stored procedures* necessários para otimizar a utilização do(s) modelo(s) de dados suportado(s) pelo(s) motor(es) de base de dados pelo(s) qual(is) é responsável.

No final do projecto deve ...

- Colaborar, sob coordenação do GPI, na realização de uma cópia de segurança (pelo menos em duplicado) de toda a informação relevante (configurações, *scripts* de inicialização de bases de dados, etc.) relacionada com as bases de dados do projecto.
- Comunicar ao CDI a sua avaliação do processo de desenvolvimento de software utilizado no projecto, sugerindo eventuais correcções ou evoluções.

No final de cada iteração deve ...

- Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis.
- Verificar se pretende efectuar alguma alteração de férias que incida sobre a iteração seguinte e, nesse caso, notificar o GPI.

4.2.2. [ANI] Analista

Apesar de não ser considerada essencial, a participação deste interveniente é altamente recomendável em determinadas situações: caso o GPI não possua os conhecimentos técnicos mínimos¹⁴⁸ para concretizar um rigoroso processo de levantamento de requisitos, a contribuição de um (ou mais) ANI será essencial para delimitar de forma inequívoca o âmbito do projecto e, assim, minimizar o risco de existência de diferentes expectativas sobre o(s) sistema(s) que a organização se propõe fornecer e o(s) que as entidades externas esperam receber. Por outro lado, mesmo que o GPI apresente as necessárias credenciais técnicas, poderá ser desejável recorrer ao contributo de um (ou mais) ANI quando o conjunto de requisitos apresente uma complexidade ou dimensão tal que o GPI não seria suficiente para o caracterizar de forma detalhada e atempada (pelo menos sem prejudicar o desempenho das suas restantes incumbências). Finalmente, convém ainda realçar que o envolvimento de um (ou mais) ANI poderá ser valioso quando este possuir um maior conhecimento (quando comparado com o GPI) sobre a área de negócio em que o projecto irá decorrer.

Responsável por ...

- Coordenar o processo de levantamento de requisitos (funcionais e não funcionais) de forma a produzir a documentação (entre a qual o modelo de casos de uso) necessária à delimitação do âmbito do projecto de uma forma tão inequívoca quanto possível, submetendo-a atempadamente à verificação do GPI.
- Validar internamente a exequibilidade dos requisitos identificados e, sempre que tal não se verifique, promover o diálogo interno com o objectivo de encontrar uma solução alternativa viável e aceitável para as entidades externas.
- Detectar eventuais novos requisitos apresentados ao longo do projecto por entidades externas e formalizar internamente os respectivos pedidos de alteração, comunicando-os ao GPI e submetendo-os rapidamente à apreciação do(s) CEI afectados pelos mesmos.
- Detectar possíveis requisitos que, não tendo sido explicitados pelas entidades externas, podem encontrar-se dentro das suas expectativas ou correspondam a uma oportunidade comercial que possa ser aproveitada, dando conhecimento dos mesmos ao GPI.
- Envidar todos os esforços no sentido de esclarecer requisitos que não tenham sido suficientemente descritos pelas entidades externas, no sentido de alinhar as expectativas de ambas as partes.
- Zelar pela integridade das concretizações dos vários casos de uso que resultam do modelo de casos de uso, assegurando que cumprem integralmente os respectivos requisitos.
- Notificar o GPI sempre que pretender proceder a uma alteração de férias.

Cardinalidade

- Este papel tem uma natureza facultativa, embora recomendável em projectos de grande dimensão ou sempre que os conhecimentos técnicos ou de negócio do GPI não sejam considerados como suficientes, podendo ser desempenhado em simultâneo por vários indivíduos num mesmo projecto e sempre sob a coordenação do GPI.

Perfil

- Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil:
 - Formação de 2º ciclo de Bolonha (ou equivalente) em informática;
 - Conhecimento sobre a área de negócio em que o projecto irá decorrer;
 - Conhecimento aprofundado da notação UML e de técnicas de gestão de requisitos;
 - Pelo menos 1 (um) ano de experiência no desenvolvimento de software, nomeadamente utilizando métodos iterativos.

¹⁴⁸ Por exemplo a familiarização com a notação UML, necessária para a criação do modelo de casos de uso.

Participação em fluxos de informação internos

- [Contingencial] CEI: No sentido de lhe transmitir os requisitos estabelecidos para o sistema pelo qual é responsável e de validar a exequibilidade dos mesmos.
- [Contingencial] CII: No sentido de lhe transmitir os requisitos estabelecidos para a infra-estrutura e de validar a exequibilidade dos mesmos.
- [Contingencial] CSI: No sentido de lhe transmitir os requisitos estabelecidos para o suporte e de validar a exequibilidade dos mesmos.

O seu desempenho é crítico para ...

- Ajudar as entidades externas a identificar as suas reais necessidades e, conseqüentemente, clarificar a totalidade dos requisitos a observar.
- Detectar requisitos não exequíveis e, para cada um, encontrar uma solução alternativa viável e aceitável para as entidades externas.
- Assegurar que GPI se encontra sempre a par de novos requisitos identificados.
- Assegurar que GPI toma conhecimento de novas oportunidades comerciais associadas ao projecto.

No início do projecto deve ...

- Coordenar o processo de levantamento de requisitos (funcionais e não funcionais) de forma a produzir a documentação (entre a qual o modelo de casos de uso) necessária à delimitação do âmbito do projecto de uma forma tão inequívoca quanto possível, submetendo-a atempadamente à verificação do GPI.
- Verificar se o seu calendário de férias se encontra actualizado e, caso não esteja, comunicar esse facto ao GPI.
- Publicar, em local definido pela organização, a identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.

Durante cada iteração deve ...

- Detectar possíveis requisitos que, não tendo sido explicitados pelas entidades externas, podem encontrar-se dentro das suas expectativas ou correspondam a uma oportunidade comercial que possa ser aproveitada, dando conhecimento dos mesmos ao GPI.
- Envidar todos os esforços no sentido de esclarecer requisitos que não tenham sido suficientemente descritos pelas entidades externas, no sentido de alinhar as expectativas de ambas as partes.
- Zelar pela integridade das concretizações dos vários casos de uso que resultam do modelo de casos de uso, assegurando que cumprem integralmente os respectivos requisitos.

No final do projecto deve ...

- Comunicar ao CDI a sua avaliação do processo de desenvolvimento de software utilizado no projecto, sugerindo eventuais correcções ou evoluções.

- [Preferencial] GPI: No sentido de lhe transmitir os requisitos identificados e eventuais oportunidades de negócio detectadas.

Participação em fluxos de informação externos

- [Contingencial] GPE: No sentido de obter as informações necessárias ao processo de levantamento de requisitos.

Deve evitar ...

- Desresponsabilizar-se da clarificação dos requisitos a observar alegando que as entidades externas não sabem as suas reais necessidades, já que é sua responsabilidade ajudar a clarificá-las.

No início de cada iteração deve ...

- Validar com o GPI se documentação de requisitos carece de actualização.

- Detectar eventuais novos requisitos apresentados por entidades externas e formalizar internamente os respectivos pedidos de alteração, comunicando-os ao GPI e submetendo-os rapidamente à apreciação do(s) CEI afectados pelos mesmos.

- Validar internamente a exequibilidade dos requisitos identificados e, sempre que tal não se verifique, promover o diálogo interno com o objectivo de encontrar uma solução alternativa viável e aceitável para as entidades externas.

No final de cada iteração deve ...

- Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis.
- Verificar se pretende efectuar alguma alteração de férias que incida sobre a iteração seguinte e, nesse caso, notificar o GPI.

4.2.3. [AQI] Auditor de Qualidade

A intervenção de cada AQI visa essencialmente auxiliar o CQI a assegurar que a qualidade percebida pelas entidades externas sobre os artefactos entregues é positiva e contribui para o reforço da imagem da organização. Nesse sentido, dele se espera que exerça com empenho e brio profissional todas as tarefas que lhe forem atribuídas pelo CQI no âmbito do plano interno de auditoria de qualidade por ele gizado.

Responsável por ...

- Executar, com o máximo empenho e brio profissional, as tarefas que lhe forem atribuídas pelo CQI.
- Contribuir para assegurar que os artefactos produzidos ao longo do projecto se encontram de acordo com os respectivos requisitos identificados pelo GPI e/ou pelo(s) ANI.
- Comunicar semanalmente ao CQI o tempo de que ainda necessita para terminar as tarefas que tem a seu cargo.
- Notificar o CQI sempre que pretender proceder a uma alteração de férias.
- Contribuir para assegurar que os artefactos documentais primam pela correcção do seu conteúdo, são coerentes entre si (do ponto de vista visual, terminológico, etc.) e não deixam transparecer as idiosincrasias próprias dos vários intervenientes.
- Contribuir para assegurar que são observadas as boas práticas em vigor na organização no que concerne à criação de artefactos documentais, entre as quais poderão estar as seguintes:
 - Utilização de identificadores de documentos unívocos e normalizados;
 - Utilização de um sistema de gestão documental;
 - Utilização de modelos pré-definidos para determinados tipos de documentos.

Cardinalidade

- Seja qual for o modelo adoptado, este papel tem um cariz obrigatório, podendo ser desempenhado em simultâneo por vários indivíduos num mesmo projecto caso o número/complexidade dos artefactos existentes o justifiquem.

Perfil

- Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil:
 - Formação de 2º ciclo de Bolonha (ou equivalente) em informática (caso se encontre responsável por analisar artefactos aplicativos);
 - Conhecimentos de informática na óptica do utilizador (caso não se encontre responsável por analisar artefactos aplicativos);
 - Conhecimento básico da notação UML (diagramas de casos de uso).

Participação em fluxos de informação internos

- [Preferencial] CQI: No sentido de tomar conhecimento sobre as tarefas que este lhe atribuir e, conseqüentemente, de lhe comunicar informação sobre a evolução da execução das mesmas.

Participação em fluxos de informação externos

- (não é expectável a sua participação em fluxos de informação externos)

O seu desempenho é crítico para ...

- Contribuir para criar e/ou reforçar uma imagem positiva da organização junto das entidades externas.
- Contribuir para assegurar a detecção de situações que, mesmo de pouca gravidade e fácil correcção, poderiam proporcionar uma má experiência de utilização às entidades externas.

Deve evitar ...

- Aceitar tarefas de auditoria que exijam conhecimentos que não possua.
- Omitir do CQI que o prazo que lhe foi dado para executar uma determinada tarefa não é suficiente.
- Omitir do CQI situações em que, apesar de inicialmente ter considerado suficiente o prazo para executar uma determinada tarefa, verifique que isso não corresponde à verdade.

No início do projecto deve ...

- Verificar se o seu calendário de férias se encontra actualizado e, caso não esteja, comunicar esse facto ao CQI.
- Publicar, em local definido pela organização, a identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.

Durante cada iteração deve ...

- Executar, com o máximo empenho e brio profissional, as tarefas que lhe forem atribuídas pelo CQI.
- Contribuir para assegurar que os artefactos produzidos ao longo do projecto se encontram de acordo com os respectivos requisitos identificados pelo GPI e/ou pelo(s) ANI.
- Contribuir para assegurar que os artefactos documentais primam pela correcção do seu conteúdo, são coerentes entre si (do ponto de vista visual, terminológico, etc.) e não deixam transparecer as idiosincrasias próprias dos vários intervenientes.

No final do projecto deve ...

- Comunicar ao CDI a sua avaliação do processo de desenvolvimento de software utilizado no projecto, sugerindo eventuais correcções ou evoluções.

No início de cada iteração deve ...

- Colaborar na execução do plano interno de auditoria de qualidade definido para os artefactos produzidos pela organização durante a iteração anterior, solicitando ao CQI a atribuição das tarefas para a iteração.

- Contribuir para assegurar que são observadas as boas práticas em vigor na organização no que concerne à criação de artefactos documentais, entre as quais poderão estar as seguintes:
 - Utilização de identificadores de documentos unívocos e normalizados;
 - Utilização de um sistema de gestão documental;
 - Utilização de modelos pré-definidos para determinados tipos de documentos.
- Notificar o CQI sempre que pretender proceder a uma alteração de férias.

No final de cada iteração deve ...

- Verificar se pretende efectuar alguma alteração de férias que incida sobre a iteração seguinte e, nesse caso, notificar o CQI.
- Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis.

4.2.4. [ASI] Arquitecto de Software

Independentemente de se encontrarem ou não instituídas práticas destinadas a gerir e fazer evoluir as competências dos recursos humanos, é habitual a existência de assimetrias de conhecimento entre os vários colaboradores de uma organização, resultantes não só da sua experiência profissional acumulada ao participar em projectos de distinta complexidade, como também das iniciativas individuais encetadas por cada indivíduo no sentido de se actualizar e acompanhar a evolução tecnológica. Adicionalmente, dada a típica exiguidade dos recursos humanos de uma organização PME, acrescida da possível dificuldade em lhes proporcionar formações com a abrangência e frequência adequada, nem sempre é possível disponibilizar elementos com o perfil desejado para um novo projecto que se inicia, em virtude de se encontrarem envolvidos em tarefas das quais não possam ser libertados. Assim, é por vezes necessário nomear um CEI que, apesar de suficientemente experiente para coordenar o(s) PRI envolvidos, poderá não possuir todos os conhecimentos necessários para gerar, pelo menos de forma autónoma, as melhores soluções técnicas que estariam ao alcance da organização (na ausência de constrangimentos de recursos).

Pelo exposto, nestas circunstâncias é aconselhável que a intervenção do(s) CEI seja apoiada por um ou mais¹⁴⁹ ASI que, tirando partido da sua maior experiência, os deverão aconselhar e ajudar no processo de tomada das principais decisões de índole técnica. Dessa forma, a organização poderá elevar a qualidade das soluções a implementar sem com isso fazer perigar outros projectos em curso, o que poderia acontecer caso decidisse nomear os elementos mais experientes como CEI do novo projecto (mantendo as responsabilidades nos restantes projectos), dada a inevitável dispersão de esforços causada.

Contudo, convém realçar que o sucesso desta colaboração depende em grande medida da capacidade do(s) CEI de não encarar o(s) ASI como rival(is), mas sim como alguém com quem terão oportunidade de aprender, trocar experiências e dialogar sobre aspectos técnicos que, usualmente, estão longe de ser consensuais. Por seu lado, será igualmente importante que o(s) ASI não negligenciem os aspectos didácticos e pedagógicos dessa relação, no sentido de facilitar e potenciar o processo de aprendizagem do(s) CEI, alertando para erros comuns e justificando a direcção a prosseguir.

¹⁴⁹ Dado que poderão ser utilizadas em simultâneo várias tecnologias ou técnicas, dominadas por diferentes ASI.

Responsável por ...

- Identificar, estimar e comunicar ao GPI e CDI os riscos técnicos do projecto decorrentes das decisões arquitecturais tomadas.
- Propor ao CDI, em diálogo com o respectivo CEI, as principais decisões de índole técnica, que resultam na arquitectura de cada linha de desenvolvimento, expressas da seguinte forma:
 - Lista das ferramentas, servidores aplicativos e motores de bases de dados a utilizar;
 - Diagrama de *Deployment* UML que descreva a forma como o sistema pelo qual é responsável se encontra interligado com todos os outros sistemas relevantes;
 - Diagrama de Componentes UML que descreva os principais componentes existentes na sua linha de desenvolvimento e identifique as relações entre si.
- Avaliar, mediante proposta de um CEI, as interfaces dos principais componentes do sistema.
- Auxiliar o CDI a avaliar o desempenho do(s) CEI.
- Auxiliar o CDI a avaliar as propostas de cada CEI para a lista de componentes a reaproveitar e não existentes a criar pela sua linha de desenvolvimento.
- Acompanhar as tendências tecnológicas no sentido de identificar possíveis ferramentas, metodologias ou conceitos que possam ser de utilidade para o(s) CEI envolvidos no projecto.
- Desenvolver acções no sentido de fazer evoluir as competências técnicas do(s) CEI envolvidos no projecto.
- Aconselhar o CDI sobre a uniformização dos métodos de trabalho a empregar pela equipas de desenvolvimento, nomeadamente no que concerne a:
 - Ferramentas (de planeamento, coordenação, gestão documental, desenvolvimento, modelação, *logging*, *unit testing* e *bug tracking*) a utilizar;
 - Metodologias e padrões de desenvolvimento;
 - Nome dos pacotes/componentes do projecto;
 - Localização base no sistema de controlo de versões;
 - Processo de *code review*;
 - Utilização de *integration build*;
 - Códigos de erro;
 - Formato de ficheiros de *log* e categorias a utilizar;
 - Forma de configuração (ficheiro, base de dados, etc.) de cada um dos componentes da sua linha de desenvolvimento.
- Assessorar o CDI no processo de validação das avaliações dos componentes reutilizados realizadas pelo(s) CEI.
- Notificar o GPI e respectivo(s) CEI sempre que pretender proceder a uma alteração de férias.

Cardinalidade

- Este papel tem um cariz facultativo mas, quando adoptado, poderá ser desempenhado em simultâneo por vários indivíduos num mesmo projecto, caso o mesmo envolva tecnologias variadas ou careça do aconselhamento em áreas dominadas por diferentes ASI.

Perfil

- Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil:
 - Formação de 2º ciclo de Bolonha (ou equivalente) em informática, complementada com um 2º ou 3º ciclo em engenharia de software;
 - Pelo menos 5 (cinco) anos de experiência no desenvolvimento de software, nomeadamente utilizando métodos iterativos;
 - Forte preocupação com a gestão do seu conhecimento, concretizada em frequentes iniciativas no sentido de o actualizar;
 - Elevada capacidade analítica;
 - Conhecimento aprofundado da notação UML.

Participação em fluxos de informação internos

- [Contingencial] CDI: No sentido de:
 - Informar sobre a prestação do(s) CEI;
 - Identificar, estimar e comunicar os riscos técnicos do projecto decorrentes das decisões arquitecturais tomadas;
 - Propor principais decisões de índole técnica, que resultam na arquitectura de cada linha de desenvolvimento;
 - Aconselhar sobre a uniformização dos métodos de trabalho a empregar pela equipas de desenvolvimento.

- [Preferencial] CEI: No sentido de o envolver na tomada das principais decisões de índole técnica relacionadas com a sua linha de desenvolvimento e de o ajudar a evoluir.
- [Contingencial] GPI: No sentido de identificar, estimar e comunicar os riscos técnicos do projecto decorrentes das decisões arquitecturais tomadas.

Participação em fluxos de informação externos

- (não é expectável a sua participação em fluxos de informação externos)

O seu desempenho é crítico para ...

- Ajudar o(s) CEI a encontrar melhores soluções técnicas para os problemas com que se confrontam e, conseqüentemente, aumentar a qualidade das mesmas.
- Ajudar o(s) CEI a fazer evoluir as suas competências.

Deve evitar ...

- Hostilizar e/ou desconsiderar a participação do(s) CEI.

No início do projecto deve ...

- Identificar, estimar e comunicar ao GPI e CDI os riscos técnicos do projecto decorrentes das decisões arquitecturais tomadas.
- Propor ao CDI, em diálogo com o respectivo CEI, as principais decisões de índole técnica, que resultam na arquitectura de cada linha de desenvolvimento, expressas da seguinte forma:
 - Lista das ferramentas, servidores aplicativos e motores de bases de dados a utilizar;
 - Diagrama de *Deployment* UML que descreva a forma como o sistema pelo qual é responsável se encontra interligado com todos os outros sistemas relevantes;
 - Diagrama de Componentes UML que descreva os principais componentes existentes na sua linha de desenvolvimento e identifique as relações entre si.
- Avaliar, mediante proposta de um CEI, as interfaces dos principais componentes do sistema.
- Verificar se o seu calendário de férias se encontra actualizado e, caso não esteja, comunicar esse facto ao GPI e respectivo(s) CEI.
- Publicar, em local definido pela organização, a identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.

- Aconselhar o CDI sobre a uniformização dos métodos de trabalho a empregar pela equipas de desenvolvimento, nomeadamente no que concerne a:
 - Ferramentas (de planeamento, coordenação, gestão documental, desenvolvimento, modelação, *logging*, *unit testing* e *bug tracking*) a utilizar;
 - Metodologias e padrões de desenvolvimento;
 - Nome dos pacotes/componentes do projecto;
 - Localização base no sistema de controlo de versões;
 - Processo de *code review*;
 - Utilização de *integration build*;
 - Códigos de erro;
 - Formato de ficheiros de *log* e categorias a utilizar;
 - Forma de configuração (ficheiro, base de dados, etc.) de cada um dos componentes da sua linha de desenvolvimento.

No início de cada iteração deve ...

- Auxiliar o CDI a avaliar as propostas de cada CEI para a lista de componentes a reaproveitar e não existentes a criar pela sua linha de desenvolvimento.

Durante cada iteração deve ...

- Sempre que se justificar, comunicar ao GPI e CDI a necessidade de alterar a arquitectura de uma linha de desenvolvimento e/ou os riscos técnicos identificados, actualizando a documentação afectada pela mesma.
- Desenvolver acções no sentido de fazer evoluir as competências técnicas do(s) CEI envolvidos no projecto.

- Acompanhar as tendências tecnológicas no sentido de identificar possíveis ferramentas, metodologias ou conceitos que possam ser de utilidade para o(s) CEI envolvidos no projecto.

No final do projecto deve ...

- Assessorar o CDI no processo de validação das avaliações dos componentes reutilizados realizadas pelo(s) CEI.
- Comunicar ao CDI a sua avaliação das competências técnicas e comportamentais do(s) CEI que acompanhou ao longo do projecto.
- Comunicar ao CDI a sua avaliação do processo de desenvolvimento de software utilizado no projecto, sugerindo eventuais correcções ou evoluções.

No final de cada iteração deve ...

- Auxiliar o CDI a avaliar o desempenho do(s) CEI.
- Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis.
- Verificar se pretende efectuar alguma alteração de férias que incida sobre a iteração seguinte e, nesse caso, notificar o GPI e respectivo(s) CEI.

4.2.5. [CCI] Coordenador de Comunicação e Imagem

Em equipas de projecto maioritariamente compostas por elementos com formação em informática é frequente verificar uma excessiva (e quase exclusiva) preocupação com os requisitos funcionais e com determinados requisitos não funcionais (escalabilidade, desempenho, capacidade, etc.), em detrimento de requisitos não funcionais relacionados com a imagem que os vários componentes/sistemas deixam transparecer. Para além disso, é também vulgar verificar que, pela natureza da sua formação, este tipo de profissionais apresenta uma tendência para menosprezar a necessidade de comunicar determinados aspectos¹⁵⁰ e para privilegiar o conteúdo da comunicação (e não tanto a forma como é transmitido¹⁵¹), o que poderá resultar na degradação da imagem da organização perante entidades externas. Por outro lado, e independentemente da qualidade implícita aos componentes/sistemas implementados, também é facto que alguns representantes de entidades externas apreciam o resultado do projecto essencialmente em função da sua aparência. Pelo exposto, em projectos com alguma relevância é importante a participação de um elemento, com formação nas áreas da comunicação e imagem, com a missão de evitar que estes aspectos sejam negligenciados pelos restantes participantes.

Responsável por ...

- | | |
|--|---|
| <ul style="list-style-type: none"> ▪ Assegurar a satisfação de todas as necessidades de imagem e comunicação do projecto, que poderão assumir (entre outras) as seguintes formas: <ul style="list-style-type: none"> ○ Apresentações; ○ Documentos (comunicados de imprensa, manuais, etc.); ○ Modelos de documentos; ○ Logótipos; ○ Brochuras; ○ Interfaces gráficas (nomeadamente <i>Web</i>). | <ul style="list-style-type: none"> ▪ Zelar pela usabilidade das interfaces gráficas utilizadas no projecto. ▪ Solicitar ao GPI todos os recursos (logótipos, modelos, etc.) relacionados com as entidades externas de que necessite para exercer as tarefas a seu cargo. ▪ Identificar, em conjunto com o GPI, a lista das necessidades de imagem e comunicação previstas para o projecto. ▪ Notificar o GPI sempre que pretender proceder a uma alteração de férias. |
|--|---|

Cardinalidade

- Este papel tem um cariz facultativo mas, quando adoptado, deverá ser desempenhado por um único indivíduo por projecto.

Perfil

- Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil:
 - Formação de 2º ciclo de Bolonha (ou equivalente) em ciências da comunicação e/ou multimédia;
 - Forte sentido estético;
 - Conhecimento aprofundado das ferramentas de tratamento de imagem adoptadas pela organização.

¹⁵⁰ Por exemplo, por considerar que são do conhecimento geral, o que poderá não se verificar com utilizadores menos experientes.

¹⁵¹ Por exemplo, não se preocupando com uma adequada estruturação dos documentos ou com a dimensão estética dos mesmos.

Participação em fluxos de informação internos

- [Contingencial] CEI: No sentido de identificar com maior detalhe as necessidades de comunicação e imagem específicas da linha de desenvolvimento cada um coordena.
- [Preferencial] GPI: No sentido de tomar conhecimento sobre as necessidades de comunicação e imagem do projecto e de o manter informado sobre o decorrer das suas actividades.

O seu desempenho é crítico para ...

- Assegurar que a imagem que transparece da organização na sequência do projecto é cuidada e contribui para a criação de associações positivas nas entidades externas.

No início do projecto deve ...

- Identificar, em conjunto com o GPI, a lista das necessidades de imagem e comunicação previstas para o projecto.
- Verificar se o seu calendário de férias se encontra actualizado e, caso não esteja, comunicar esse facto ao GPI.
- Publicar, em local definido pela organização, a identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.

Durante cada iteração deve ...

- Assegurar a satisfação de todas as necessidades de imagem e comunicação do projecto, que poderão assumir (entre outras) as seguintes formas:
 - Apresentações;
 - Documentos (comunicados de imprensa, manuais, etc.);
 - Modelos de documentos;
 - Logótipos;
 - Brochuras;
 - Interfaces gráficas (nomeadamente *Web*).

No final do projecto deve ...

- Comunicar ao CDI a sua avaliação do processo de desenvolvimento de software utilizado no projecto, sugerindo eventuais correcções ou evoluções.

Participação em fluxos de informação externos

- (não é expectável a sua participação em fluxos de informação externos)

Deve evitar ...

- Actuar de forma incoerente com as normas de identidade e política de comunicação da organização.

No início de cada iteração deve ...

- Rever, em conjunto com o GPI, a lista das necessidades de imagem e comunicação previstas para a iteração que se inicia.

- Zelar pela usabilidade das interfaces gráficas utilizadas no projecto.
- Dialogar com o(s) CEI no sentido de identificar com maior detalhe as necessidades de comunicação e imagem específicas de cada linha de desenvolvimento.
- Solicitar ao GPI todos os recursos (logótipos, modelos, etc.) relacionados com as entidades externas de que necessite para exercer as tarefas a seu cargo.

No final de cada iteração deve ...

- Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis.
- Verificar se pretende efectuar alguma alteração de férias que incida sobre a iteração seguinte e, nesse caso, notificar o GPI.

4.2.6. [CDI] Coordenador de Desenvolvimento

Ao contrário do que se verifica com os restantes intervenientes, as funções de CDI devem ser desempenhadas pelo mesmo indivíduo em todos os projectos de desenvolvimento de software em que a organização se encontre envolvida, no sentido de lhe permitir efectuar uma eficiente gestão e afectação do(s) ASI, CEI e PRI que coordena pelos vários projectos que deles necessitam. Assim, o CDI pode e deve servir como elo de ligação entre os vários projectos estabelecendo, sempre que se justifique, fluxos de comunicação entre intervenientes em projectos distintos cuja colaboração considere benéfica para a organização.

Para além disso, recai também sobre o CDI a responsabilidade de acompanhar o trabalho dos vários CEI que se encontram em funções em cada momento, potenciando a troca de experiências entre eles e zelando pela homogeneidade dos métodos de trabalho, essencial para promover uma fácil mobilidade de recursos entre projectos, o que por vezes é necessário para fazer face a situações inesperadas. Caso se justifique, o acompanhamento de um determinado CEI poderá ser complementado com a intervenção de um ou mais ASI que, fruto de uma maior experiência, ajudarão o CEI em causa a encontrar as melhores soluções técnicas para os problemas com que a sua linha de desenvolvimento se confrontar. Quando isso se verifica, o CDI deverá não só assegurar que o(s) ASI proporciona(m) ao(s) CEI o devido aconselhamento, como também que este(s) último(s) observa(m) as directrizes definidas pelo(s) primeiro(s).

Uma outra vertente não menos importante do trabalho deste interveniente consiste em zelar pela manutenção e expansão do repositório de componentes da organização, de forma a potenciar a capacidade de reutilização do mesmo e, conseqüentemente, aumentar a sua eficiência (em virtude de não precisar de desenvolver os componentes que já possui) e qualidade (em virtude de utilizar componentes que já foram previamente testados). Por outro lado, é também responsável por gerir adequadamente as competências do(s) ASI, CEI e PRI que coordena, tomando as medidas necessárias no sentido de assegurar a evolução das mesmas de acordo com o ritmo e direcção apropriados.

Finalmente, convém ainda referir que se espera que o CDI se certifique da existência e disponibilidade de todas as ferramentas necessárias para suportar o processo de desenvolvimento, para além de zelar pelo estrito cumprimento das suas responsabilidades por parte de todos os restantes intervenientes no processo.

Responsável por ...

- Servir de elo de ligação entre os vários projectos em que a organização se encontra envolvida estabelecendo, sempre que se justifique, fluxos de comunicação entre intervenientes em projectos distintos cuja colaboração considere benéfica para a organização.
- Nomear o(s) CEI, ASI e PRI envolvidos em cada projecto.
- Comunicar ao(s) CEI o perfil de competências dos PRI que estarão sobre a sua coordenação.
- Acompanhar o trabalho do(s) ASI e CEI que se encontram em funções, potenciando a troca de experiências entre eles.
- Zelar pela manutenção e expansão do repositório de componentes da organização, promovendo a reutilização e aperfeiçoamento dos seus elementos.
- Assegurar, em diálogo com o(s) ASI e CEI, a homogeneidade dos métodos de trabalho utilizados nos vários projectos, de modo a planear de forma estável a infra-estrutura de suporte ao processo de desenvolvimento e a facilitar a mobilidade dos PRI entre projectos, independentemente das características específicas de cada CEI.
- Sempre que um GPI identificar um obstáculo à concretização de um determinado objectivo, envidar todos os esforços para, em conjunto com o respectivo CEI, o conseguir ultrapassar (eventualmente através da reafecção de recursos).
- Mediar situações de desacordo entre um CEI e um ASI, PRI ou GPI.
- Efectuar uma eficaz gestão das competências do(s) ASI, CEI e PRI que coordena, no sentido de não só os preparar para as responsabilidades que irão assumir no projecto, como também para os papéis que poderão vir a assumir em projectos futuros.
- Zelar pela existência e disponibilidade de todas as ferramentas necessárias para suportar o processo de desenvolvimento.
- Avaliar, em conjunto com o GPI e com o(s) ASI, o desempenho do(s) CEI e, quando o mesmo não for satisfatório, tomar as necessárias medidas correctivas.
- Avaliar, em conjunto com cada CEI, o desempenho do(s) PRI que coordena e, quando o mesmo não for satisfatório, tomar as necessárias medidas correctivas.
- Avaliar, em diálogo com o CII e CQI, as propostas do(s) CEI sobre o conteúdo, formato e localização da documentação interna a produzir durante o projecto no sentido de assegurar de forma apropriada as actividades futuras de manutenção.
- Avaliar, mediante proposta de um ASI ou CEI, as principais decisões de índole técnica que resultam na arquitectura de cada linha de desenvolvimento.
- Zelar pelo cumprimento das suas responsabilidades por parte de todos os restantes intervenientes no processo.
- Zelar pela melhoria contínua do processo de desenvolvimento de software da organização.

Cardinalidade

- Um interveniente para a totalidade dos projectos.

Perfil

- Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil:
 - Responsável máximo pelo departamento/área de desenvolvimento de software da organização;
 - Formação de 2º ciclo de Bolonha (ou equivalente) em informática, complementada com um 2º ciclo em engenharia de software e/ou 2º ciclo em ciências empresariais;
 - Pelo menos 5 (cinco) anos de experiência no desenvolvimento de software, nomeadamente utilizando métodos iterativos;
 - Elevada capacidade de decisão;
 - Elevada capacidade negocial;
 - Elevada tolerância à pressão;
 - Conhecimento aprofundado da notação UML.

O seu desempenho é crítico para ...

- Assegurar o cumprimento das suas responsabilidades por parte de todos os restantes intervenientes no processo.
- Afectar com a máxima eficácia e eficiência o(s) ASI, CEI e PRI que coordena aos vários projectos em curso em cada momento.
- Assegurar que o(s) ASI proporciona(m) ao(s) CEI o devido aconselhamento.
- Assegurar que o(s) CEI observa(m) as directrizes definidas pelo(s) ASI.
- Assegurar a motivação do(s) ASI, CEI e PRI que coordena.

Deve evitar ...

- Atribuir tarefas aos PRI (relacionadas ou não com o projecto) sem o conhecimento prévio do respectivo CEI.

Participação em fluxos de informação internos

- [Contingencial, *apenas no modelo base*] ASI: No sentido de avaliar as principais decisões de índole técnica, que resultam na arquitectura de cada linha de desenvolvimento.
- [Preferencial] CEI: No sentido de:
 - Se manter informado sobre o decorrer das actividades das várias linhas de desenvolvimento;
 - Assegurar que dispõem do(s) PRI adequado(s), suficiente(s) e essencial(is) à sua linha de desenvolvimento;
 - Assegurar a uniformidade da forma de trabalho da sua equipa;
 - Assegurar o reaproveitamento de componentes previamente existentes e/ou a desenvolver durante o projecto;
 - [*Apenas no modelo reduzido*] Avaliar as principais decisões de índole técnica, que resultam na arquitectura da linha de desenvolvimento que coordena.

- [Contingencial] CII: Com o objectivo de recolher a sua opinião sobre as propostas do(s) CEI para o conteúdo, formato e localização da documentação interna a produzir durante o projecto no sentido de assegurar de forma apropriada as actividades futuras de manutenção.
- [Contingencial] CQI: Com o objectivo de recolher a sua opinião sobre as propostas do(s) CEI para o conteúdo, formato e localização da documentação interna a produzir durante o projecto no sentido de assegurar de forma apropriada as actividades futuras de manutenção.
- [Contingencial] GPI: No sentido de se manter informado sobre a prestação do(s) CEI envolvidos no projecto.

Participação em fluxos de informação externos

- (não é expectável a sua participação em fluxos de informação externos)

No início do projecto deve ...

- Nomear o(s) CEI, ASI e PRI envolvidos em cada projecto.
- Comunicar ao(s) CEI o perfil de competências dos PRI que estarão sobre a sua coordenação.
- Avaliar, mediante proposta de um ASI ou CEI, as principais decisões de índole técnica que resultam na arquitectura de cada linha de desenvolvimento.
- Avaliar se as competências do(s) ASI, CEI e PRI são adequadas e suficientes para fazer face às necessidades do projecto e, caso não sejam, envidar todos os esforços (por exemplo, através de formação interna e/ou externa) no sentido de colmatar as lacunas identificadas.
- Avaliar, em diálogo com o CII e CQI, as propostas do(s) CEI sobre o conteúdo, formato e localização da documentação interna a produzir durante o projecto no sentido de assegurar de forma apropriada as actividades futuras de manutenção.
- Avaliar se as ferramentas existentes e disponíveis na organização são adequadas e suficientes para suportar o processo de desenvolvimento do projecto e, caso não sejam, envidar todos os esforços (por exemplo, seleccionando e licenciando as ferramentas necessárias) no sentido de colmatar as lacunas identificadas.
- Publicar, em local definido pela organização, a identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.

- Ajudar, em diálogo com o ASI, o(s) CEI a uniformizar os métodos de trabalho da sua equipa, nomeadamente no que concerne a:
 - Ferramentas (de planeamento, coordenação, gestão documental, desenvolvimento, modelação, *logging*, *unit testing* e *bug tracking*) a utilizar;
 - Metodologias e padrões de desenvolvimento;
 - Nome dos pacotes/componentes do projecto;
 - Localização base no sistema de controlo de versões;
 - Processo de *code review*;
 - Utilização de *integration build*;
 - Códigos de erro;
 - Formato de ficheiros de *log* e categorias a utilizar;
 - Forma de configuração (ficheiro, base de dados, etc.) de cada um dos componentes da sua linha de desenvolvimento.

No início de cada iteração deve ...

- Avaliar, em diálogo com o(s) ASI, as propostas de cada CEI para a lista de componentes a reaproveitar e não existentes a criar pela sua linha de desenvolvimento.

Durante cada iteração deve ...

- Servir de elo de ligação entre os vários projectos em que a organização se encontra envolvida estabelecendo, sempre que se justifique, fluxos de comunicação entre intervenientes em projectos distintos cuja colaboração considere benéfica para a organização.
- Sempre que um GPI identificar um obstáculo à concretização de um determinado objectivo, envidar todos os esforços para, em conjunto com o respectivo CEI, o conseguir ultrapassar (eventualmente através da reafectação de recursos).

- Promover reuniões de ponto de situação semanais (máx. 30 min.) com o(s) CEI e ASI que coordena.
- Zelar pelo cumprimento das suas responsabilidades por parte de todos os restantes intervenientes no processo.
- Mediar em situações de desacordo entre um CEI e um ASI ou GPI.

No final do projecto deve ...

- Verificar se todos os componentes que cada CEI se propôs criar ao longo do projecto foram desenvolvidos e publicados no repositório da organização.
- Validar, em conjunto com o(s) ASI, as avaliações dos componentes reutilizados realizadas pelo(s) CEI.
- Validar as avaliações de competências realizadas pelo(s) GPI, CEI e, quando existam, pelo(s) ASI.
- Analisar, de acordo com a estratégia assumida pela organização e com as conclusões retiradas das avaliações do(s) ASI, CEI e PRI, a pertinência de iniciativas (por exemplo, formação interna e/ou externa) que visem preparar os mesmos para os papéis que poderão vir a assumir em projectos futuros.
- Analisar, de acordo com as conclusões retiradas das avaliações dos restantes intervenientes e com a sua percepção sobre o desenrolar do projecto, a pertinência de proceder a correcções ou evoluções do processo de desenvolvimento de software utilizado na organização.
- [*Apenas no modelo base*] Proceder à avaliação de competências comportamentais do(s) ASI envolvidos no projecto.
- [*Apenas no modelo reduzido*] Realizar a avaliação das competências técnicas do(s) CEI que acompanhou ao longo do projecto.

No final de cada iteração deve ...

- Avaliar, em conjunto com o GPI e com o(s) ASI, o desempenho do(s) CEI e, quando o mesmo não for satisfatório, tomar as necessárias medidas correctivas.
- Avaliar, em conjunto com cada CEI, se o(s) PRI de que dispôs durante a iteração que termina são adequados, suficientes e essenciais à sua linha de desenvolvimento na iteração seguinte.
- Avaliar, em conjunto com cada CEI, o desempenho do(s) PRI que coordena e, quando o mesmo não for satisfatório, tomar as necessárias medidas correctivas.
- Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis.

4.2.7. [CEI] Chefe de Equipa

À semelhança do GPI e CQI, este é seguramente um dos elementos mais importantes do elenco processual proposto, ao ser responsável pela coordenação das actividades de produção dos artefactos necessários à concretização dos objectivos estipulados pelo GPI para um determinado sistema, na observância das prioridades veiculadas por este. Assim, compete-lhe não só atribuir tarefas ao(s) PRI, como também efectuar o controlo periódico da sua execução no sentido de detectar (tão precocemente quanto possível) eventuais atrasos ou problemas, que deverá resolver em cooperação com o GPI. Estabelecendo uma analogia para a área da construção civil, este interveniente pode ser considerado como o equivalente ao mestre de obra¹⁵², responsável por controlar de forma contínua e presencial o desenrolar da mesma. Adicionalmente, deve colaborar com o CQI no sentido de lhe prestar toda a colaboração necessária à avaliação dos artefactos disponibilizados pela sua linha de desenvolvimento, para além de agir em conformidade com as conclusões emanadas durante esse processo. Convém ainda realçar que a sua intervenção é crucial para endereçar dois outros aspectos: (1) a provável carência por parte do GPI dos conhecimentos técnicos necessários à eficaz orientação das actividades do(s) PRI alocados ao projecto, lacuna que o CEI deverá conseguir suprir em virtude da sua experiência; (2) o elevado número de PRI envolvidos na implementação de projectos de grande envergadura tornaria virtualmente impossível que uma só pessoa (em concreto o GPI) conseguisse coordenar e controlar o seu trabalho. Logo, ao repartir os PRI por várias linhas de desenvolvimento e delegar num CEI a coordenação de cada uma delas, é possível ultrapassar esse problema e libertar o GPI dessa preocupação. Pelo exposto, a actividade deste interveniente encontra-se essencialmente focada no interior da organização, embora possa necessitar de interagir directamente com o exterior, principalmente em projectos em que as entidades externas disponham de uma equipa de controlo de qualidade.

Os chefes de equipa desempenham ainda um importante papel na operacionalização da estratégia de reutilização de código fonte da organização, já que lhes compete não só assegurar o reaproveitamento do maior número possível de componentes já existentes, como também promover (dentro dos limites de âmbito do projecto) a criação de novos componentes de uso genérico com potencial relevância para projectos futuros.

¹⁵² Equiparando, por exemplo, o CDI ao empreiteiro, o GPI ao promotor da obra, o ASI ao engenheiro civil e o CQI ao inspector camarário.

Responsável por ...

-
- Zelar pelo cumprimento de todos os objectivos definidos para a sua linha de desenvolvimento.
 - Coordenar, de forma tão eficiente e eficaz quanto possível, o trabalho do(s) PRI que lhe foram disponibilizados pelo CDI, atribuindo-lhe(s) as tarefas mais apropriadas ao seu perfil e evitando situações de dependência ociosa. Nesse sentido, deverão planear (com a antecedência possível) as actividades a desenvolver ao longo de cada iteração do projecto, definindo de forma atempada e inequívoca a responsabilidade de execução de cada uma.
 - Propor ao GPI o número e duração das iterações¹⁵³ a efectuar, o conteúdo das mesmas e respectivos artefactos, juntamente com as características técnicas de cada um.
 - Planear e executar a integração dos componentes implementados na sua linha de desenvolvimento, no sentido de produzir as versões requeridas em cada iteração.
 - Planear e executar os testes de integração apropriados sobre cada versão produzida, no sentido de assegurar que os componentes incluídos na mesma funcionam correctamente em conjunto.
 - Propor ao CDI o conteúdo, formato e localização da documentação interna a produzir durante o projecto.
 - Assegurar a produção e publicação atempada da documentação interna considerada necessária.
 - Classificar cada pedido de alteração recebido como *simples* (passível de ser incorporado na actual iteração sem prejuízo de outras funcionalidades) ou *complexo* (implicam aumento de prazo ou custo ou redução de qualidade/funcionalidade).
 - Alertar (tão cedo quanto possível) o GPI para situações em que não seja possível atingir todos os objectivos da iteração dentro do prazo previsto.
 - Propor ao CDI a lista de componentes a reaproveitar e não existentes que possam ser criados pela sua linha de desenvolvimento.
 - Manter, em colaboração com o GPI, um registo dos eventos mais significativos relacionados com o desenvolvimento do projecto (por exemplo, atrasos protagonizados por entidades externas, aceitação dos artefactos, etc.).
 - Gerir a comunicação entre o GPI e o(s) PRI que coordena.
 - Avaliar, em conjunto com o CDI, o desempenho do(s) PRI que coordena.
 - Identificar e comunicar ao GPI as necessidades de infra-estrutura do projecto (nos vários ambientes a estabelecer).
 - Efectuar os pedidos de passagem a certificação/produção e garantir a criação da respectiva documentação de suporte.
 - Assegurar a disponibilidade do projecto no ambiente de desenvolvimento.
 - Conhecer pormenorizadamente os detalhes de implementação da sua linha de desenvolvimento, estando em condições de os discutir com outros intervenientes, internos ou externos.
 - Inspeccionar periodicamente o código fonte gerado pelo(s) PRI que coordena e aferir a sua qualidade e conformidade com políticas definidas para o projecto.
 - Uniformizar, em consonância com o CDI, os métodos de trabalho da sua equipa, nomeadamente no que concerne a:
 - Ferramentas (de planeamento, coordenação, gestão documental, desenvolvimento, modelação, *logging*, *unit testing* e *bug tracking*) a utilizar;
 - Metodologias e padrões de desenvolvimento;
 - Nome dos pacotes/componentes do projecto;
 - Localização base no sistema de controlo de versões;
 - Processo de *code review*;
 - Utilização de *integration build*;
 - Códigos de erro;
 - Formato de ficheiros de *log* e categorias a utilizar;
 - Forma de configuração (ficheiro, base de dados, etc.) de cada um dos componentes da sua linha de desenvolvimento.
 - Assegurar que o(s) PRI que coordena observa(m) as boas práticas em vigor na organização no que concerne ao desenvolvimento de software, entre as quais poderão estar as seguintes:
 - Não utilização de valores *hard-coded*, optando antes pela sua inclusão na configuração (seja em que formato for) do componente/aplicação;
 - Implementação de testes unitários nos componentes por si desenvolvidos;
 - Actualizar frequentemente (períodos inferiores a uma semana) no sistema de controlo de versões o código fonte do(s) componente(s) em que se encontra envolvido, com a condição de o mesmo se encontrar compilável sem erros.
 - Acompanhar a execução do plano interno (e eventualmente externo) de auditoria de qualidade sobre os artefactos produzidos pela sua linha de desenvolvimento.
 - Verificar periodicamente o calendário de férias do(s) PRI que coordena.
 - Assegurar a produção, disponibilização e preservação da documentação necessária para suportar a execução dos processos de passagem a certificação/produção relacionados com a linha de desenvolvimento pela qual é responsável.
 - Notificar o GPI sempre que pretender proceder a uma alteração de férias.

¹⁵³ Sugerem-se iterações com a duração de 1 (um) mês, cujo início coincida com o primeiro dia útil do mês.

Responsabilidades adicionais no modelo base

- Definir as interfaces dos principais componentes do sistema.
- Assessorar o ASI na tomada das principais decisões de índole técnica relacionadas com a sua linha de desenvolvimento.

Responsabilidades adicionais no modelo reduzido

- Avaliar, mediante proposta de um PRI, as interfaces dos principais componentes do sistema.
- Propor ao CDI as principais decisões de índole técnica, que resultam na arquitectura da sua linha de desenvolvimento, expressas da seguinte forma:
 - Lista das ferramentas, servidores aplicativos e motores de bases de dados a utilizar;
 - Diagrama de *Deployment* UML que descreva a forma como o sistema pelo qual é responsável se encontra interligado com todos os outros sistemas relevantes;
 - Diagrama de Componentes UML que descreva os principais componentes existentes na sua linha de desenvolvimento e identifique as relações entre si.
- Identificar, estimar e comunicar ao GPI e CDI os riscos técnicos do projecto decorrentes das decisões arquitecturais tomadas.
- Preparar os índices, *views*, *constraints*, *triggers* e *stored procedures* necessários para otimizar a utilização do(s) modelo(s) de dados suportado(s) pelo(s) motor(es) de base de dados pelo(s) qual(is) é responsável.
- Identificar, em diálogo com o GPI, o conteúdo, formato e localização da documentação de suporte a produzir durante o projecto pela sua linha de desenvolvimento.
- Conceber o(s) modelo(s) de dados de que necessita a linha de desenvolvimento pela qual é responsável.

Cardinalidade

- Seja qual for o modelo adoptado, este papel tem um cariz obrigatório, podendo até ser desempenhado em simultâneo por vários indivíduos num mesmo projecto, ficando cada um responsável por uma linha de desenvolvimento que, apesar de possivelmente relacionada com as restantes, possui objectivos próprios e evolui de forma independente.

Perfil

- Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil:
 - Formação de 2º ciclo de Bolonha (ou equivalente) em informática;
 - Pelo menos 2 (dois) anos de experiência no desenvolvimento de software, nomeadamente utilizando métodos iterativos;
 - Conhecimento aprofundado da notação UML.

Participação em fluxos de informação internos

- [Preferencial, *apenas no modelo base*] ABI: No sentido de lhe transmitir as informações necessárias sobre o(s) modelo(s) de dados, índices, views, constraints, triggers, stored procedures e política de cópias de segurança específicas à linha de desenvolvimento que coordena.
- [Contingencial, *apenas no modelo base*] ANI: No sentido de tomar conhecimento sobre os requisitos estabelecidos para o sistema pelo qual é responsável, cuja exequibilidade lhe compete validar.
- [Preferencial, *apenas no modelo base*] ASI: No sentido de o assessorar na tomada das principais decisões de índole técnica relacionadas com a sua linha de desenvolvimento.
- [Contingencial, *apenas no modelo base*] CCI: No sentido de o ajudar a identificar com maior detalhe as necessidades de comunicação e imagem específicas da linha de desenvolvimento que coordena.
- [Contingencial] CII: No sentido de esclarecer eventuais dúvidas que surjam durante os processos de passagem a certificação/produção, de identificar os itens que deverão ser sujeitos a monitorização (activa e/ou passiva) e de acordar a divisão de tarefas necessária para o concretizar.
- [Contingencial] CQI: No sentido de prestar os esclarecimentos necessários à concepção e execução do plano interno de auditoria de qualidade sobre os artefactos produzidos pela sua linha de desenvolvimento e de obter comentários sobre os resultados obtidos.
- [Contingencial, *apenas no modelo base*] CSI: No sentido de colaborar na concepção de um apropriado plano de formação a ministrar aos utilizadores finais e às diferentes linhas das várias equipas de suporte e na identificação dos conteúdos de suporte a produzir.

- [Preferencial] CDI: No sentido de:
 - Informar sobre o decorrer das actividades da linha de desenvolvimento que coordena;
 - Assegurar a colaboração do(s) PRI de que necessita;
 - Assegurar a uniformidade da forma de trabalho da sua equipa;
 - Assegurar o reaproveitamento dos componentes previamente existentes e/ou a desenvolver durante o projecto;
 - [*Apenas no modelo reduzido*] Propor e obter a sua aprovação para as principais decisões de índole técnica, que resultam na arquitectura da linha de desenvolvimento que coordena.
- [Preferencial] GPI: No sentido de o manter informado sobre o decorrer das actividades da linha de desenvolvimento que coordena.
- [Preferencial] PRI: No sentido de lhe(s) atribuir tarefas e de coordenar a execução das mesmas.

Participação em fluxos de informação externos

- [Contingencial] CQE: No sentido de prestar os esclarecimentos necessários à concepção e execução do plano externo de auditoria de qualidade sobre os artefactos produzidos pela sua linha de desenvolvimento e de obter comentários sobre os resultados obtidos.
- [Contingencial] GPE: No sentido, caso necessário e com o conhecimento do GPI, esclarecer ou acordar detalhes de implementação da linha de desenvolvimento que coordena.

O seu desempenho é crítico para ...

- Assegurar o cumprimento dos objectivos de cada iteração.
- Assegurar a motivação do(s) PRI que coordena.
- Proteger o(s) PRI que coordena de pressões internas e/ou externas que possam condicionar o seu desempenho.
- Detectar requisitos não-exequíveis ou insuficientemente descritos.

Deve evitar ...

- Influenciar decisões (o que não significa que não possam efectuar sugestões) meramente comerciais, da competência do GPI.
- Hostilizar e/ou rejeitar a colaboração do(s) ASI.
- Hostilizar o CQI e/ou não lhe prestar todos os esclarecimentos solicitados.

No início do projecto deve ...

- Propor ao GPI o número e duração das iterações a efectuar, o conteúdo das mesmas e respectivos artefactos, juntamente com as características técnicas de cada um, que deverão incluir:
 - *Aplicações*: Novas funcionalidades (descritas em texto livre, documentos ou diagramas), forma de disponibilização (on-line, instalação externa, DVD, etc.), requisitos de comunicação e imagem (interfaces gráficas, etc.);
 - *Documentos*: Língua(s), tópicos a abordar, formato (Word, Excel, PDF, etc.), requisitos de comunicação e imagem (modelos a utilizar, etc.).
- Identificar e comunicar ao GPI as necessidades de infra-estrutura do projecto (nos vários ambientes a estabelecer).
- Uniformizar, em consonância com o CDI, os métodos de trabalho da sua equipa, nomeadamente no que concerne a:
 - Ferramentas (de planeamento, coordenação, gestão documental, desenvolvimento, modelação, *logging*, *unit testing* e *bug tracking*) a utilizar;
 - Metodologias e padrões de desenvolvimento;
 - Nome dos pacotes/componentes do projecto;
 - Localização base no sistema de controlo de versões;
 - Processo de *code review*;
 - Utilização de *integration build*;
 - Códigos de erro;
 - Formato de ficheiros de *log* e categorias a utilizar;
 - Forma de configuração (ficheiro, base de dados, etc.) de cada um dos componentes da sua linha de desenvolvimento.
- Propor ao CDI o conteúdo, formato e localização da documentação interna a produzir durante a iteração actual.
- Verificar o calendário de férias do(s) PRI que coordena para o período previsto de execução do projecto.
- Verificar se o seu calendário de férias se encontra actualizado e, caso não esteja, comunicar esse facto ao GPI.
- Publicar, em local definido pela organização, as seguintes informações:
 - Número e duração das iterações a efectuar (e respectivo conteúdo) acordadas com o GPI;
 - Conteúdo, formato e localização da documentação interna acordada com o CDI;
 - Definição dos métodos de trabalho acordados com o CDI;
 - Referência para o(s) modelo(s) de dados utilizado(s) pela sua linha de desenvolvimento;
 - Identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.

- [*Apenas no modelo base*] Assessorar o ASI na tomada das principais decisões de índole técnica relacionadas com a sua linha de desenvolvimento.
- [*Apenas no modelo reduzido*] Conceber o(s) modelo(s) de dados de que necessita a linha de desenvolvimento pela qual é responsável.
- [*Apenas no modelo reduzido*] Identificar, estimar e comunicar ao GPI e CDI os riscos técnicos do projecto decorrentes das decisões arquitecturais tomadas.
- [*Apenas no modelo reduzido*] Avaliar, mediante proposta de um PRI, as interfaces dos principais componentes do sistema.
- [*Apenas no modelo reduzido*] Identificar, em diálogo com o GPI, o conteúdo, formato e localização da documentação de suporte a produzir durante o projecto pela sua linha de desenvolvimento.
- [*Apenas no modelo reduzido*] Propor ao CDI as principais decisões de índole técnica, que resultam na arquitectura da sua linha de desenvolvimento, expressas da seguinte forma:
 - Lista das ferramentas, servidores aplicativos e motores de bases de dados a utilizar;
 - Diagrama de *Deployment UML* que descreva a forma como o sistema pelo qual é responsável se encontra interligado com todos os outros sistemas relevantes;
 - Diagrama de Componentes UML que descreva os principais componentes existentes na sua linha de desenvolvimento e identifique as relações entre si.

No início de cada iteração deve ...

- Propor ao CDI a lista de componentes a reaproveitar e não existentes que possam ser criados pela sua linha de desenvolvimento.
- Inspeccionar o código fonte gerado pelo(s) PRI que coordena durante a iteração anterior e aferir a sua qualidade e conformidade com políticas definidas para o projecto.
- Atribuir tarefas ao(s) PRI que coordena.
- Planear a integração dos componentes implementados na sua linha de desenvolvimento, no sentido de produzir as versões requeridas em cada iteração.
- Acompanhar a execução do plano interno (e eventualmente externo) de auditoria de qualidade sobre os artefactos produzidos pela sua linha de desenvolvimento na iteração anterior.
- Acompanhar a execução das passagens a certificação/produção dos artefactos aplicativos produzidos na iteração anterior.

Durante cada iteração deve ...

- Zelar pelo cumprimento de todos os objectivos definidos para a sua linha de desenvolvimento.
- Classificar cada pedido de alteração recebido como *simples* (passível de ser incorporado na actual iteração sem prejuízo de outras funcionalidades) ou *complexo* (implicam aumento de prazo ou custo ou redução de qualidade/funcionalidade).
- Manter, em colaboração com o GPI, um registo dos eventos mais significativos relacionados com o desenvolvimento do projecto (por exemplo, atrasos protagonizados por entidades externas, aceitação dos artefactos, etc.).
- Efectuar os pedidos de passagem a certificação/produção e garantir a criação da respectiva documentação de suporte.
- Conhecer pormenorizadamente os detalhes de implementação da sua linha de desenvolvimento, estando em condições de os discutir com outros intervenientes, internos ou externos.
- Promover reuniões de ponto de situação semanais (máx. 30 min.) com o(s) PRI que coordena.
- Coordenar o trabalho do(s) PRI que lhe foram disponibilizados pelo CDI, atribuindo-lhe(s) as tarefas mais apropriadas ao seu perfil.
- Gerir a comunicação entre o GPI e o(s) PRI que coordena.
- Alertar (tão cedo quanto possível) o GPI para situações em que não seja possível atingir todos os objectivos da iteração dentro do prazo previsto.
- Assegurar a disponibilidade do projecto no ambiente de desenvolvimento.
- Validar a exequibilidade dos requisitos estabelecidos para o sistema pelo qual é responsável (e, caso não exista CSI, para o suporte ao mesmo) e, sempre que tal não se verifique, ajudar a encontrar uma solução alternativa viável e aceitável para as entidades externas.
- Detectar eventuais novos requisitos que decorram da implementação do projecto que, após devidamente documentados, deverá dar a conhecer ao GPI (e caso exista(m) ao(s) ANI).
- Detectar requisitos estabelecidos para o sistema pelo qual é responsável (e, caso não exista CSI, para o suporte ao mesmo) que não se encontrem suficientemente descritos, reportando-os ao GPI (e caso exista(m) ao(s) ANI).
- Detectar possíveis requisitos que, não tendo sido explicitados pelas entidades externas, podem encontrar-se dentro das suas expectativas ou correspondam a uma oportunidade comercial que possa ser aproveitada, dando conhecimento dos mesmos ao GPI (e caso exista(m) ao(s) ANI).
- Assegurar que o(s) PRI que coordena observa(m) as boas práticas em vigor na organização no que concerne ao desenvolvimento de software, entre as quais poderão estar as seguintes:
 - Não utilização de valores *hard-coded*, optando antes pela sua inclusão na configuração (seja em que formato for) do componente/aplicação;
 - Implementação de testes unitários nos componentes por si desenvolvidos;
 - Actualizar frequentemente (períodos inferiores a uma semana) no sistema de controlo de versões o código fonte do(s) componente(s) em que se encontra envolvido, com a condição de o mesmo se encontrar compilável sem erros.
- [Apenas no modelo base] Definir as interfaces dos principais componentes do sistema.
- [Apenas no modelo base] Dialogar com o CCI no sentido de identificar com maior detalhe as necessidades de comunicação e imagem específicas da sua linha de desenvolvimento.
- [Apenas no modelo reduzido] Sempre que se justificar, comunicar ao GPI e CDI a necessidade de alterar a arquitectura da sua linha de desenvolvimento e/ou os riscos técnicos identificados, actualizando a documentação afectada pela mesma.
- [Apenas no modelo reduzido] Preparar os índices, *views*, *constraints*, *triggers* e *stored procedures* necessários para otimizar a utilização do(s) modelo(s) de dados suportado(s) pelo(s) motor(es) de base de dados pelo(s) qual(is) é responsável.
- [Apenas no modelo reduzido] Preparar os conteúdos necessários para a execução do plano de formação a ministrar aos utilizadores finais e às diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas).

No final do projecto deve ...	No final de cada iteração deve ...
<ul style="list-style-type: none"> ▪ Garantir que o ABI e o(s) PRI que coordena cumpriram as suas actividades de fecho de projecto. ▪ Comunicar ao CDI a sua avaliação das competências técnicas e comportamentais do(s) PRI que coordenou ao longo do projecto. ▪ Comunicar ao CDI a sua avaliação dos componentes reutilizados pela sua linha de desenvolvimento, identificando eventuais correcções ou evoluções a realizar. ▪ Colaborar, sob coordenação do GPI, na realização de uma cópia de segurança (pelo menos em duplicado) de toda a informação relevante (código fonte, artefactos, etc.) associada à sua linha de desenvolvimento. ▪ Comunicar ao CDI a sua avaliação do processo de desenvolvimento de software utilizado no projecto, sugerindo eventuais correcções ou evoluções. 	<ul style="list-style-type: none"> ▪ Executar a integração dos componentes implementados na sua linha de desenvolvimento, no sentido de produzir as versões requeridas em cada iteração, assegurando a consequente disponibilização para passagem a certificação/produção. ▪ Planear e executar os testes de integração apropriados sobre cada versão produzida, no sentido de assegurar que os componentes incluídos na mesma funcionam correctamente em conjunto. ▪ Garantir que todos os componentes desenvolvidos se encontram actualizados no sistema de controlo de versões. ▪ Garantir que o ABI e o(s) PRI que coordena cumpriram as suas actividades de fecho de iteração. ▪ Verificar o calendário de férias do(s) PRI que coordena para o período previsto para a próxima iteração. ▪ Rever se as necessidades de infra-estrutura do projecto se mantêm na iteração seguinte e, caso contrário, comunicar ao GPI as alterações necessárias. ▪ Avaliar, em conjunto com o CDI, se o(s) PRI de que dispôs durante a iteração que termina são adequados, suficientes e essenciais à sua linha de desenvolvimento na iteração seguinte. ▪ Avaliar, em conjunto com o CDI, o desempenho do(s) PRI que coordena. ▪ Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis. ▪ Rever, em conjunto com o GPI, a lista de artefactos a produzir na iteração seguinte, juntamente com as características técnicas de cada um, que deverão incluir: <ul style="list-style-type: none"> ○ <i>Aplicações</i>: Novas funcionalidades (descritas em texto livre, documentos ou diagramas), forma de disponibilização (on-line, instalação externa, DVD, etc.), requisitos de comunicação e imagem (interfaces gráficas, etc.); ○ <i>Documentos</i>: Língua(s), tópicos a abordar, formato (Word, Excel, PDF, etc.), requisitos de comunicação e imagem (modelos a utilizar, etc.). ▪ Verificar se pretende efectuar alguma alteração de férias que incida sobre a iteração seguinte e, nesse caso, notificar o GPI. ▪ Assegurar a produção e disponibilização da documentação necessária para suportar a execução do processo de passagem a certificação/produção dos artefactos aplicativos produzidos na iteração. ▪ [<i>Apenas no modelo reduzido</i>] Assegurar a disponibilização do necessário material de formação e de suporte.

4.2.8. [CII] Coordenador de Infra-Estrutura

O desenrolar de um projecto de desenvolvimento de software depende inexoravelmente, embora em medida variável, da existência da infra-estrutura necessária à sua implementação e disponibilização, que poderá variar entre as várias linhas de desenvolvimento que compõem o projecto. Contudo, e independentemente do(s) CEI envolvidos poderem possuir as competências necessárias para o fazer, o estabelecimento e gestão dessa infra-estrutura deverá estar a cargo de um terceiro – o CII – o que, para além de possibilitar que o(s) CEI se concentrem nas suas principais incumbências, permitirá assegurar a coerência¹⁵⁴ das infra-estruturas utilizadas pelas várias linhas de desenvolvimento e a existência da documentação adequada sobre a instalação e configuração dos componentes/sistemas desenvolvidos, concretizadas através das necessárias passagens a certificação/produção.

Adicionalmente, o contributo deste interveniente é também particularmente importante para garantir que a infra-estrutura se encontra dimensionada para fazer face aos requisitos não funcionais e aos níveis de serviço associados a cada linha de desenvolvimento, sem esquecer as preocupações de segurança inerentes à disponibilidade e fiabilidade da infra-estrutura. Finalmente, compete ao CII dotar a infra-estrutura do nível de monitorização necessário para aferir, a cada momento, o seu estado de funcionamento.

Responsável por ...

-
- Satisfazer as necessidades de infra-estrutura que lhe forem comunicadas pelo GPI.
 - Aconselhar o CDI sobre o conteúdo, formato e localização da documentação interna a produzir durante o projecto.
 - Identificar, estimar e comunicar ao GPI os riscos do projecto relacionados com a infra-estrutura (por exemplo, prováveis falhas externas no fornecimento de equipamento).
 - Assegurar a execução das passagens a certificação/produção agendadas com o GPI e documentadas pelo(s) CEI.
 - Zelar pela segurança, disponibilidade e fiabilidade da infra-estrutura, nomeadamente através da sua adequada configuração e da instalação dos *patches* pertinentes.
 - Zelar pelo adequado dimensionamento da infra-estrutura no sentido de suportar os requisitos não funcionais e os níveis de serviço contratados.
 - Zelar pelo cumprimento das suas responsabilidades por parte de cada uma das linhas de suporte exterior contratadas pela organização.
 - Identificar, em diálogo com o(s) CEI, os itens que deverão ser sujeitos a monitorização (activa e/ou passiva), para além de acordar a divisão de tarefas necessária para o concretizar
 - Notificar o GPI sempre que pretender proceder a uma alteração de férias
 - Manter um repositório de todo o software de base utilizado para estabelecer os ambientes necessários, juntamente com a documentação de todos os procedimentos necessários para a sua instalação e configuração.
 - Comunicar à equipa de manutenção todas as informações relevantes para que possam assegurar o futuro bom funcionamento da infra-estrutura (nos vários ambientes), identificando e descrevendo claramente as acções periódicas necessárias.

¹⁵⁴ Evitando a utilização (sem a devida justificação) de distintos sistemas operativos, servidores aplicativos, motores de base de dados, etc, o que tornará mais complexas as futuras actividades de manutenção sem aportar quaisquer vantagens.

Responsabilidades adicionais no modelo base	Responsabilidades adicionais no modelo reduzido
<ul style="list-style-type: none"> ▪ (o modelo base não implica quaisquer responsabilidades adicionais) 	<ul style="list-style-type: none"> ▪ Configurar e otimizar o funcionamento do(s) motor(es) de base de dados pelo(s) qual(is) é responsável. ▪ Conceber, em diálogo com o(s) respectivo(s) CEI, a política de cópias de segurança mais apropriada para cada um do(s) motor(es) de base de dados pelo(s) qual(is) é responsável.
Cardinalidade	Perfil
<ul style="list-style-type: none"> ▪ Um interveniente por projecto. 	<ul style="list-style-type: none"> ▪ Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil: <ul style="list-style-type: none"> ○ Formação de 2º ciclo de Bolonha (ou equivalente) em informática; ○ Pelo menos 2 (dois) anos de experiência na administração de sistemas; ○ Fortes conhecimentos sobre segurança de redes IP; ○ Experiência na utilização e configuração de protocolos e ferramentas de monitorização.
Participação em fluxos de informação internos	Participação em fluxos de informação externos
<ul style="list-style-type: none"> ▪ [Contingencial, <i>Apenas no modelo base</i>] ANI: No sentido de tomar conhecimento sobre os requisitos estabelecidos para a infra-estrutura e de validar a exequibilidade dos mesmos. ▪ [Contingencial] CDI: Com o objectivo de o aconselhar sobre as propostas do(s) CEI para o conteúdo, formato e localização da documentação interna a produzir durante o projecto no sentido de assegurar de forma apropriada as actividades futuras de manutenção. ▪ [Contingencial] CEI: No sentido de esclarecer eventuais dúvidas que surjam durante os processos de passagem a certificação/produção, de identificar os itens que deverão ser sujeitos a monitorização (activa e/ou passiva) e de acordar a divisão de tarefas necessária para o concretizar. ▪ [Preferencial] GPI: No sentido de tomar conhecimento sobre as necessidades de infra-estrutura do projecto e de o manter informado sobre o decorrer das suas actividades. 	<ul style="list-style-type: none"> ▪ [Contingencial] CIE: No sentido de não só prestar os esclarecimentos necessários à concepção e estruturação da infra-estrutura da respectiva entidade, como também colaborar na detecção e resolução de problemas de integração entre a infra-estrutura interna e externa.
O seu desempenho é crítico para ...	Deve evitar ...
<ul style="list-style-type: none"> ▪ Detectar requisitos não-exequíveis ou insuficientemente descritos. ▪ Garantir o cumprimento dos níveis de serviço contratados. 	<ul style="list-style-type: none"> ▪ Ter uma postura demasiadamente passiva, não tomando iniciativas no sentido de se informar sobre o funcionamento do software base a utilizar no projecto e esperando que lhe sejam transmitidas todas as informações de que necessita sobre o mesmo.

No início do projecto deve ...

- Satisfazer as necessidades de infra-estrutura que lhe forem comunicadas pelo GPI.
- Aconselhar o CDI sobre o conteúdo, formato e localização da documentação interna a produzir durante o projecto.
- Identificar, estimar e comunicar ao GPI os riscos do projecto relacionados com a infra-estrutura (por exemplo, prováveis falhas externas no fornecimento de equipamento).
- Identificar, em diálogo com o(s) CEI, os itens que deverão ser sujeitos a monitorização (activa e/ou passiva), para além de acordar a divisão de tarefas necessária para o concretizar.
- Publicar, em local definido pela organização, as seguintes informações:
 - Caracterização da infra-estrutura disponibilizada para suportar a coordenação, desenvolvimento e bom funcionamento do projecto (incluindo diagramas de rede, contas, directorias base, etc.);
 - Identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.
- Verificar se o seu calendário de férias se encontra actualizado e, caso não esteja, comunicar esse facto ao GPI.
- [*Apenas no modelo reduzido*] Conceber, em diálogo com o(s) respectivo(s) CEI, a política de cópias de segurança mais apropriada para cada um do(s) motor(es) de base de dados pelo(s) qual(is) é responsável.

No início de cada iteração deve ...

- Assegurar a execução das passagens a certificação/produção previamente agendadas com o GPI e documentadas pelo(s) CEI, notificando-os (juntamente com o CQI) aquando da sua conclusão.
- Caso tal lhe seja solicitado pelo GPI, concretizar as alterações necessárias à infra-estrutura do projecto.
- Verificar se já se encontram sujeitos a monitorização (activa e/ou passiva) todos os itens que, de acordo com o que foi previamente combinado, já se deveriam encontrar em funcionamento.
- Rever, em conjunto com o(s) CEI, os itens que deverão ser sujeitos a monitorização (activa e/ou passiva), para além de acordar a divisão de tarefas necessária para o concretizar.
- Verificar se pretende efectuar alguma alteração de férias que incida sobre a iteração seguinte e, nesse caso, notificar o GPI.
- [*Apenas no modelo reduzido*] Verificar que a política de cópias de segurança que definiu para cada um do(s) motor(es) de base de dados pelo(s) qual(is) é responsável se encontra a funcionar correctamente e/ou se é pertinente a concretização de alguma alteração à mesma.

Durante cada iteração deve ...

- Validar a exequibilidade dos requisitos estabelecidos para a infra-estrutura e, sempre que tal não se verifique, ajudar a encontrar uma solução alternativa viável e aceitável para as entidades externas.
- Detectar requisitos estabelecidos para a infra-estrutura que não se encontrem suficientemente detalhados, reportando-os ao GPI (e caso exista(m) ao(s) ANI).
- Zelar pela segurança, disponibilidade e fiabilidade da infra-estrutura, nomeadamente através da sua adequada configuração e da instalação dos *patches* pertinentes.
- Zelar pelo adequado dimensionamento da infra-estrutura no sentido de suportar os requisitos não funcionais e os níveis de serviço contratados.
- Manter um repositório de todo o software de base utilizado para estabelecer os ambientes necessários, juntamente com a documentação de todos os procedimentos necessários para a sua instalação e configuração.
- [*Apenas no modelo reduzido*] Configurar e otimizar o funcionamento do(s) motor(es) de base de dados pelo(s) qual(is) é responsável.

No final do projecto deve ...

- Verificar se já se encontram sujeitos a monitorização (activa e/ou passiva) todos os itens identificados.
- Comunicar ao CDI a sua avaliação do processo de desenvolvimento de software utilizado no projecto, sugerindo eventuais correcções ou evoluções.
- [*Apenas no modelo base*] Colaborar, sob coordenação do GPI, na realização de uma cópia de segurança (pelo menos em duplicado) de toda a informação relevante (configurações, servidores aplicativos, etc.) relacionada com a infra-estrutura do projecto.
- [*Apenas no modelo reduzido*] Colaborar, sob coordenação do GPI, na realização de uma cópia de segurança (pelo menos em duplicado) de toda a informação relevante (configurações, servidores aplicativos, *scripts* de inicialização de bases de dados, etc.) relacionada com a infra-estrutura e as bases de dados do projecto.
- Comunicar à equipa de manutenção todas as informações relevantes para que possam assegurar o futuro bom funcionamento da infra-estrutura (nos vários ambientes), identificando e descrevendo claramente as acções periódicas necessárias.
- Executar um simulacro para cada uma das linhas de suporte exterior contratadas pela organização, no sentido de verificar se respondem de acordo com as condições contratadas e se a organização possui todas as informações necessárias à sua activação.

No final de cada iteração deve ...

- Agendar com o GPI as passagens a certificação/produção necessárias para a iteração seguinte.
- Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis.

4.2.9. [CQI] Coordenador de Qualidade

Este interveniente é responsável pela concepção e operacionalização de um plano interno de auditoria de qualidade destinado a garantir que os artefactos (aplicacionais, documentais, etc.) produzidos ao longo do projecto se encontram de acordo com os requisitos identificados pelo GPI e/ou pelo(s) ANI. Para além disso, a sua intervenção é também essencial no sentido de evitar (na medida do possível) que as entidades externas tomem contacto com os artefactos (ou com determinadas características destes) sem que tenham sido previamente analisados por alguém que não os envolvidos na sua criação, por vários motivos: (1) porque a complexidade usualmente assumida pelos componentes/sistemas implementados carece de um demorado processo de testes funcionais que, por norma, o(s) CEI e PRI não têm oportunidade de realizar devido ao indesejável período de interregno no processo de desenvolvimento; (2) mesmo que o(s) CEI e PRI tenham oportunidade de conduzir alguns testes funcionais, é possível que os seus “vícios”¹⁵⁵ de utilização impeçam a detecção de determinados problemas. Assim, a contribuição do CQI é preponderante para a detecção de situações que, mesmo de pouca gravidade e fácil correcção, poderiam proporcionar uma má experiência de utilização às entidades externas, o que convém evitar a todo o custo; (3) o conjunto de artefactos documentais a produzir ao longo de um projecto carece usualmente da colaboração de um vasto número de intervenientes, por vezes até no âmbito de um mesmo documento. Contudo, e apesar de poderem existir recomendações internas nesse sentido, é vulgar que diferentes indivíduos se expressem de forma distinta, consubstanciada nomeadamente na formalidade da linguagem, nos termos e siglas empregues e na formatação visual dos documentos. Pelo exposto, é importante que este tipo de artefactos seja sujeito ao crivo de alguém com a preocupação de assegurar que os documentos primam pela correcção do seu conteúdo, são coerentes entre si (do ponto de vista visual, terminológico, etc.) e não deixam transparecer as idiossincrasias próprias dos vários intervenientes.

Para concretizar estes desígnios, o CQI conta com a colaboração de um ou mais AQI, cujas actividades deverá coordenar no sentido de assegurar que a qualidade percebida pelas entidades externas sobre os artefactos entregues é positiva e contribui para o reforço da imagem da organização.

¹⁵⁵ Por exemplo, a não introdução de valores que se encontrem fora do intervalo permitido para determinados parâmetros, a realização de operações de acordo com uma determinada sequência constante e de não de várias, etc.

Responsável por ...

- Zelar pela qualidade de todos os artefactos produzidos pela organização.
- Conceber, em diálogo com o GPI e o(s) CEI, o plano interno de auditoria de qualidade (cuja execução deverá coordenar), destinado a assegurar que os artefactos produzidos ao longo do projecto se encontram de acordo com os respectivos requisitos identificados pelo GPI e/ou pelo(s) ANI.
- Aprovar a entrega às entidades externas de todos os artefactos produzidos pela organização.
- Reportar periodicamente ao GPI e ao(s) CEI sobre o estado de cumprimento dos requisitos acordados com o cliente.
- Após se aconselhar com o(s) CEI, investigar a existência de metodologias e/ou ferramentas úteis para apoiar, da forma mais eficiente e eficaz possível, a execução do plano interno de auditoria de qualidade e, caso existam, envidar todos os esforços necessários à sua adopção.
- Coordenar, de forma tão eficiente e eficaz quanto possível, o trabalho do(s) AQI que lhe foram disponibilizados, atribuindo-lhe(s) as tarefas mais apropriadas ao seu perfil e evitando situações de dependência ociosa. Nesse sentido, deverão planear (com a antecedência possível) as actividades a desenvolver ao longo de cada iteração do projecto, definindo de forma atempada e inequívoca a responsabilidade de execução de cada uma.
- Aconselhar o CDI sobre o conteúdo, formato e localização da documentação interna a produzir durante o projecto, cujo cumprimento deverá verificar.
- Notificar o GPI sempre que pretender proceder a uma alteração de férias.

Cardinalidade

- Um interveniente por projecto.

Participação em fluxos de informação internos

- [Preferencial] AQI: No sentido de lhe(s) atribuir tarefas e de coordenar a execução das mesmas.
- [Contingencial] CDI: Com o objectivo de o aconselhar sobre as propostas do(s) CEI para o conteúdo, formato e localização da documentação interna a produzir durante o projecto no sentido de assegurar de forma apropriada as actividades futuras de manutenção.
- [Contingencial] CEI: No sentido de solicitar os esclarecimentos necessários à concepção e execução do plano interno de auditoria de qualidade sobre os artefactos produzidos pela respectiva linha de desenvolvimento e de lhe comunicar os resultados obtidos.
- [Preferencial] GPI: No sentido de o manter informado sobre o decorrer da execução do plano interno de auditoria de qualidade e de aprovar a entrega de cada artefacto.

- Assegurar que são observadas as boas práticas em vigor na organização no que concerne à criação de artefactos documentais, entre as quais poderão estar as seguintes:
 - Utilização de identificadores de documentos unívocos e normalizados;
 - Utilização de um sistema de gestão documental;
 - Utilização de modelos pré-definidos para determinados tipos de documentos.
- Assegurar que os artefactos documentais primam pela correcção do seu conteúdo, são coerentes entre si (do ponto de vista visual, terminológico, etc.) e não deixam transparecer as idiosincrasias próprias dos vários intervenientes.
- Assegurar que as cópias de segurança realizadas sobre a coordenação do GPI contêm toda a informação relevante (código fonte, artefactos, configurações, servidores aplicativos, *scripts* de inicialização de bases de dados, etc.) associada ao projecto.

Responsabilidades adicionais no modelo base

- (o modelo base não implica quaisquer responsabilidades adicionais)

Responsabilidades adicionais no modelo reduzido

- Zelar pela integridade das concretizações dos vários casos de uso que resultam do modelo de casos de uso, assegurando que cumprem integralmente os respectivos requisitos.

Perfil

- Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil:
 - Formação de 2º ciclo de Bolonha (ou equivalente) em informática;
 - Pelo menos 2 (dois) anos de experiência no desenvolvimento de software, nomeadamente utilizando métodos iterativos;
 - Conhecimento aprofundado da notação UML.

Participação em fluxos de informação externos

- (não é expectável a sua participação em fluxos de informação externos)

O seu desempenho é crítico para ...

- Criar e/ou reforçar uma imagem positiva da organização junto das entidades externas.
- Assegurar a motivação do(s) AQI que coordena.
- Optimizar o processo de avaliação dos artefactos no sentido de contribuir para uma célere entrega dos mesmos às entidades externas.

- Assegurar a detecção de situações que, mesmo de pouca gravidade e fácil correcção, poderiam proporcionar uma má experiência de utilização às entidades externas.

Deve evitar ...

- Hostilizar e/ou rejeitar a colaboração do(s) CEI.

No início do projecto deve ...

- Conceber, em diálogo com o GPI e o(s) CEI, o plano interno de auditoria de qualidade, destinado a assegurar que os artefactos produzidos ao longo do projecto se encontram de acordo com os respectivos requisitos identificados pelo GPI e/ou pelo(s) ANI.
- Após se aconselhar com o(s) CEI, investigar a existência de metodologias e/ou ferramentas úteis para apoiar, da forma mais eficiente e eficaz possível, a execução do plano interno de auditoria de qualidade e, caso existam, envidar todos os esforços necessários à sua adopção.
- Verificar se o seu calendário de férias se encontra actualizado e, caso não esteja, comunicar esse facto ao GPI.
- Publicar, em local definido pela organização, a identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.

- Aconselhar o CDI sobre o conteúdo, formato e localização da documentação interna a produzir durante o projecto.

No início de cada iteração deve ...

- Atribuir tarefas ao(s) AQI que coordena.
- Verificar se a documentação interna relativa à iteração anterior foi produzida, publicada e cumpre os requisitos estipulados pelo CDI.
- Coordenar a execução do plano interno de auditoria de qualidade definido para os artefactos produzidos na iteração anterior.
- Comunicar tão cedo quanto possível ao GPI e ao respectivo CEI a sua (des)aprovação sobre a entrega a entidades externas de cada um dos artefactos produzidos na iteração anterior.

Durante cada iteração deve ...

- Promover reuniões de ponto de situação semanais (máx. 30 min.) com o(s) AQI que coordena.
- Assegurar que são observadas as boas práticas em vigor na organização no que concerne à criação de artefactos documentais, entre as quais poderão estar as seguintes:
 - Utilização de identificadores de documentos unívocos e normalizados;
 - Utilização de um sistema de gestão documental;
 - Utilização de modelos pré-definidos para determinados tipos de documentos.

- Assegurar que os artefactos documentais primam pela correcção do seu conteúdo, são coerentes entre si (do ponto de vista visual, terminológico, etc.) e não deixam transparecer as idiosincrasias próprias dos vários intervenientes.
- [*Apenas no modelo reduzido*] Zelar pela integridade das concretizações dos vários casos de uso que resultam do modelo de casos de uso, assegurando que cumprem integralmente os respectivos requisitos.

No final do projecto deve ...

- Verificar se a documentação interna relativa à última iteração foi produzida, publicada e cumpre os requisitos estipulados pelo CDI.
- Coordenar a execução do plano interno de auditoria de qualidade definido para os artefactos produzidos na última iteração.
- Comunicar tão cedo quanto possível ao GPI e ao respectivo CEI a sua (des)aprovação sobre a entrega a entidades externas de cada um dos artefactos produzidos na última iteração.
- Assegurar que as cópias de segurança realizadas sobre a coordenação do GPI contêm toda a informação relevante (código fonte, artefactos, configurações, servidores aplicativos, *scripts* de inicialização de bases de dados, etc.) associada ao projecto.
- Comunicar ao CDI a sua avaliação do processo de desenvolvimento de software utilizado no projecto, sugerindo eventuais correcções ou evoluções.

- Garantir que o(s) AQI que coordena cumpriu(ram) as suas actividades de fecho de projecto.

No final de cada iteração deve ...

- Reportar ao GPI e ao(s) CEI o estado de cumprimento dos requisitos acordados com o cliente.
- Verificar se pretende efectuar alguma alteração de férias que incida sobre a iteração seguinte e, nesse caso, notificar o GPI.
- Garantir que o(s) AQI que coordena cumpriu(ram) as suas actividades de fecho de iteração.
- Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis.
- Verificar o calendário de férias do(s) AQI que coordena para o período previsto para a próxima iteração.

4.2.10. [CSI] Coordenador de Suporte

O sucesso de um projecto de desenvolvimento de software depende, em larga medida, da aceitação conseguida junto dos seus utilizadores finais. Assim, é especialmente importante o esforço realizado no sentido de tornar a experiência de utilização o mais agradável possível e para o qual a colaboração do CSI é especialmente importante, ao desenvolver vários tipos de acções: (1) compete-lhe preparar as acções de formação a ministrar aos utilizadores finais e zelar pela produção dos conteúdos de suporte (manuais, lista de perguntas frequentes, etc.) apropriados para os auxiliar na instalação, configuração e utilização dos componentes/sistemas desenvolvidos no âmbito do projecto; (2) é responsável pela estruturação e formação das diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas), que deverão estar aptas a responder às solicitações (via e-mail, telefone, etc.) dos utilizadores finais imediatamente a partir do momento em que os componentes/sistemas lhes são disponibilizados.

Responsável por ...

- | | |
|--|---|
| <ul style="list-style-type: none"> ▪ Identificar, em diálogo com o(s) CEI, o conteúdo, formato e localização da documentação de suporte a produzir durante o projecto por cada linha de desenvolvimento. ▪ Conceber, preparar e executar o plano de formação a ministrar aos utilizadores finais e às diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas). ▪ Definir a composição e estruturação adequadas para as diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas), que deverão estar aptas a responder às solicitações (via e-mail, telefone, etc.) dos utilizadores finais. ▪ Zelar pelo cumprimento das suas responsabilidades por parte de cada linha (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas). | <ul style="list-style-type: none"> ▪ Identificar e descrever o perfil dos utilizadores finais do projecto. ▪ Identificar e comunicar ao GPI as necessidades de infra-estrutura necessárias (endereços de e-mail, ferramentas CMS e/ou CRM, etc.) ao bom funcionamento das várias linhas de suporte internas. ▪ Observar as boas práticas em vigor na organização no que concerne à criação de artefactos documentais, entre as quais poderão estar as seguintes: <ul style="list-style-type: none"> ○ Utilização de identificadores de documentos unívocos e normalizados; ○ Utilização de um sistema de gestão documental; ○ Utilização de modelos pré-definidos para determinados tipos de documentos. ▪ Notificar o GPI sempre que pretender proceder a uma alteração de férias. |
|--|---|

Cardinalidade

- Este papel tem um cariz facultativo mas, quando adoptado, deverá ser desempenhado por um único indivíduo por projecto.

Perfil

- Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil:
 - Pelo menos 2 (dois) anos de experiência na prestação de suporte a utilizadores finais;
 - Conhecimentos de informática na óptica do utilizador.

Participação em fluxos de informação internos

- [Contingencial] ANI: No sentido de tomar conhecimento sobre os requisitos estabelecidos para o suporte, cuja exequibilidade lhe compete validar.
- [Contingencial] CEI: No sentido de conceber um apropriado plano de formação a ministrar aos utilizadores finais e às diferentes linhas das várias equipas de suporte e de identificar os conteúdos de suporte a produzir.
- [Preferencial] GPI: No sentido de lhe propor a estruturação e composição das diferentes linhas das várias equipas de suporte e o plano de formação a ministrar às mesmas e aos utilizadores finais.

O seu desempenho é crítico para ...

- Assegurar uma experiência agradável aos utilizadores finais no esclarecimento de eventuais questões sobre a instalação, configuração ou utilização dos componentes/sistemas desenvolvidos.
- Assegurar que as diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas) possuem a estrutura e formação necessária para responder adequadamente às solicitações (via e-mail, telefone, etc.) dos utilizadores finais.

No início do projecto deve ...

- Identificar e descrever o perfil dos utilizadores finais do projecto.
- Propor ao GPI a composição e estruturação adequadas para as diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas), que deverão estar aptas a responder às solicitações (via e-mail, telefone, etc.) dos utilizadores finais.
- Propor ao GPI, após dialogar com o(s) CEI, o plano de formação a ministrar aos utilizadores finais e às diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas).
- Identificar, em diálogo com o(s) CEI, o conteúdo, formato e localização da documentação de suporte a produzir durante o projecto por cada linha de desenvolvimento.
- Identificar e comunicar ao GPI as necessidades de infra-estrutura necessárias (endereços de e-mail, ferramentas CMS e/ou CRM, etc.) ao bom funcionamento das várias linhas de suporte internas.
- Publicar, em local definido pela organização, a identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.
- Verificar se o seu calendário de férias se encontra actualizado e, caso não esteja, comunicar esse facto ao GPI.

Participação em fluxos de informação externos

- [Contingencial] CSE: No sentido de prestar os esclarecimentos necessários à estruturação das linhas de suporte da respectiva entidade e de acordar a divisão de responsabilidades entre estas e as linhas de suporte internas à organização.

Deve evitar ...

- Que sejam o(s) CEI e PRI a produzir conteúdos de suporte que não careçam do seu conhecimento especializado.
- Que sejam o(s) CEI e PRI a tratar de pedidos de suporte que, pela sua natureza, podiam/deviam ser tratados por outra qualquer linha das várias equipas de suporte.

No início de cada iteração deve ...

- Identificar, em conjunto com o(s) CEI, os conteúdos necessários para uma eventual acção de formação a ministrar aos utilizadores finais e às diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas) no final da iteração.
- Verificar se a documentação de suporte relativa à iteração anterior foi produzida, publicada e cumpre os requisitos estipulados.
- Rever, em conjunto com o(s) CEI, a lista de conteúdos, formatos e localizações da documentação de suporte a produzir durante a iteração que se inicia.
- Verificar se pretende efectuar alguma alteração de férias que incida sobre a iteração seguinte e, nesse caso, notificar o GPI.

Durante cada iteração deve ...

- Preparar os conteúdos necessários para a execução do plano de formação a ministrar aos utilizadores finais e às diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas).
- Validar a exequibilidade dos requisitos estabelecidos para o suporte e, sempre que tal não se verifique, ajudar a encontrar uma solução alternativa viável e aceitável para as entidades externas.
- Zelar pelo cumprimento das suas responsabilidades por parte de cada linha (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas).
- Assegurar a produção do necessário material de formação e de suporte.
- Observar as boas práticas em vigor na organização no que concerne à criação de artefactos documentais, entre as quais poderão estar as seguintes:
 - Utilização de identificadores de documentos unívocos e normalizados;
 - Utilização de um sistema de gestão documental;
 - Utilização de modelos pré-definidos para determinados tipos de documentos.
- Detectar requisitos estabelecidos para o suporte que não se encontrem suficientemente detalhados, reportando-os ao GPI (e caso exista(m) ao(s) ANI).

No final do projecto deve ...

- Executar vários simulacros no sentido de verificar se as diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas) se encontram a funcionar eficazmente e de acordo com a estruturação definida.
- Verificar se a documentação de suporte relativa à última iteração foi produzida, publicada e cumpre os requisitos estipulados.
- Comunicar ao CDI a sua avaliação do processo de desenvolvimento de software utilizado no projecto, sugerindo eventuais correcções ou evoluções.

No final de cada iteração deve ...

- Assegurar a disponibilização aos utilizadores finais do necessário material de formação e de suporte.
- Caso se justifique, ministrar acções de formação aos utilizadores finais e às diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas) no sentido de os elucidar sobre as funcionalidades a disponibilizar no início da iteração seguinte.
- Rever se as necessidades de infra-estrutura necessárias (endereços de e-mail, ferramentas CMS e/ou CRM, etc.) ao bom funcionamento das várias linhas de suporte internas se mantêm na iteração seguinte e, caso contrário, comunicar ao GPI as alterações necessárias.
- Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis.
- Verificar se pretende efectuar alguma alteração de férias que incida sobre a iteração seguinte e, nesse caso, notificar o GPI.

4.2.11. [GPI] Gestor de Projecto

Em virtude da sua centralidade e leque de responsabilidades, o desempenho do GPI é preponderante para o sucesso de um projecto de desenvolvimento de software, de que é o principal responsável. Pode, por isso, dizer-se em traços gerais que lhe compete envidar todos os esforços necessários no sentido de assegurar o cumprimento integral e atempado das expectativas funcionais e qualitativas do cliente, sem ultrapassar os custos de produção inicialmente previstos (salvo situações imponderáveis devidamente justificadas). Logo, cabe-lhe a coordenação geral do projecto, orientando as actividades dos restantes elementos de acordo com as prioridades que estabelece, que resultam dos compromissos por si assumidos perante entidades externas. Assim, deve ser o ponto de contacto preferencial entre o interior e o exterior da organização, de forma a garantir que se encontra constantemente informado sobre todos os eventos relevantes, nomeadamente os relacionados com a gestão de âmbito do projecto (e possíveis implicações comerciais). Para isso, espera-se que participe activamente no processo de levantamento de requisitos, responsabilizando-se por assegurar que estes são transmitidos à restante equipa de projecto de uma forma clara e sistematizada. Contudo, dada a complexidade que este processo poderá assumir em projectos de grande envergadura, o GPI poderá necessitar da colaboração de um ou mais ANI, cuja coordenação lhe compete. Em conjunto, deverão ainda assegurar um eficaz controlo sobre a concretização dos vários requisitos, no sentido de permitir a qualquer momento saber a que distância o projecto se encontra do seu final. Caso a organização disponha de uma ferramenta de gestão de requisitos, cabe-lhe garantir que o projecto se encontra devidamente configurado e zelar pela contínua actualização do conjunto de requisitos registados na mesma.

Um outro aspecto importante da sua intervenção consiste na mediação das relações entre os restantes intervenientes, no sentido de ultrapassar falhas de comunicação e gerir de forma adequada a tensão implícita que usualmente as caracteriza. Como exemplo, destaco não só a relação entre o(s) CEI (responsáveis pela coordenação das actividades de produção dos artefactos) e o CQI (responsável por aferir a qualidade destes), como também entre o(s) CEI e o CII (responsável por disponibilizar e gerir a infra-estrutura de que os primeiros necessitam para desenvolver os sistemas pelo quais se encontram responsáveis). Por último, convém referir que, dada a natureza das suas incumbências, não é essencial que o GPI possua aprofundados conhecimentos técnicos, ausência que poderá ser suprida através da colaboração do(s) CEI e, caso exista(m), do(s) ANI.

Responsável por ...

- Avaliar as propostas de cada CEI para o número e duração das iterações a efectuar, o conteúdo das mesmas e respectivos artefactos associados a cada linha de desenvolvimento, que deverão incluir:
 - *Aplicações*: Novas funcionalidades (descritas em texto livre, documentos ou diagramas), forma de disponibilização (on-line, instalação externa, DVD, etc.), requisitos de comunicação e imagem (interfaces gráficas, etc.);
 - *Documentos*: Língua(s), tópicos a abordar, formato (Word, Excel, PDF, etc.), requisitos de comunicação e imagem (modelos a utilizar, etc.).
- Depois de acertar o conteúdo das iterações a realizar com o(s) CEI, elaborar um plano de alto nível e submetê-lo à aprovação do(s) PPI e, posteriormente das entidades externas. Este plano deverá:
 - Identificar explicitamente os momentos de entrega de cada um dos artefactos (aplicacionais e não só) a disponibilizar pela organização durante o projecto;
 - Identificar explicitamente os compromissos da organização que se encontram dependentes de acções a realizar por entidades externas;
 - Servir de suporte para a monitorização (interna e externa) da evolução do projecto.
- Obter comentários oficiais atempados das entidades externas sobre todos os artefactos produzidos pela organização.
- Obter comentários (oficiais ou officiosos) periódicos das entidades externas sobre a evolução do projecto.
- Zelar por uma célere aceitação do projecto e pela satisfação das entidades externas em relação ao desempenho da organização, promovendo a sua avaliação.
- Controlar preventivamente o cumprimento dos objectivos da iteração e, quando detectar um obstáculo à sua concretização, avaliar com o CDI e o com o respectivo CEI se este pode ser ultrapassado através da reafectação de recursos. Quando tal não for possível, decidir (eventualmente em diálogo com as entidades externas) se a iteração deve terminar no prazo previsto (concretizando os objectivos possíveis) ou se deve ser prolongada até que todos os objectivos sejam atingidos.
- Formalizar internamente os pedidos de alteração recebidos de entidades externas, submetendo-os rapidamente à apreciação do(s) CEI afectados pelos mesmos.
- Avaliar as necessidades de infra-estrutura do projecto identificadas pelo CSI, CQI e pelo(s) CEI (nos vários ambientes a estabelecer), formalizando o seu pedido ao CII e assegurando-se que as mesmas são atempadamente satisfeitas por este.
- Centralizar a comunicação oficial (relacionada com o projecto) entre a organização e as entidades externas.
- Assegurar a facturação (total ou fraccionada) do projecto imediatamente após se encontrarem reunidas as condições necessárias para o fazer.
- Caso a organização disponha de uma ferramenta de gestão de requisitos, cabe-lhe garantir que o projecto se encontra devidamente configurado e zelar pela contínua actualização do conjunto de requisitos registados na mesma.
- Zelar pelo cumprimento do plano de alto nível, assegurando a entrega atempada às entidades externas dos artefactos produzidos.
- Zelar pelo cumprimento das suas responsabilidades por parte de todos os intervenientes externos, avisando-os imediatamente sempre que as suas falhas fizerem perigar os compromissos assumidos pela organização e, caso algum dos PPI possua uma relação privilegiada com a entidade em causa, solicitar a sua colaboração.
- Identificar, em diálogo com os restantes intervenientes internos, os factores de risco do projecto, efectuar a sua gestão e tomar iniciativas no sentido de os mitigar ou eliminar.
- Manter, em colaboração com o(s) CEI, um registo dos eventos mais significativos relacionados com o desenvolvimento do projecto (por exemplo, atrasos protagonizados por entidades externas, aceitação dos artefactos, etc.).
- No caso de uma alteração ser classificada por um CEI como *complexa*, deverá decidir se pretende:
 - Que a iteração actual se prolongue até se atingir esse objectivo;
 - Que a iteração actual termine na altura prevista com as funcionalidades originais e a alteração passe para a iteração seguinte;
 - Negociar com o cliente a inclusão da alteração em detrimento de uma funcionalidade prevista para a iteração actual que ainda não tenha sido implementada.
- Elaborar, a partir dos dados sobre a alocação de recursos disponíveis na organização, relatórios periódicos sobre o consumo de recursos verificado, comparando-o com o consumo de recursos previamente estimado (quer até ao momento quer para a totalidade do projecto).
- Assegurar que todas as entidades externas envolvidas no projecto nomeiam o seu responsável oficial pelo mesmo, que deverá ter poder de decisão para assumir compromissos em nome da respectiva entidade e para aprovar os artefactos entregues.
- Zelar pela contenção dos custos incorridos pela organização, nomeadamente através da minimização da necessidade de deslocação dos seus intervenientes (por exemplo, promovendo conferências telefónicas em detrimento de reuniões presenciais).
- Assegurar que no final do projecto é efectuada uma cópia de segurança (pelo menos em duplicado) de toda a informação relevante (código fonte, artefactos, etc.) associada ao mesmo.
- Observar as boas práticas em vigor na organização no que concerne à criação de artefactos documentais, entre as quais poderão estar as seguintes:
 - Utilização de identificadores de documentos unívocos e normalizados;
 - Utilização de um sistema de gestão documental;
 - Utilização de modelos pré-definidos para determinados tipos de documentos.
- Comunicar ao CDI a sua avaliação das competências comportamentais do(s) CEI envolvidos no projecto.

Responsável por ... (cont.)

- Executar todas as acções importantes para o sucesso do projecto que não se encontrem inequivocamente atribuídas a nenhum interveniente ou que, apesar disso, carecem de uma operacionalização mais rápida do que este pode assegurar (por, por exemplo, se encontrar ausente).
- Desenvolver iniciativas, em coordenação com o(s) PPI, no sentido de explorar eventuais oportunidades comerciais viáveis (associadas ao projecto) detectadas.
- Mediar situações de conflito entre quaisquer dos restantes intervenientes.

Responsabilidades adicionais no modelo reduzido

- Coordenar o processo de levantamento de requisitos (funcionais e não funcionais) de forma a produzir a documentação (entre a qual o modelo de casos de uso) necessária à delimitação do âmbito do projecto de uma forma tão inequívoca quanto possível.
- Validar internamente a exequibilidade dos requisitos identificados e, sempre que tal não se verifique, promover o diálogo interno com o objectivo de encontrar uma solução alternativa viável e aceitável para as entidades externas.
- Detectar eventuais novos requisitos apresentados por entidades externas e formalizar internamente os respectivos pedidos de alteração, submetendo-os rapidamente à apreciação do(s) CEI afectados pelos mesmos.
- Detectar possíveis requisitos que, não tendo sido explicitados pelas entidades externas, podem encontrar-se dentro das suas expectativas ou correspondam a uma oportunidade comercial que possa ser aproveitada.
- Envidar todos os esforços no sentido de esclarecer requisitos que não tenham sido suficientemente detalhados pelas entidades externas, no sentido de alinhar as expectativas de ambas as partes.
- Conceber, preparar e coordenar a execução do plano de formação a ministrar aos utilizadores finais e às diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas).
- Definir a composição e estruturação adequadas para as diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas), que deverão estar aptas a responder às solicitações (via e-mail, telefone, etc.) dos utilizadores finais.
- Identificar, em diálogo com o(s) CEI, o conteúdo, formato e localização da documentação de suporte a produzir durante o projecto por cada linha de desenvolvimento.

Cardinalidade

- Um interveniente por projecto.

- Zelar pelo cumprimento das suas responsabilidades por parte de cada linha (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas).
- Identificar e descrever o perfil dos utilizadores finais do projecto.
- Identificar as necessidades de infra-estrutura necessárias (endereços de e-mail, ferramentas CMS e/ou CRM, etc.) ao bom funcionamento das várias linhas de suporte internas.

Responsabilidades adicionais no modelo base

- Coordenar, de forma tão eficiente e eficaz quanto possível, o trabalho do(s) ANI que lhe foram disponibilizados, atribuindo-lhe(s) as tarefas mais apropriadas ao seu perfil e evitando situações de dependência ociosa. Nesse sentido, deverão planejar (com a antecedência possível) as actividades a desenvolver ao longo de cada iteração do projecto, definindo de forma atempada e inequívoca a responsabilidade de execução de cada uma.
- Avaliar a documentação (entre a qual o modelo de casos de uso) produzida pelo(s) ANI no sentido de assegurar que delimita o âmbito do projecto de uma forma tão inequívoca quanto possível.
- Avaliar a proposta do CSI para o plano de formação a ministrar aos utilizadores finais e às diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas).
- Avaliar a proposta do CSI para a composição e estruturação adequadas para as diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas), que deverão estar aptas a responder às solicitações (via e-mail, telefone, etc.) dos utilizadores finais.

Perfil

- Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil:
 - Formação de 2º ciclo de Bolonha (ou equivalente) em informática, complementada com um 2º ciclo em ciências empresariais;
 - Pelo menos 2 (dois) anos de experiência no desenvolvimento de software, nomeadamente utilizando métodos iterativos;
 - Conhecimento sobre a área de negócio em que o projecto irá decorrer;
 - Elevada capacidade negocial e de decisão;
 - Conhecimento básico da notação UML (diagramas de casos de uso).

Participação em fluxos de informação internos

- [Preferencial, *apenas no modelo base*] ANI: No sentido de tomar conhecimento sobre os requisitos identificados e eventuais oportunidades de negócio detectadas.
- [Contingencial, *apenas no modelo base*] ASI: No sentido de tomar conhecimento sobre os riscos técnicos do projecto decorrentes das decisões arquiteturais tomadas.
- [Preferencial, *apenas no modelo base*] CCI: No sentido de lhe transmitir as necessidades de comunicação e imagem do projecto e se manter informado sobre o decorrer das suas actividades.
- [Contingencial] CDI: No sentido de o manter informado sobre a prestação do(s) CEI envolvidos no projecto.
- [Preferencial] CEI: No sentido de se manter informado sobre o decorrer das actividades da linha de desenvolvimento que coordena.
- [Preferencial] CII: No sentido de lhe transmitir as necessidades de infra-estrutura do projecto e se manter informado sobre o decorrer das suas actividades.
- [Preferencial] CQI: No sentido de se manter informado sobre o decorrer da execução do plano interno de auditoria de qualidade e de obter a sua aprovação para a entrega de cada artefacto.

O seu desempenho é crítico para ...

- Proteger a equipa de projecto de pressões externas que possam condicionar o seu desempenho.
- Assegurar, tão rapidamente quanto possível, a aceitação do projecto, no sentido de libertar os recursos envolvidos.
- Garantir que alterações aos requisitos iniciais do projecto são devidamente analisadas, quer com entidades externas quer com os intervenientes internos, antes de se passar à sua concretização.
- Assegurar a motivação dos elementos da equipa de projecto.
- Assegurar tão cedo quanto possível a facturação do projecto.
- Assegurar que as entidades externas avaliam tão cedo quanto possível os artefactos entregues pela organização, no sentido de:
 - Mensurar a satisfação das entidades externas;
 - Detectar rapidamente eventuais diferenças de expectativas;
 - Minimizar a entropia que resulta de analisar os comentários das entidades externas a artefactos entregues pela organização há algum tempo atrás;
 - Minimizar o risco de desperdiçar trabalho aplicado em características não valorizadas ou desejadas pelas entidades externas;
 - Minimizar o risco de alguma entidade externa vir a questionar posteriormente o cumprimento dos compromissos assumidos pela organização e, consequentemente, a aceitação do projecto.
- Assegurar que a organização persegue novas oportunidades comerciais (associadas ao projecto) consideradas interessantes.

- [Preferencial, *apenas no modelo base*] CSI: No sentido de tomar conhecimento sobre a sua proposta para a estruturação e composição das diferentes linhas das várias equipas de suporte e o plano de formação a ministrar às mesmas e aos utilizadores finais.
- [Preferencial] PPI: No sentido de lhe transmitir os requisitos identificados, o plano de alto nível do projecto e concertar com ele as suas acções.

Participação em fluxos de informação externos

- [Preferencial] GPE: No sentido de:
 - Estabelecer o canal oficial de comunicação com a respectiva entidade externa;
 - Acordar o plano de alto nível do projecto;
 - Entregar os artefactos produzidos;
 - Obter aprovação atempada dos artefactos entregues;
 - [*Apenas no modelo reduzido*] Obter as informações necessárias ao processo de levantamento de requisitos.

- Assegurar que os CEI têm conhecimento sobre o desenrolar das actividades das restantes linhas de desenvolvimento e sobre as dependências que existem entre as mesmas.
- [*Apenas no modelo reduzido*] Ajudar as entidades externas a identificar as suas reais necessidades e, consequentemente, clarificar a totalidade dos requisitos a observar.
- [*Apenas no modelo reduzido*] Detectar requisitos não exequíveis e, para cada um, encontrar uma solução alternativa viável e aceitável para as entidades externas.

Deve evitar ...

- Aceitar alterações aos requisitos iniciais sem estimar o seu impacto e, quando este for negativo, sem negociar contrapartidas (custo adicional, adiamento ou cancelamento de outros requisitos, etc.).
- Comunicar factos ou solicitar acções ao(s) PRI que não sejam do conhecimento do CEI que coordena o seu trabalho.
- Comunicar factos ou solicitar acções ao(s) AQI que não sejam do conhecimento do CQI que coordena o seu trabalho.
- Proceder a entregas de artefactos a entidades externas sem assegurar a prévia aprovação dos mesmos por parte do CQI.
- Influenciar decisões (o que não significa que não possam efectuar sugestões) meramente técnicas, da competência do CDI, do(s) ASI e do(s) CEI.
- [*Apenas no modelo reduzido*] Desresponsabilizar-se da clarificação dos requisitos a observar alegando que as entidades externas não sabem as suas reais necessidades, já que é sua responsabilidade ajudar a clarificá-las.

No início do projecto deve ...

- Conduzir uma reunião de início de projecto com todos os intervenientes internos, no sentido de lhes apresentar o seu âmbito, a sua inserção na estratégia da organização e o que se espera dos participantes.
- Identificar, em diálogo com os restantes intervenientes internos, os factores de risco do projecto, definindo as iniciativas que deverão ser tomadas no sentido de os mitigar ou eliminar.
- Depois de acertar o conteúdo das iterações a realizar com o(s) CEI, elaborar um plano de alto nível e submetê-lo à aprovação do(s) PPI e, posteriormente das entidades externas. Este plano deverá:
 - Identificar explicitamente os momentos de entrega de cada um dos artefactos (aplicacionais e não só) a disponibilizar pela organização durante o projecto;
 - Identificar explicitamente os compromissos da organização que se encontram dependentes de acções a realizar por entidades externas;
 - Servir de suporte para a monitorização (interna e externa) da evolução do projecto.
- Verificar o calendário de férias do ABI, do(s) ANI, do(s) ASI, do(s) CEI, do CQI, do CII e do CSI para o período previsto de execução do projecto.
- Avaliar as propostas de cada CEI para o número e duração das iterações a efectuar, o conteúdo das mesmas e respectivos artefactos associados a cada linha de desenvolvimento, que deverão incluir:
 - *Aplicações*: Novas funcionalidades (descritas em texto livre, documentos ou diagramas), forma de disponibilização (on-line, instalação externa, DVD, etc.), requisitos de comunicação e imagem (interfaces gráficas, etc.);
 - *Documentos*: Língua(s), tópicos a abordar, formato (Word, Excel, PDF, etc.), requisitos de comunicação e imagem (modelos a utilizar, etc.).
- Avaliar as necessidades de infra-estrutura do projecto identificadas pelo CSI, CQI e pelo(s) CEI (nos vários ambientes a estabelecer), formalizando o seu pedido ao CII e assegurando-se que as mesmas são atempadamente satisfeitas por este.
- Publicar, em local definido pela organização, as seguintes informações:
 - Identificação de todos os elementos da organização envolvidos no projecto e do respectivo papel;
 - Lista dos factores de risco do projecto;
 - Identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.
- Comunicar ao(s) PPI o conjunto de requisitos (funcionais e não funcionais) identificados e respectiva documentação associada (entre a qual o modelo de casos de uso).
- Assegurar que todas as entidades externas envolvidas no projecto nomeiam o seu responsável oficial pelo mesmo, que deverá ter poder de decisão para assumir compromissos em nome da respectiva entidade e para aprovar os artefactos entregues.
- [*Apenas no modelo base*] Avaliar a documentação (entre a qual o modelo de casos de uso) produzida pelo(s) ANI no sentido de assegurar que delimita o âmbito do projecto de uma forma tão inequívoca quanto possível.
- [*Apenas no modelo base*] Identificar, em conjunto com o CCI, a lista das necessidades de imagem e comunicação previstas para o projecto.
- [*Apenas no modelo base*] Avaliar a proposta do CSI para o plano de formação a ministrar aos utilizadores finais e às diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas).
- [*Apenas no modelo base*] Avaliar a proposta do CSI para a composição e estruturação adequadas para as diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas), que deverão estar aptas a responder às solicitações (via e-mail, telefone, etc.) dos utilizadores finais.
- [*Apenas no modelo reduzido*] Identificar e descrever o perfil dos utilizadores finais do projecto.
- [*Apenas no modelo reduzido*] Definir, após dialogar com o(s) CEI, o plano de formação a ministrar aos utilizadores finais e às diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas).
- [*Apenas no modelo reduzido*] Definir a composição e estruturação adequadas para as diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas), que deverão estar aptas a responder às solicitações (via e-mail, telefone, etc.) dos utilizadores finais.
- [*Apenas no modelo reduzido*] Identificar, em diálogo com o(s) CEI, o conteúdo, formato e localização da documentação de suporte a produzir durante o projecto por cada linha de desenvolvimento.
- [*Apenas no modelo reduzido*] Identificar as necessidades de infra-estrutura necessárias (endereços de e-mail, ferramentas CMS e/ou CRM, etc.) ao bom funcionamento das várias linhas de suporte internas.
- [*Apenas no modelo reduzido*] Coordenar o processo de levantamento de requisitos (funcionais e não funcionais) de forma a produzir a documentação (entre a qual o modelo de casos de uso) necessária à delimitação do âmbito do projecto de uma forma tão inequívoca quanto possível.

No início de cada iteração deve ...

- Rever, em conjunto com os restantes intervenientes, a lista de factores de risco do projecto, definindo as iniciativas que deverão ser tomadas no sentido de os mitigar ou eliminar.
- Assegurar, após a aprovação do CQI, a entrega atempada às entidades externas dos artefactos produzidos na iteração anterior.
- Obter comentários oficiais das entidades externas sobre todos os artefactos entregues pela organização na iteração anterior.
- Elaborar, a partir dos dados sobre a alocação de recursos relativos às iterações anteriores, um relatório sobre o consumo de recursos verificado, comparando-o com o consumo de recursos previamente estimado (quer até ao momento quer para a totalidade do projecto).
- [Apenas no modelo base] Rever, em conjunto com o CCI, a lista das necessidades de imagem e comunicação previstas para a iteração que se inicia.
- [Apenas no modelo base] Atribuir tarefas ao(s) ANI que coordena.
- [Apenas no modelo base] Validar com o(s) ANI se documentação de requisitos carece de actualização.
- [Apenas no modelo reduzido] Identificar, em conjunto com o(s) CEI, os conteúdos necessários para uma eventual acção de formação a ministrar aos utilizadores finais e às diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas) no final da iteração.
- [Apenas no modelo reduzido] Verificar se a documentação de suporte relativa à iteração anterior foi produzida, publicada e cumpre os requisitos estipulados.
- [Apenas no modelo reduzido] Rever, em conjunto com o(s) CEI, a lista de conteúdos, formatos e localizações da documentação de suporte a produzir durante a iteração que se inicia.

Durante cada iteração deve ...

- Zelar pelo cumprimento das suas responsabilidades por parte de todos os intervenientes externos, avisando-os imediatamente sempre que as suas falhas fizerem perigar os compromissos assumidos pela organização e, caso algum dos PPI possua uma relação privilegiada com a entidade em causa, solicitar a sua colaboração.
- Obter comentários (oficiais ou officiosos) periódicos das entidades externas sobre a evolução do projecto.
- Controlar preventivamente o cumprimento dos objectivos da iteração e, quando detectar um obstáculo à sua concretização, avaliar com o CDI e o com o respectivo CEI se este pode ser ultrapassado através da reafectação de recursos. Quando tal não for possível, decidir (eventualmente em diálogo com as entidades externas) se a iteração deve terminar no prazo previsto (concretizando os objectivos possíveis) ou se deve ser prolongada até que todos os objectivos sejam atingidos.
- Manter, em colaboração com o(s) CEI, um registo dos eventos mais significativos relacionados com o desenvolvimento do projecto (por exemplo, atrasos protagonizados por entidades externas, aceitação dos artefactos, etc.).
- Promover reuniões de ponto de situação semanais (máx. 30 min.) com o(s) CEI envolvidos no projecto.
- Formalizar internamente os pedidos de alteração recebidos de entidades externas, submetendo-os rapidamente à apreciação do(s) CEI afectados pelos mesmos.
- Analisar a viabilidade de exploração de oportunidades comerciais (associadas ao projecto) detectadas e, quando se justificar, cooperar com o(s) PPI para que sejam aproveitadas pela organização.
- Zelar pela contenção dos custos incorridos pela organização, nomeadamente através da minimização da necessidade de deslocação dos seus intervenientes (por exemplo, promovendo conferências telefónicas em detrimento de reuniões presenciais).
- No caso de uma alteração ser classificada por um CEI como *complexa*, deverá decidir se pretende:
 - Que a iteração actual se prolongue até se atingir esse objectivo;
 - Que a iteração actual termine na altura prevista com as funcionalidades originais e a alteração passe para a iteração seguinte;
 - Negociar com o cliente a inclusão da alteração em detrimento de uma funcionalidade prevista para a iteração actual que ainda não tenha sido implementada.
- Executar todas as acções importantes para o sucesso do projecto que não se encontrem inequivocamente atribuídas a nenhum interveniente ou que, apesar disso, carecem de uma operacionalização mais rápida do que este pode assegurar (por, por exemplo, se encontrar ausente).
- Mediar situações de conflito entre quaisquer dos restantes intervenientes.
- [Apenas no modelo base] Promover reuniões de ponto de situação semanais (máx. 30 min.) com o(s) ANI que coordena.
- [Apenas no modelo reduzido] Detectar eventuais novos requisitos apresentados por entidades externas e formalizar internamente os respectivos pedidos de alteração, submetendo-os rapidamente à apreciação do(s) CEI afectados pelos mesmos.
- [Apenas no modelo reduzido] Detectar possíveis requisitos que, não tendo sido explicitados pelas entidades externas, podem encontrar-se dentro das suas expectativas ou correspondam a uma oportunidade comercial que possa ser aproveitada.
- [Apenas no modelo reduzido] Validar internamente a exequibilidade dos requisitos identificados e, sempre que tal não se verifique, promover o diálogo interno com o objectivo de encontrar uma solução alternativa viável e aceitável para as entidades externas.

Durante cada iteração deve ... (cont.)

- Zelar pela contenção dos custos incorridos pela organização, nomeadamente através da minimização da necessidade de deslocação dos seus intervenientes (por exemplo, promovendo conferências telefónicas em detrimento de reuniões presenciais).
- Observar as boas práticas em vigor na organização no que concerne à criação de artefactos documentais, entre as quais poderão estar as seguintes:
 - Utilização de identificadores de documentos unívocos e normalizados;
 - Utilização de um sistema de gestão documental;
 - Utilização de modelos pré-definidos para determinados tipos de documentos.

- [*Apenas no modelo reduzido*] Evitar todos os esforços no sentido de esclarecer requisitos que não tenham sido suficientemente detalhados pelas entidades externas, no sentido de alinhar as expectativas de ambas as partes.
- [*Apenas no modelo reduzido*] Zelar pelo cumprimento das suas responsabilidades por parte de cada linha (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas).
- [*Apenas no modelo reduzido*] Assegurar a produção do necessário material de formação e de suporte.

No final do projecto deve ...

- Assegurar que o projecto foi facturado na sua totalidade.
- Assegurar que é efectuada uma cópia de segurança (pelo menos em duplicado) de toda a informação relevante (código fonte, artefactos, configurações, servidores aplicativos, *scripts* de inicialização de bases de dados, etc.) associada ao projecto, sendo que as cópias deverão ser distribuídas por localizações distintas.
- Solicitar às entidades externas o preenchimento de um formulário de avaliação do desempenho da organização que, após preenchido, deverá fazer chegar até à equipa directiva.
- Conduzir uma reunião de encerramento de projecto com todos os intervenientes internos, no sentido de realizar um balanço dos acontecimentos, transmitindo-lhes os comentários recolhidos de entidades externas e procurando retirar ensinamentos úteis.
- Garantir que CII, CSI, CEI e CQI cumpriram as suas actividades de fecho de projecto.
- Comunicar ao CDI a sua avaliação das competências comportamentais do(s) CEI envolvidos no projecto.
- Comunicar ao CDI a sua avaliação do processo de desenvolvimento de software utilizado no projecto, sugerindo eventuais correcções ou evoluções.
- [*Apenas no modelo reduzido*] Verificar se a documentação de suporte relativa à última iteração foi produzida, publicada e cumpre os requisitos estipulados.
- [*Apenas no modelo reduzido*] Executar vários simulacros no sentido de verificar se as diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas) se encontram a funcionar eficazmente e de acordo com a estruturação definida.
- [*Apenas no modelo reduzido*] Verificar se a documentação de suporte relativa à última iteração foi produzida, publicada e cumpre os requisitos estipulados.

No final de cada iteração deve ...

- Avaliar, em conjunto com as entidades externas, a necessidade de efectuar alterações ao plano de alto nível, de forma a rever com o(s) CEI a lista de artefactos a produzir na iteração seguinte.
- Verificar se se encontram reunidas as condições necessárias para proceder à facturação de mais uma fracção do projecto e, em caso afirmativo, agir no sentido de o assegurar.
- Agendar com o CII, em consonância com o(s) CEI e CQI, as passagens a certificação/produção necessárias para a iteração seguinte.
- Avaliar, em conjunto com o CDI e com o(s) ASI, o desempenho do(s) CEI.
- Verificar o calendário de férias do ABI, do(s) ANI, do(s) ASI, do(s) CEI, do CQI, do CII e do CSI para o período previsto para a próxima iteração.
- Garantir que CII, CSI, CEI e CQI cumpriram as suas actividades de fecho de iteração.
- Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis.
- [*Apenas no modelo reduzido*] Assegurar a disponibilização aos utilizadores finais do necessário material de formação e de suporte.
- [*Apenas no modelo reduzido*] Caso se justifique, assegurar a execução de acções de formação aos utilizadores finais e às diferentes linhas (atendimento geral, triagem e laboratório) das várias equipas de suporte (sejam estas internas ou externas) no sentido de os elucidar sobre as funcionalidades a disponibilizar no início da iteração seguinte.
- [*Apenas no modelo reduzido*] Rever se as necessidades de infra-estrutura necessárias (endereços de e-mail, ferramentas CMS e/ou CRM, etc.) ao bom funcionamento das várias linhas de suporte internas se mantêm na iteração seguinte e, caso contrário, solicitar as alterações necessárias.

4.2.12. [PPI] Patrocinador de Projecto

Em traços gerais, a participação deste interveniente pode ser resumida a duas vertentes essenciais: em primeiro lugar, deverá aproveitar a sua influência interna e externa no sentido de se tornar um agente facilitador do projecto e de assim contribuir para a criação das necessárias condições de sucesso. Por outro lado, espera-se que, para além de acompanhar e aconselhar o GPI no decurso da sua intervenção, escrutine e avalie o seu desempenho.

Responsável por ...

- | | |
|--|--|
| <ul style="list-style-type: none"> ▪ Tirando partido da sua influência junto de uma determinada entidade externa, intervir (de forma concertada com o GPI) no sentido de: <ul style="list-style-type: none"> ○ Minimizar os efeitos de eventuais problemas ocorridos no âmbito da implementação do projecto (por exemplo, o incumprimento de um prazo); ○ Desbloquear situações que dela dependam e que prejudiquem ou impeçam o normal desenrolar do projecto (por exemplo, ausência de tomada de decisões importantes); ○ Negociar concessões importantes para a organização (por exemplo, a dilatação de um prazo de execução); ○ Explorar eventuais oportunidades comerciais associadas ao projecto (por exemplo, adição de novas funcionalidades); ○ Negociar compensações devidas por incumprimento das suas responsabilidades (por exemplo, por atraso significativo na entrega de equipamento); ○ Assegurar a nomeação de um GPE com os poderes e condições necessárias para o bom desempenho da função. ▪ Analisar o conjunto de requisitos (funcionais e não funcionais) identificados e respectiva documentação associada (entre a qual o modelo de casos de uso), comunicados pelo GPI. | <ul style="list-style-type: none"> ▪ Tirando partido da sua influência no interior da organização, intervir (de forma concertada com o GPI) no sentido de: <ul style="list-style-type: none"> ○ Assegurar que são disponibilizados os recursos (humanos ou outros) necessários ao normal desenrolar do projecto; ○ Assegurar a satisfação atempada de requisitos de elevada importância para uma entidade externa; ○ Assegurar a execução atempada das passagens a certificação/produção. ▪ Aprovar, mediante proposta do GPI, o plano de alto nível do projecto que deverá: <ul style="list-style-type: none"> ○ Identificar explicitamente os momentos de entrega de cada um dos artefactos (aplicacionais e não só) a disponibilizar pela organização durante o projecto; ○ Identificar explicitamente os compromissos da organização que se encontram dependentes de acções a realizar por entidades externas; ○ Servir de suporte para a monitorização (interna e externa) da evolução do projecto. |
|--|--|

Cardinalidade

- Este papel tem um cariz obrigatório, podendo ser desempenhado em simultâneo por vários indivíduos num mesmo projecto, se isso possibilitar o exercício de uma influência mais significativa ou abrangente junto das várias entidades externas.

Perfil

- Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil:
 - Conhecimento sobre a área de negócio em que o projecto irá decorrer;
 - Elevada capacidade de decisão;
 - Elevada capacidade negocial;
 - Elevada tolerância à pressão.

Participação em fluxos de informação internos

- [Preferencial] GPI: No sentido de tomar conhecimento sobre os requisitos identificados, avaliar o plano de alto nível do projecto e concertar com ele as suas acções.

Participação em fluxos de informação externos

- [Contingencial] PPE: No sentido de desenvolver as suas acções de influência externa.

O seu desempenho é crítico para ...

- A criação de condições (internas e externas) favoráveis ao desenvolvimento do projecto.
- Potenciar a exploração de novas oportunidades comerciais (associadas ao projecto) consideradas interessantes.

Deve evitar ...

- Desenvolver iniciativas relacionadas com o projecto sem o conhecimento do GPI.

No início do projecto deve ...

- Analisar o conjunto de requisitos (funcionais e não funcionais) identificados e respectiva documentação associada (entre a qual o modelo de casos de uso), comunicados pelo GPI.
 - Aprovar, mediante proposta do GPI, o plano de alto nível do projecto que deverá:
 - Identificar explicitamente os momentos de entrega de cada um dos artefactos (aplicacionais e não só) a disponibilizar pela organização durante o projecto;
 - Identificar explicitamente os compromissos da organização que se encontram dependentes de acções a realizar por entidades externas;
 - Servir de suporte para a monitorização (interna e externa) da evolução do projecto.
- Publicar, em local definido pela organização, a identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.

Durante cada iteração deve ...

- Tirando partido da sua influência junto de uma determinada entidade externa, intervir (de forma concertada com o GPI) no sentido de:
 - Minimizar os efeitos de eventuais problemas ocorridos no âmbito da implementação do projecto (por exemplo, o incumprimento de um prazo);
 - Desbloquear situações que dela dependam e que prejudiquem ou impeçam o normal desenrolar do projecto (por exemplo, ausência de tomada de decisões importantes);
 - Negociar concessões importantes para a organização (por exemplo, a dilatação de um prazo de execução);
 - Explorar eventuais oportunidades comerciais associadas ao projecto (por exemplo, adição de novas funcionalidades);
 - Negociar compensações devidas por incumprimento das suas responsabilidades (por exemplo, por atraso significativo na entrega de equipamento);
 - Assegurar a nomeação de um GPE com os poderes e condições necessárias para o bom desempenho da função.

No final do projecto deve ...

- Aquilatar o cumprimento do plano de alto nível para a última iteração.
- Proceder a uma avaliação global do projecto e do desempenho do GPI, encaminhando internamente os resultados de acordo com os processos instituídos.
- Comunicar ao CDI a sua avaliação do processo de desenvolvimento de software utilizado no projecto, sugerindo eventuais correcções ou evoluções.

No início de cada iteração deve ...

- Aquilatar o cumprimento do plano de alto nível para a iteração anterior.
- Rever, em conjunto com o GPI, eventuais alterações ao conjunto de requisitos (funcionais e não funcionais) identificados e respectiva documentação associada (entre a qual o modelo de casos de uso).
- Analisar, em conjunto com o GPI, eventuais alterações ao plano de alto nível solicitadas por entidades externas.

- Tirando partido da sua influência no interior da organização, intervir (de forma concertada com o GPI) no sentido de:

- Assegurar que são disponibilizados os recursos (humanos ou outros) necessários ao normal desenrolar do projecto;
- Assegurar a satisfação atempada de requisitos de elevada importância para uma entidade externa;
- Assegurar a execução atempada das passagens a certificação/produção.

No final de cada iteração deve ...

- Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis.

4.2.13. [PRI] Programador

As incumbências deste interveniente são genéricas por definição, já que variam de acordo com os requisitos estabelecidos para os componentes/sistemas que deverão ser desenvolvidos no âmbito do projecto. Contudo, em traços gerais pode dizer-se que o empenho e brio profissional com que se deve pautar a sua intervenção serão preponderantes para o cumprimento dos compromissos externos assumidos pela organização e para a obtenção dos níveis qualitativos desejados para os artefactos que ajudar a produzir. Nesse sentido, é extremamente importante a observação das boas práticas que se encontrem em vigor na organização.

Responsável por ...

- | | |
|---|---|
| <ul style="list-style-type: none"> ▪ Observar as boas práticas em vigor na organização no que concerne ao desenvolvimento de software, entre as quais poderão estar as seguintes: <ul style="list-style-type: none"> ○ Não utilização de valores <i>hard-coded</i>, optando antes pela sua inclusão na configuração (seja em que formato for) do componente/aplicação; ○ Implementação de testes unitários nos componentes por si desenvolvidos; ○ Actualizar frequentemente (períodos inferiores a uma semana) no sistema de controlo de versões o código fonte do(s) componente(s) em que se encontra envolvido, com a condição de o mesmo se encontrar compilável sem erros. ▪ Executar, com o máximo empenho e brio profissional, as tarefas que lhe forem atribuídas pelo CEI encarregue de coordenar o seu trabalho. ▪ Notificar o respectivo CEI sempre que pretender proceder a uma alteração de férias. | <ul style="list-style-type: none"> ▪ Observar as boas práticas em vigor na organização no que concerne à criação de artefactos documentais, entre as quais poderão estar as seguintes: <ul style="list-style-type: none"> ○ Utilização de identificadores de documentos unívocos e normalizados; ○ Utilização de um sistema de gestão documental; ○ Utilização de modelos pré-definidos para determinados tipos de documentos. ▪ Alertar, tão cedo quanto possível, o respectivo CEI para situações em que não lhe seja possível (pelo menos sem intervenção de terceiros) terminar uma tarefa dentro do prazo previsto. ▪ Comunicar semanalmente ao respectivo CEI o tempo de que ainda necessita para terminar as tarefas que tem a seu cargo. |
|---|---|

Cardinalidade

- Seja qual for o modelo adoptado, este papel tem um cariz obrigatório, podendo ser desempenhado em simultâneo por vários indivíduos num mesmo projecto e divididos por várias linhas de desenvolvimento que, apesar de possivelmente relacionadas entre si, possuem objectivos próprios e evoluem de forma independente.

Perfil

- Embora não obrigatório, é desejável que os seus titulares satisfaçam o seguinte perfil:
 - Formação de 2º ciclo de Bolonha (ou equivalente) em informática;
 - Conhecimento aprofundado das ferramentas de desenvolvimento adoptadas pela organização;
 - Conhecimento aprofundado da notação UML.

Participação em fluxos de informação internos

- [Preferencial] CEI: No sentido de tomar conhecimento sobre as tarefas que este lhe atribuir e, consequentemente, de lhe comunicar informação sobre a evolução da execução das mesmas.

Participação em fluxos de informação externos

- (não é expectável a sua participação em fluxos de informação externos)

O seu desempenho é crítico para ...

- Permitir uma possível intervenção correctiva tão precoce quanto possível em situações de potencial incumprimento de objectivos.
- Criar artefactos (aplicacionais ou não) com o nível de qualidade desejado pela organização.

Deve evitar ...

- Não avisar o respectivo CEI sempre que:
 - Considerar não possui os conhecimentos necessários para executar uma tarefa que lhe foi atribuída;
 - Considerar que o prazo que lhe foi dado para executar uma determinada tarefa não é suficiente;
 - Apesar de inicialmente ter considerado suficiente o prazo para executar uma determinada tarefa, verifique que isso não corresponde à verdade.

No início do projecto deve ...

- Verificar se o seu calendário de férias se encontra actualizado e, caso não esteja, comunicar esse facto ao respectivo CEI.
- Publicar, em local definido pela organização, a identificação de todos os intervenientes externos relacionados com o projecto que sejam do seu conhecimento, sua caracterização e contactos conhecidos.

No início de cada iteração deve ...

- Solicitar ao respectivo CEI a atribuição das tarefas para a iteração.

Durante cada iteração deve ...

- Observar as boas práticas em vigor na organização no que concerne ao desenvolvimento de software, entre as quais poderão estar as seguintes:
 - Não utilização de valores *hard-coded*, optando antes pela sua inclusão na configuração (seja em que formato for) do componente/aplicação;
 - Implementação de testes unitários nos componentes por si desenvolvidos;
 - Actualizar frequentemente (períodos inferiores a uma semana) no sistema de controlo de versões o código fonte do(s) componente(s) em que se encontra envolvido, com a condição de o mesmo se encontrar compilável sem erros.
- Executar, com o máximo empenho e brio profissional, as tarefas que lhe forem atribuídas pelo CEI encarregue de coordenar o seu trabalho.
- Notificar o respectivo CEI sempre que pretender proceder a uma alteração de férias.

- Observar as boas práticas em vigor na organização no que concerne à criação de artefactos documentais, entre as quais poderão estar as seguintes:
 - Utilização de identificadores de documentos unívocos e normalizados;
 - Utilização de um sistema de gestão documental;
 - Utilização de modelos pré-definidos para determinados tipos de documentos.
- Alertar, tão cedo quanto possível, o respectivo CEI para situações em que não lhe seja possível (pelo menos sem intervenção de terceiros) terminar uma tarefa dentro do prazo previsto.
- Comunicar semanalmente ao respectivo CEI o tempo de que ainda necessita para terminar as tarefas que tem a seu cargo.

No final do projecto deve ...

- Comunicar ao respectivo CEI a sua avaliação dos componentes reutilizados na sua linha de desenvolvimento, identificando eventuais correcções ou evoluções a realizar.
- Comunicar ao CDI a sua avaliação do processo de desenvolvimento de software utilizado no projecto, sugerindo eventuais correcções ou evoluções.

No final de cada iteração deve ...

- Assegurar que código fonte do(s) componente(s) em que se encontra envolvido se encontra actualizado no sistema de controlo de versões.
- Verificar se pretende efectuar alguma alteração de férias que incida sobre a iteração seguinte e, nesse caso, notificar o respectivo CEI.
- Assegurar que os dados existentes sobre a alocação do seu tempo de trabalho se encontram actualizados e disponíveis.

4.3. Conclusões

Pelos motivos anteriormente referidos, o desempenho dos participantes num processo de desenvolvimento de software levado a cabo numa PME é bastante condicionado pela existência de um reduzido leque de recursos humanos que, usualmente, acumulam o novo papel a que são chamados com responsabilidades em outros projectos em curso, para além de outras eventualmente resultantes da sua intervenção em projectos anteriores (por exemplo, caso seja responsável por actividades de manutenção de um serviço previamente desenvolvido e que já se encontre em produção). Por isso mesmo, e independentemente das competências e capacidades individuais de cada um dos envolvidos, estas circunstâncias criam um ambiente propenso ao erro, quer por acção (provavelmente pouco reflectida) ou omissão (seja esta por insuficiência de tempo para concretizar tudo quando dele se espera ou por não conseguir ter presente todas as tarefas que lhe foram atribuídas).

Assim, este capítulo poderá ser um contributo positivo para endereçar este último problema, ao auxiliar cada um dos intervenientes não só a saber claramente o que dele se espera (através da enumeração exaustiva das suas responsabilidades), como também a identificar o momento apropriado para o fazer (ao associar cada uma das suas incumbências a um momento-chave do projecto). Adicionalmente, procurou-se reduzir a margem de erro, nomeando (sempre que possível e aplicável) um determinado interveniente para verificar/aprovar o cumprimento de uma acção por parte de um terceiro. Dessa forma, caso o responsável pela tarefa não a execute (eventualmente por esquecimento) no momento apropriado, poderá ser lembrado por outro elemento de que o deveria fazer, ajudando a evitar alguns erros por omissão.

Contudo, a utilidade e eficácia deste contributo apenas pode ser efectivamente mensurada através da sua aplicação a contextos PME reais, que permitam retirar algumas ilações sobre a sua adequabilidade. Nesse sentido, e apesar de não ser objectivo desta dissertação avaliar a adequabilidade dos modelos propostos a situações reais, no capítulo que se segue o leitor poderá encontrar um relato dos resultados que o autor desta dissertação tem vindo a observar na sequência da sua aplicação aos contextos organizacionais em que desenvolve a sua actividade profissional. Para além disso, serão também identificadas algumas das direcções que poderão ser exploradas no sentido de dar sequência ao trabalho realizado no âmbito desta dissertação.

5. Conclusões

O capítulo que aqui se inicia tem como objectivo transmitir algumas conclusões que resultam da aplicação num contexto PME de um misto dos modelos propostos nesta dissertação, pelo que poderão ser de utilidade para organizações que ponderem a sua adopção. Na sequência dos problemas detectados e das vertentes dos modelos que não foi possível explorar, fruto da exiguidade temporal de uma dissertação deste género, são, ainda, identificadas possíveis formas para dar seguimento ao trabalho desenvolvido.

5.1. Experimentação e Resultados Obtidos

O *Rational Unified Process* é, de acordo com o que foi apresentado ao longo deste documento, um abrangente processo de desenvolvimento de software, que visa auxiliar as organizações adoptantes a utilizar eficientemente os recursos de que dispõem para assegurar a eficaz concretização dos objectivos (associados a esta área do conhecimento) que pretendem alcançar. Contudo, não obstante os seus inegáveis méritos, foram também expostas algumas lacunas da metodologia, como sejam a sua omissão sobre questões relacionadas com o desenvolvimento simultâneo de projectos (essenciais para gerir adequadamente eventuais recursos humanos partilhados pelos mesmos), com a melhoria contínua (essencial nos níveis CMM mais elevados) e com a preparação das actividades futuras de suporte e manutenção (essenciais para assegurar que a fase de exploração do projecto decorre sem sobressaltos).

Adicionalmente, ficou também claro que a complexidade inerente ao RUP dificulta a sua aplicação em contextos PME, aconselhando a condução de um esforço adaptativo. Porém, e apesar da sua natureza parametrizável, o processo de configuração da metodologia, para que melhor se adapte às especificidades das organizações destinatárias, é complexo e moroso, resultando numa barreira (de maior ou menor dimensão, consoante o desfasamento

verificado) que estas necessitam de ultrapassar. Nesse sentido, a recolha de literatura efectuada permitiu identificar não só casos bem sucedidos na adopção do RUP por organizações de reduzida dimensão, como também algumas experiências de configuração relevantes para esses contextos, nomeadamente no que concerne ao conjunto de artefactos a produzir. Contudo, a ausência de configurações do elenco processual do RUP apropriadas para uma organização PME justificou a realização desta dissertação, através da qual se demonstrou ser possível reduzir significativamente a dimensão do elenco processual envolvido na operacionalização da metodologia, observando um processo racional e devidamente fundamentado. Como resultado, foram produzidos dois modelos, denominados de *base* e *reduzido*, passíveis de aplicação em organizações de média e micro/pequena dimensão, respectivamente. Para além disso, procedeu-se ainda à caracterização detalhada (consubstanciada nas fichas individuais apresentadas no capítulo 4) de cada um dos papéis pertencentes aos modelos apresentados, de modo a auxiliar os seus titulares no cumprimento das competências que lhe são conferidas.

Por outro lado, e como já foi referido anteriormente, não era objectivo desta dissertação aplicar os modelos propostos de uma forma exaustiva e em contextos reais. No entanto, tendo o autor desta dissertação tido, no âmbito do seu contexto profissional (uma PME da área do desenvolvimento de software), oportunidade de aplicar o modelo reduzido¹⁵⁶ proposto, considerou-se como relevante a observação (ainda que subjectiva e não mensurada) das consequências aparentes da sua aplicação a um projecto de elevada complexidade e visibilidade para a organização, iniciado em final de 2006. Contudo, mais do que mero pano de fundo para esta experimentação, o contexto profissional do autor serviu também para ajudar a conceptualizar e refinar os elencos processuais propostos, no sentido de encontrar soluções para os problemas usualmente vivenciados nesse tipo de organizações.

O referido projecto, no âmbito do qual o autor desta dissertação exerceu a função de CDI, envolveu a existência de quatro linhas de desenvolvimento distintas (embora dependentes entre si) e causou um elevado impacto na organização, em virtude da participação de um total de vinte e quatro indivíduos¹⁵⁷ (correspondendo a 77% dos recursos humanos ao dispor da mesma), dispersos por duas localizações geográficas distintas. Aprofundando um pouco mais o contexto desta organização à data de início do projecto, era

¹⁵⁶ Apesar de complementado com a participação de um CSI e de um CCI.

¹⁵⁷ Subdivididos em três PPI, um CSI, um CII, um CDI, um GPI, quatro CEI, um CCI, onze PRI (quatro dos quais acumulavam o papel de CEI), um CQI e quatro AQI.

possível identificar várias disfunções que careciam de resolução: (1) apesar de antes dos modelos propostos nesta dissertação começarem a ser aplicados na organização já se proceder, em cada projecto, à nomeação de um *Gestor de Projecto* e de um ou mais *Programadores*, não existia uma definição clara das responsabilidades de cada um, verificando-se consideráveis diferenças de projecto para projecto e alguma frustração por parte dos envolvidos, por não saberem claramente o que deles era esperado; (2) não existia o papel equivalente ao CEI, pelo que o então denominado *Gestor de Projecto* acumulava a coordenação do(s) *Programador(es)* com o acompanhamento das entidades externas, o que por vezes resultava em perdas de eficiência, causadas pela falta de resposta em tempo útil do *Gestor de Projecto*; (3) nem sempre era assegurado o adequado e atempado envolvimento dos responsáveis pelos serviços de infra-estrutura e suporte, o que, por vezes, se traduzia na sua falta de resposta em momentos-chave ou na descoordenação de acções com os restantes intervenientes; (4) não era prática na organização a existência de uma equipa de qualidade, independente dos envolvidos na implementação do projecto, a cargo dos quais era entregue o esforço de testes.

Focando agora nas consequências da aplicação do modelo reduzido à organização adoptante, pode dizer-se que o balanço é positivo, já que: (1) permitiu a concretização atempada dos objectivos definidos para o projecto; (2) contribuiu para delimitar claramente as áreas de intervenção de cada interveniente e, assim, orientar as suas acções individuais; (3) foi possível gerir eficazmente o âmbito do projecto e analisar cuidadosamente o impacto das alterações de âmbito necessárias ao longo do seu curso.

Contudo, e como seria expectável, a experiência não foi inteiramente satisfatória, não só em virtude de incorrecções e/ou omissões do modelo proposto, como também devido a uma errada e/ou deficiente aplicação das recomendações que encerra. Em primeiro lugar, e apesar de se encontrar em dedicação exclusiva ao projecto, foi notório que o GPI não conseguiu assegurar o cumprimento de todas as suas responsabilidades, o que poderá indiciar uma excessiva acumulação de papéis¹⁵⁸ pelo GPI no modelo reduzido. Um dos reflexos dessa incapacidade foi a inexistência das suas reuniões periódicas com o(s) CEI, CII e CSI, o que resultou: (1) em alguma descoordenação entre as actividades das várias linhas de desenvolvimento; (2) no frequente desconhecimento dos CEI sobre as respostas às questões endereçadas às entidades externas por intermédio do GPI; (3) na falta de conhecimento do

¹⁵⁸ *Project Manager, Business-Process Analyst, Change Control Manager, Deployment Manager, Requirements Specifier, Review Coordinator, Test Analyst, System Analyst, Business Designer e Use Case Specifier*, num total de dez papéis.

GPI sobre alguns dos obstáculos enfrentados pelos CEI, que por vezes se traduziram em atrasos que poderiam e deveriam ter sido evitados/minimizados. Por outro lado, é possível dizer que o desempenho do CDI não assumiu a eficácia desejada, essencialmente devido à acumulação por parte deste de outras responsabilidades no interior da organização que, pela sua natureza e envolvimento, o impediram de acompanhar de forma mais próxima os CEI e, dessa forma, compensar mais eficazmente as falhas de coordenação do GPI. Um outro interveniente que também não prestou o contributo que dele se esperava foi o CQI, por motivos semelhantes aos referidos para o CDI, expondo claramente as dificuldades existentes num contexto PME em assegurar as condições necessárias a um óptimo exercício das funções individuais. Apesar de não ter tido oportunidade de definir totalmente o plano interno de auditoria de qualidade, o CQI optou por não o fazer da forma recomendada pelo modelo, não envolvendo os CEI na sua concepção e, conseqüentemente, incorrendo em alguns erros que poderiam ser facilmente evitados com um maior conhecimento de causa sobre as particularidades de cada linha de desenvolvimento. Adicionalmente, não foi possível executar na sua plenitude o plano interno de auditoria de qualidade existente, o que se traduziu em algumas ocorrências indesejáveis: (1) ausência de detecção de alguns erros aplicativos; (2) ausência de verificação periódica sobre estado de cumprimento de requisitos (resultando num esforço adicional e esporádico por parte do GPI); (3) incumprimento temporal na satisfação de alguns requisitos, não por falta de capacidade para o fazer mas sim por esquecimento (dado que por vezes se encontram formalizados no interior de documentos de grande dimensão). Finalmente, e ao contrário do que o modelo advoga, deve também ser registado que nem todas as acções dos PPI foram devidamente concertadas com o GPI, potenciando equívocos e falhas de comunicação, para além de não contribuir para prestigiar o GPI perante as entidades externas.

5.2. Trabalho Futuro

Dada a inerente exiguidade temporal associada a uma dissertação deste género, o seu desenvolvimento não permite a exploração de várias vertentes que, apesar de complementares ao tema a tratar, tornariam o âmbito demasiadamente abrangente para possibilitar uma obtenção atempada de resultados. Assim, a continuidade a dar a esta dissertação poderá trilhar diversos caminhos: por um lado, prosseguindo a linha de adaptação do RUP a contextos PME, poderá proceder-se a um esforço de simplificação do lote de artefactos previstos pela metodologia e a uma redefinição dos responsáveis pela sua produção à luz do novo elenco processual proposto. Para além disso, e dado que os modelos propostos apenas

são válidos para a fase de execução de um projecto de desenvolvimento de software, seria interessante propor modelos complementares que endereçassem o período a montante (fase de angariação, durante a qual é necessária a interacção de um conjunto de elementos no sentido de concretizar a oportunidade de realizar o projecto) e a jusante (fase de manutenção, durante a qual pode ser necessária a interacção de um conjunto de elementos no sentido de assegurar a disponibilidade dos sistemas, corrigir problemas aplicativos e responder atempadamente às solicitações dos utilizadores finais). Dessa forma, seria possível identificar os pontos (e intervenientes) de contacto entre as referidas fases e, assim, contribuir para assegurar uma transição suave e eficaz entre as mesmas. Ainda no âmbito do novo elenco processual proposto, seria interessante analisar a relevância da criação de grupos de intervenientes¹⁵⁹ aos quais fosse pertinente associar atribuições que, de uma forma isolada, não fariam sentido apenas no âmbito das competências individuais de cada um dos elementos.

Por outro lado, e de forma a facilitar ainda mais o desempenho dos vários papéis propostos, poderiam ser desenvolvidos diagramas de actividade que permitissem ilustrar, mesmo que apenas a alto nível, a sequenciação mais apropriada para as tarefas pelas quais se encontra responsável. Finalmente, e para se poder aquilatar verdadeiramente a eficácia dos modelos propostos nesta dissertação, seria de todo desejável a realização de um estudo exaustivo sobre a sua aplicação em contextos PME reais, após o qual seria certamente possível retirar um conjunto de conclusões essenciais para a melhoria qualitativa dos mesmos.

¹⁵⁹ Por exemplo, um *steering committee* (ou comité de direcção), um *technical committee* (ou comité técnico), etc.

Bibliografia

[Alfresco] *Alfresco*. <http://www.alfresco.com/>.

[Assembla] *Assembla*. <http://www.assembla.com/>.

[Beira, et al., 2006] Beira, E., Sousa, H., Borges, P., Kaldeich, C. (2006). *WP57 - Mapa TIC de Portugal. Análise por distritos*.

[Bergström, Råberg, 2003] Bergström, S., Råberg, L. (2003). *Adopting the Rational Unified Process: Success with the RUP*. Addison-Wesley Professional.

[Booch, et al., 2007] Booch, G., Maksimchuk, R. A., Engel, M. W., Young, B. J., Conallen, J., Houston, K. A. (2007). *Object-Oriented Analysis and Design with Applications*. Addison-Wesley Professional.

[Booch, Rumbaugh, 1999] Booch, G., Rumbaugh, J. (1999). *The Unified Modeling Language User Guide*. Addison-Wesley.

[CVS] *CVS*. http://en.wikipedia.org/wiki/Concurrent_Versions_System.

[Dia] *Dia*. <http://www.gnome.org/projects/dia/>.

[Duarte, et al., 2006] Duarte, F. J., Fernandes, J. M., Machado, R. J. (2006). *Business Modeling in Process-Oriented Organizations for RUP-based Software Development*. Reference Modeling for Business Systems Analysis, Idea Group Publishing, section II: Reference Modeling Models, chap. V, pp. 98-117.

[Eclipse] *Eclipse*. <http://www.eclipse.org/>.

[Eclipse Process Framework] *Eclipse Process Framework*. <http://www.eclipse.org/epf/>.

[Fernandes, Duarte, 2005] Fernandes, J. M., Duarte, F. (2005). *A reference framework for process-oriented software development organizations*. Software and Systems Modeling (SoSyM), Springer-Verlag, vol. 4, nr.1, 94-105.

[Fernandes, Duarte, 2003] Fernandes, J. M., Duarte, F. (2003). *A reference framework for process-oriented software development organizations*.

[Henderson-Sellers, et al., 2001] Henderson-Sellers, B., Collins, G., Dué, R., Graham, I. (2001). *A qualitative comparison of two processes for object-oriented software development*. Information and Software Technology, Elsevier, vol. 43, 705-724.

[Hesse, 2003] Hesse, W. (2003). *Dinosaur meets Archaeopteryx? or: Is there an alternative for Rational's Unified Process?* Software and Systems Modeling (SoSyM), Springer-Verlag, vol. 2, 240-247.

[Hesse, 1997] Hesse, W. (1997). *Improving the Software Process guided by the EOS model*. Proc. SPI '97 European Conference on Software Process Improvement.

[Hirsch, 2002] Hirsch, M. (2002). *Making RUP Agile*. ACM - The Guide, OOPSLA 2002 Practitioner Report.

[IAPMEI, 2007] IAPMEI (2007). *Perguntas Frequentes sobre as PME*. <http://www.iapmei.pt/iapmei-faq-02.php?tema=7>.

[IEEE, 2004] IEEE (2004). *SWEBOK - Guide of the Software Engineering Body of Knowledge*.

[Ivar Jacobson, et al., 1992] Ivar Jacobson, Christerson, M., Jonsson, P., Övergaard, G. (1992). *Object-Oriented Software Engineering: A Use-Case Driven Approach*. Addison-Wesley.

[Jacobson, et al., 1999] Jacobson, I., Booch, G., Rumbaugh, J. (1999). *The Unified Software Development Process*. Addison-Wesley.

[JIRA] JIRA. <http://www.atlassian.com/software/jira/>.

[Kroll, Kruchten, 2003] Kroll, P., Kruchten, P. (2003). *The Rational Unified Process made easy: a practitioner's guide to the RUP*. Addison Wesley.

[Kroll, MacIsaac, 2006] Kroll, P., MacIsaac, B. (2006). *Agility and Discipline Made Easy: Practices from OpenUP and RUP*. Addison-Wesley Professional.

[Kruchten, 2004] Kruchten, P. (2004). *The Rational Unified Process: an introduction*. Addison Wesley.

[Larman, 2003] Larman, C. (2003). *Agile and iterative development*. Addison-Wesley.

[Magic Draw] Magic Draw. <http://www.magicdraw.com/>.

[Microsoft Project] Microsoft Project. <http://office.microsoft.com/pt-br/project/default.aspx>.

[Microsoft Word] Microsoft Word. <http://office.microsoft.com/pt-br/word/default.aspx>.

- [Mr Project] *Mr Project*. <http://mrproject.codefactory.se/>.
- [Naur, Randell, 1969] Naur, P., Randell, B. (1969). *Software Engineering*. Garmisch, Germany.
- [OMG, 2001] OMG (2001). *Software Process Engineering Metamodel (SPEM)*. Object Management Group.
- [Open Office] *Open Office*. <http://www.openoffice.org/>.
- [Paulk, et al., 1995] Paulk, M. C., Weber, C. V., Curtis, B., Chrissis, M. B., (eds.) (1995). *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley.
- [Pollice, et al., 2003] Pollice, G., Augustine, L., Lowe, C., Madhur, J. (2003). *Software Development for Small Teams: A RUP-Centric Approach*. Addison-Wesley Professional.
- [Rational, 2006] Rational (2006). *RUP Tutorial*. IBM Rational Method Composer v7.1.0.00.
- [Rational, 2001] Rational (2001). *RUP Tutorial*.
- [Rational RequisitePro] *Rational RequisitePro*. <http://www-306.ibm.com/software/awdtools/reqpro/>.
- [Schewe, 2001] Schewe, K. D. (2001). *UML: A Modern Dinosaur? – A Critical Analysis of the Unified Modelling Language*. Information Modelling and Knowledge Bases XII. Proc. 10th European-Japanese Conf., vol. 67.
- [Subversion] *Subversion*. <http://subversion.tigris.org/>.
- [Visual Paradigm] *Visual Paradigm*. <http://www.visual-paradigm.com/>.

Anexo I Elenco Processual do RUP

Papel	Descrição
Analyst	
<i>Business Reviewer</i> (ou <i>Business-Model Reviewer</i>)	Responsável por rever os artefactos criados no âmbito da disciplina de <i>Business Modeling</i> , nomeadamente os relacionados com os modelos de casos de uso de negócio e de objectos de negócio.
<i>Business-Process Analyst</i>	Lidera e coordena a definição da arquitectura de negócio e a modelação dos casos de uso de negócio, demarcando e delimitando a organização a modelar (por exemplo, estabelecendo quais os actores e casos de uso de negócio existem e como interagem entre si).
<i>Business Designer</i>	Detalha a especificação de uma parte da organização e especifica o fluxo dos casos de uso de negócio em termos de <i>workers</i> e entidades, para além de definir as responsabilidades, operações, atributos e relações destes.
<i>Requirements Reviewer</i>	Planeia e conduz a revisão formal do modelo de casos de uso.
<i>Requirements Specifier</i>	Especifica as funcionalidades do sistema ao descrever um ou mais aspectos dos requisitos, documentados das mais variadas formas: casos de uso, documento de especificações, etc. Adicionalmente, é responsável por assegurar a manutenção ao longo do tempo da congruência entre os vários documentos que descrevem os requisitos, de acordo com a evolução destes. Também pode ser responsável pela manutenção da integridade de um determinado pacote de casos de uso, sendo recomendável nesses casos que também o seja sobre os casos de uso e actores que o compõem.
<i>System Analyst</i>	Lidera e coordena a enumeração dos requisitos e a modelação dos casos de uso, delimitando o sistema e sublinhando a sua funcionalidade (por exemplo, estabelecendo quais os actores e casos de uso existem e como interagem entre si). Assim, é responsável por todo o conjunto de requisitos (funcionais e não funcionais) que são modelados como casos de uso, incluindo todos os requisitos específicos a cada caso de uso. Para além disso, é responsável por garantir que o modelo de casos de uso está completo e consistente. Para a consistência, o <i>System Analyst</i> pode utilizar um glossário para chegar a um consenso sobre os termos comuns, noções e conceitos quando os requisitos são capturados. Contudo, apesar de ser responsável pelo modelo de casos de uso e actores lá incluídos, não é responsável por cada caso de uso individual, atribuição do <i>Use Case Specifier</i> . Adicionalmente, o <i>System Analyst</i> é também o líder e coordenador da captura de requisitos. Apesar de apenas existir um <i>System Analyst</i> para cada sistema, na prática, este papel é suportado por uma equipa (em <i>workshops</i> ou eventos semelhantes) que inclui várias outras pessoas, que também trabalham como analistas.
<i>Use Case Specifier</i>	Usualmente, as actividades de captura de requisitos não podem ser concluídas por apenas um indivíduo. Em vez disso, estes intervenientes assessoram o <i>System Analyst</i> , responsabilizando-se pela descrição detalhada de um ou mais casos de uso. Nesse sentido, estes intervenientes necessitam de trabalhar de perto com os utilizadores finais dos seus casos de uso.
<i>User-Interface Designer</i>	Lidera e coordena a prototipagem e concepção das interfaces de utilizador, através da: <ul style="list-style-type: none"> ▪ captura de requisitos para a interface de utilizador, nomeadamente no que concerne à sua usabilidade; ▪ construção de protótipos das interfaces de utilizador; ▪ envolvendo outros <i>stakeholders</i> da interface de utilizador nas revisões de usabilidade e nas sessões de teste; ▪ revisão da implementação final das interfaces de utilizador, reportando as conclusões retiradas.
Developer	
<i>Architecture Reviewer</i>	Responsável por rever os artefactos-chave criados no âmbito da disciplina de <i>Analysis and Design</i> , nomeadamente os relacionados com a arquitectura de software inerente ao projecto.
<i>Capsule Designer</i>	É responsável pela concepção das cápsulas, assegurando que o sistema é capaz de responder atempadamente a eventos e de acordo com os requisitos de concorrência.
<i>Code Reviewer</i>	Assegura a qualidade do código fonte, sua conformidade com as respectivas políticas definidas para o projecto e planeia a realização de revisões ao mesmo, para além de se responsabilizar pelo reporte das conclusões resultantes desse processo.

Papel	Descrição
<i>Component Engineer</i>	<p>Define e mantém:</p> <ul style="list-style-type: none"> ▪ as responsabilidades, atributos, relações e requisitos especiais de uma ou mais classes de análise, assegurando que cada uma delas cumpre os requisitos impostos pelas concretizações de casos de uso em que participa; ▪ as operações, métodos, atributos, relações e requisitos de uma ou mais classes de concepção, assegurando que cada uma delas cumpre os requisitos impostos pelas concretizações dos casos de uso em que participa; ▪ o código fonte de um ou vários componentes (do tipo ficheiro), assegurando que cada componente implementa a funcionalidade correcta (ou seja, a que se encontra especificada na classe de concepção). <p>Compete-lhe também:</p> <ul style="list-style-type: none"> ▪ manter a integridade de um ou mais pacotes de análise, garantindo que os seus conteúdos (por exemplo, classes e suas relações) estão correctos e que as suas dependências em relação a outros pacotes de análise são correctas e mínimas; ▪ a implementação de componentes de teste que automatizem alguns dos procedimentos de teste (já que nem todos podem ser automatizados), dado que a criação destes componentes pode requerer substanciais competências de programação (de que o <i>Component Engineer</i> também necessitará durante o fluxo de implementação). <p>Usualmente:</p> <ul style="list-style-type: none"> ▪ é apropriado permitir que o <i>Component Engineer</i> que é responsável por uma pacote de análise o seja também pelas classes de análise que contém. Para além disso, quando existe um mapeamento directo entre um pacote de análise e os correspondentes sub-sistemas, o mesmo <i>Component Engineer</i> deverá ser também responsável por esses sub-sistemas, de modo a usar o conhecimento adquirido durante a análise efectuada ao desenhar e implementar o pacote de análise. Caso não exista esse mapeamento directo, outros <i>Component Engineers</i> poderão ser envolvidos na concepção e implementação no pacote de análise; ▪ também pode manter a integridade de um ou mais sub-sistemas, o que inclui assegurar que os seus conteúdos (por exemplo, classes e suas relações) estão correctas, que as suas dependências em relação a outros sub-sistemas e/ou interfaces estão correctas e minimizadas, e que concretizam correctamente as interfaces que proporcionam. É frequente permitir que o <i>Component Engineer</i> responsável por um sistema também o seja pelos elementos do modelo que contém. Para além disso, para atingir um desenvolvimento suave, é natural que os artefactos do modelo de concepção (por exemplo, as classes e sub-sistemas) sejam implementados durante do fluxo de desenvolvimento pelo mesmo <i>Component Engineer</i>; ▪ também mantém a integridade de um ou mais sub-sistemas de implementação. Dado que entre estes e os sub-sistemas concebidos existe uma correspondência um-para-um, a maior parte das alterações a estes sub-sistemas são geridas durante a concepção. Contudo, o <i>Component Engineer</i> necessita de garantir que os conteúdos (componentes, etc.) dos sub-sistemas de implementação estão correctos, que as suas dependências em relação a outros sub-sistemas e/ou interfaces estão correctas e que implementam correctamente as interfaces que proporcionam. É frequente que o <i>Component Engineer</i> responsável por um sub-sistema também o seja pelos elementos (componentes, etc.) contidos no seu modelo. Para além disso, para atingir um desenvolvimento suave, é natural que um <i>Component Engineer</i> seja responsável por um sub-sistema e respectivo conteúdo ao longo dos fluxos de concepção e implementação. Nesses casos, ele trataria de desenhar e implementar as classes que se encontrassem à sua responsabilidade.
<i>Database Designer</i>	Define as tabelas, índices, <i>views</i> , <i>constraints</i> , <i>triggers</i> , <i>stored procedures</i> e outros parâmetros específicos ao motor de base de dados relacionados com o armazenamento, recuperação e remoção de objectos persistentes.
<i>Designer</i>	É responsável pela concepção de uma parte do sistema, dentro das limitações impostas pelos requisitos, arquitectura e processo de desenvolvimento do projecto. Para além disso, identifica e define as responsabilidades, operações, atributos e relações de uma ou mais classes e determina como estas devem ser ajustadas ao ambiente de implementação. Adicionalmente, pode ser responsável por um ou mais pacotes ou sub-sistemas, incluindo quaisquer classes que pertençam aos referidos pacotes ou sub-sistemas. Assegura que a concepção do(s) sistema(s) é consistente com a arquitectura de software e que está suficientemente detalhado para que a implementação possa prosseguir.
<i>Design Reviewer</i>	Responsável por rever os artefactos-chave criados no âmbito da disciplina de <i>Analysis and Design</i> , nomeadamente os relacionados com o modelo de concepção.
<i>Implementer</i>	É responsável pelo desenvolvimento e teste de componentes, de acordo com as normas adoptadas no projecto para a integração em sub-sistemas de maior dimensão. Quando componentes de teste, como <i>drivers</i> ou <i>stubs</i> , têm de ser criados para suportar os testes, é também responsável pelo seu desenvolvimento e teste (e dos sub-sistemas correspondentes).
<i>Integrator (ou System Integrator)</i>	É responsável por combinar os componentes entregues pelos <i>Implementers</i> no sentido de produzir uma <i>build</i> . Tal inclui o planeamento da sequência de versões requeridas em cada iteração e a integração de cada versão assim que as suas partes estejam implementadas. Este planeamento resulta no Plano de Integração de Versões (<i>Integration Build Plan</i>).

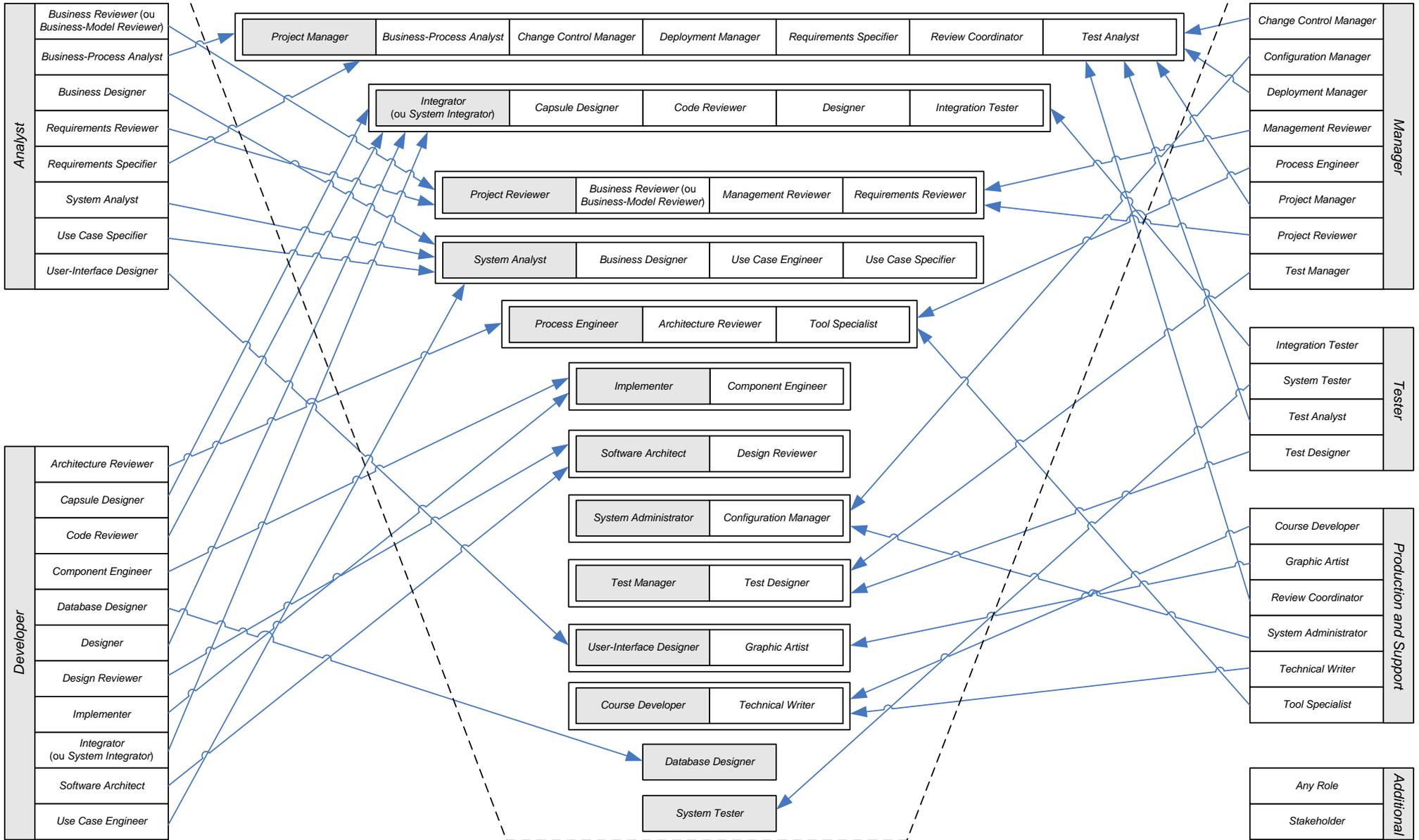
Papel	Descrição
<p><i>Software Architect</i></p>	<p>É responsável:</p> <ul style="list-style-type: none"> ▪ por estabelecer a estrutura geral de cada visão arquitectural: a decomposição da visão, o agrupamento de elementos e as interfaces entre os maiores agrupamentos. Logo, comparativamente com os restantes papéis, a visão do arquitecto deve ser mais abrangente do que profunda; ▪ por participar no fluxo de requisitos para descrever a vista arquitectural do modelo de casos de uso; ▪ por assegurar, durante o fluxo de análise, a integridade do modelo de análise, assegurando que este está correcto, consistente e legível. Em sistema de grande dimensão e complexidade estas responsabilidades podem, após algumas iterações, requerer uma maior manutenção e o trabalho envolvido poderá tornar-se rotineiro. Nesses casos, o <i>Architect</i> poderá delegar estas funções noutro interveniente, possivelmente num <i>Component Engineer</i> de alto nível. Contudo, o <i>Architect</i> continuará responsável pelo que for arquitecturalmente significativo – a descrição arquitectural – enquanto o outro interveniente será responsável pelo pacote de topo do modelo de análise, que necessita de cumprir a descrição arquitectural; ▪ pela arquitectura do modelo de análise, ou seja, pela existência das suas partes arquitecturalmente significativas, de acordo com a visão arquitectural do modelo (recorde-se que esta visão é parte da descrição arquitectural do sistema); ▪ por assegurar, durante do fluxo de concepção, a integridade do modelo de concepção e de instalação, assegurando que os modelos estão correctos, consistentes e legíveis. Quanto ao modelo de análise, para sistemas de grande dimensão e complexidade, pode ser envolvido um outro interveniente para assumir a responsabilidade sobre o sistema de topo do modelo de concepção (ou seja, pelo processo de concepção). Os modelos estão correctos quando concretizam (apenas) a funcionalidade descrita no modelo de casos de uso, os requisitos adicionais e o modelo de análise; ▪ pela arquitectura dos modelos de concepção e instalação, ou seja, pela existência das suas partes arquitecturalmente relevantes, como representado na vista arquitectural desses modelos, que é parte da descrição arquitectural do sistema; ▪ por assegurar, durante a fase de implementação, a integridade do modelo de implementação e por assegurar que o modelo está correcto, consistente e legível. Quanto à análise e concepção, para sistemas de grande dimensão e complexidade, pode ser envolvido um outro interveniente para assumir a responsabilidade sobre o sistema de topo do modelo de implementação (ou seja, pelo sistema de implementação). O modelo é correcto quando implementa (apenas) a funcionalidade descrita no modelo de concepção e nos (relevantes) requisitos adicionais; ▪ pela arquitectura do modelo de implementação, ou seja, pela existência das suas partes arquitecturalmente relevantes, como representado na vista arquitectural desses modelos, que é parte da descrição arquitectural do sistema. Finalmente, um resultado importante da implementação é o mapeamento dos componentes executáveis para os nodos, processo pelo qual o arquitecto é responsável. <p>Não é responsável pela:</p> <ul style="list-style-type: none"> ▪ pelo desenvolvimento contínuo e pela manutenção dos vários artefactos contidos nos modelos de análise e de concepção. Em vez disso, estas responsabilidades cabem ao <i>Use-Case Engineer</i>; ▪ pelo desenvolvimento contínuo e pela manutenção dos vários artefactos contidos no modelo de implementação. Em vez disso, estas responsabilidades cabem ao <i>Component Engineer</i>.
<p><i>Use-Case Engineer</i></p>	<p>Responsável pela:</p> <ul style="list-style-type: none"> ▪ integridade de uma ou mais concretizações de casos de uso, assegurando que cumprem integralmente os respectivos requisitos. Uma concretização de um caso de uso deve descrever correctamente (e apenas) o comportamento correspondente ao caso de uso respectivo. Nesse sentido, é necessário assegurar que todas as descrições textuais e diagramas que descrevem a concretização do caso de uso são legíveis e servem o seu propósito. Note-se que o <i>Use-Case Engineer</i> não é responsável pelas classes e relações de análise empregues na concretização do caso de uso, funções a desempenhar pelo <i>Component Engineer</i>. Adicionalmente, o <i>Use-Case Engineer</i> é também responsável pela concepção das concretizações do caso de uso já que, ao tratar da análise e concepção, assegura uma transição natural; ▪ integridade da concepção de uma ou mais concretizações de casos de uso, assegurando que cumprem integralmente os respectivos requisitos. A concretização de um caso de uso deve descrever correctamente (e apenas) o comportamento correspondente ao caso de uso respectivo (análise no modelo de análise e comportamento do caso de uso respectivo no modelo de caso de uso). Isto inclui a criação de todos os textos e diagramas que descrevam a concretização do caso de uso de forma legível e adequada ao objectivo. Note-se que o <i>Use-Case Engineer</i> não é responsável pela concepção das classes, sub-sistemas, interfaces e relações empregues na concretização do caso de uso, funções a desempenhar pelo <i>Component Engineer</i>.
Manager	
<p><i>Change Control Manager</i></p>	<p>Supervisiona o processo de controlo de alterações. Usualmente, este papel é desempenhado por um Comité de Controlo da Mudança, composto por representantes de todas as partes interessadas, incluindo clientes, <i>developers</i> e utilizadores, etc. Em projectos de pequena dimensão, uma única pessoa, como o <i>Project Manager</i> ou o <i>Software Architect</i>, pode desempenhar este papel.</p>

Papel	Descrição
<i>Configuration Manager</i>	É responsável por disponibilizar a infra-estrutura de gestão de configurações para a equipa de desenvolvimento, que suporta a sua actividade de modo a que os <i>developers</i> e integradores possuam os espaços de trabalho apropriados às suas acções de implementação, teste e disponibilização dos artefactos que produzem. Assegura que o ambiente de gestão de configurações facilita as actividades de revisão, alteração e gestão de defeitos do produto. Adicionalmente, é também responsável por escrever o plano de gestão de configurações e por reportar estatísticas sobre o progresso de pedidos de alteração.
<i>Deployment Manager</i>	É responsável por planear a transição do produto para a comunidade de utilizadores, assegurando que esse plano é executado de forma apropriada, documentando adequadamente o processo, gerindo os problemas suscitados e monitorizando o seu progresso.
<i>Management Reviewer</i>	É responsável pela avaliação do planeamento do projecto e dos principais artefactos produzidos em determinados momentos-chave do ciclo de vida do projecto.
<i>Process Engineer</i>	É responsável pelo processo de desenvolvimento de software em si, incluindo a sua configuração antes de se iniciar e a sua melhoria contínua ao longo do esforço de desenvolvimento.
<i>Project Manager</i>	Aloca recursos, define prioridades, coordena interações com clientes e utilizadores e esforça-se por manter a equipa de projecto focada nos objectivos correctos. Adicionalmente, estabelece um conjunto de práticas no sentido de assegurar a integridade e qualidade dos artefactos gerados pelo projecto.
<i>Project Reviewer</i>	É responsável por analisar os artefactos de planeamento do projecto e de avaliação do desenrolar do mesmo em determinados pontos-chaves do seu ciclo de vida. Estes são eventos de revisão significativos porque marcam pontos nos quais o projecto pode ser cancelado se o planeamento for inadequado ou se o progresso for inaceitavelmente mau.
<i>Test Manager</i>	É o responsável geral pelo sucesso do esforço de testes, promovendo a qualidade e a realização de testes. Para isso, necessita de desenvolver actividades de planeamento e gestão de recursos e de resolver os problemas que possam impedir o esforço de testes.
Tester	
<i>Integration Tester</i>	São responsáveis por realizar os testes de integração necessários para cada versão produzida no fluxo de implementação. Estes testes são executados de modo a verificar que os componentes integrados numa versão funcionam correctamente em conjunto. Por isso, os testes de integração são usualmente derivados de casos de teste que especificam como testar as concretizações dos casos de uso. Os resultados dos testes de integração são defeitos emanados pelo <i>Integration Tester</i> . Note-se que um <i>Integration Tester</i> testa o resultado (ou seja, uma versão) criada por um <i>System Integrator</i> durante o fluxo de implementação. Consequentemente, em alguns projectos estes papéis são acumulados por um ou mais indivíduos, de modo minimizar a necessidade de replicar conhecimento comum em colaboradores distintos.
<i>System Tester</i>	É responsável por realizar os testes de sistema necessários para uma versão de um resultado (executável) de uma iteração completa, avaliando a sua execução e recuperando de erros que possam ter lugar durante os mesmos. Estes testes são principalmente executados para verificar as interações entre os actores e o sistema. Por isso, os testes de sistema são usualmente derivados dos casos de teste que especificam como testar os casos de uso, mas outros tipos de testes também são aplicáveis ao sistema como um todo. Os resultados dos testes de sistema são defeitos emanados pelo <i>System Tester</i> . Dada a natureza dos testes de sistema, os indivíduos que actuam como <i>System Testers</i> não necessitam de saber muito sobre o funcionamento interno do sistema.
<i>Test Analyst</i>	É responsável por identificar e definir os testes necessários, monitorizar o progresso e resultados dos testes detalhados em cada ciclo de testes e avaliar a qualidade global experienciada em resultado das actividades de teste. Tipicamente, tem ainda a responsabilidade adicional de representar de forma adequada as necessidades dos <i>stakeholders</i> que não têm representação directa ou regular no projecto.
<i>Test Designer</i>	É o interveniente com maior envolvimento no esforço de testes, sendo responsável pelo seu planeamento, concepção, implementação e avaliação, incluindo a: <ul style="list-style-type: none"> ▪ geração do plano e modelo de testes, definindo os objectivos de cada teste e a sua calendarização; ▪ gestão da integridade do modelo de testes, assegurando que este cumpre os seus objectivos; ▪ selecção e descrição dos casos de teste e respectivos procedimentos de teste necessários; ▪ implementação de procedimentos de teste; ▪ avaliação da abrangência dos testes (incluindo os de integração e de sistema), dos seus resultados e da sua eficácia; ▪ geração de um sumário de avaliação do esforço de testes. <p>Note-se que o <i>Test Designer</i> não executa os testes (quem o faz são os <i>Integration Testers</i> e os <i>System Testers</i>), concentrando-se em vez disso na sua preparação e avaliação.</p>
Production and Support	
<i>Course Developer</i>	Desenvolve material de formação (slides, exemplos, tutoriais, etc.) para ensinar os utilizadores a compreender e usar o produto.
<i>Graphic Artist</i>	Desenvolve o trabalho gráfico necessário ao desenvolvimento do produto, relacionado entre outras coisas com o seu pacote de distribuição e respectiva documentação.

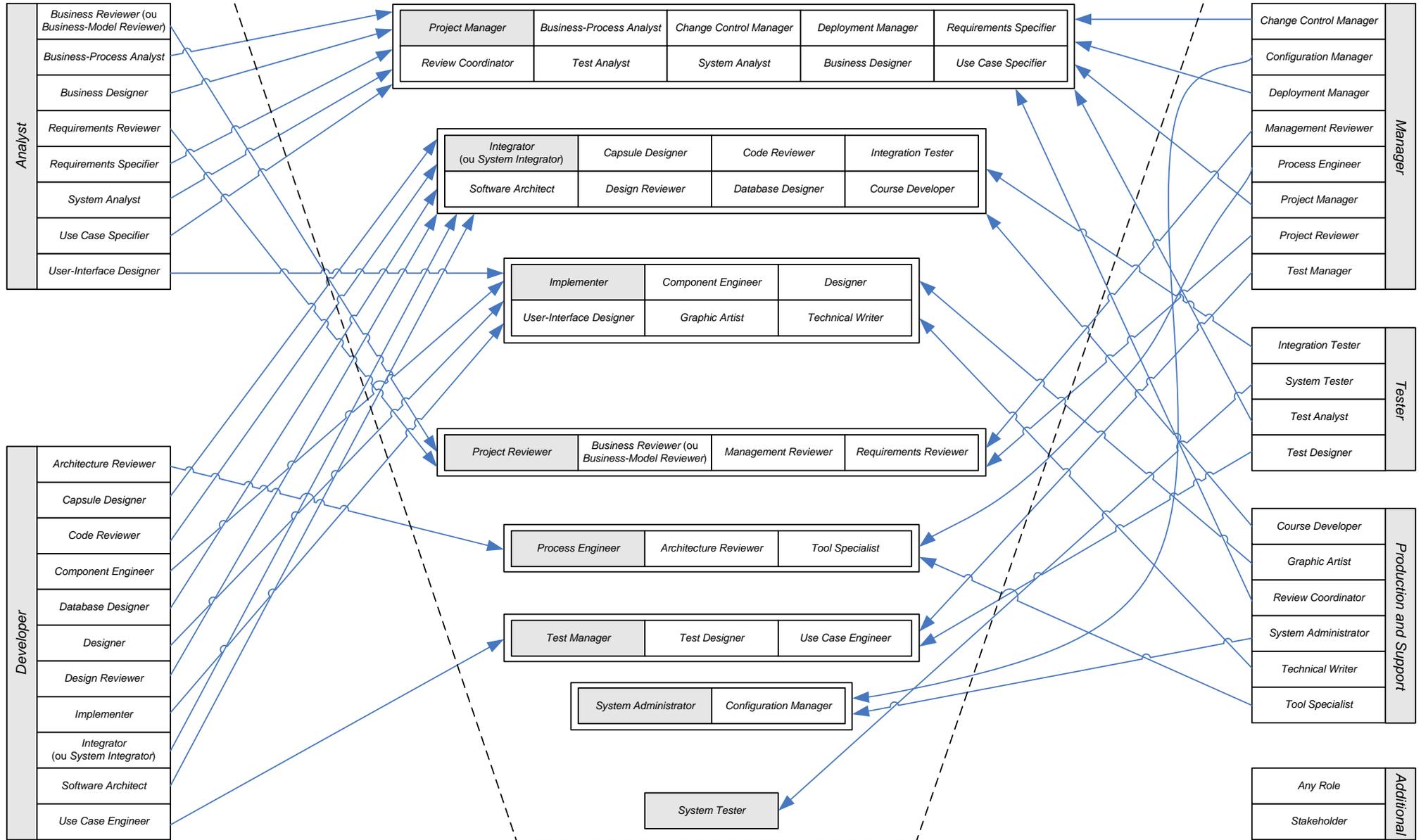
Papel	Descrição
<i>Review Coordinator</i>	É responsável por facilitar as revisões e inspecções formais, assegurando que elas ocorrem quando necessário e que são conduzidas de forma satisfatória e de acordo com os processos instituídos.
<i>System Administrator</i>	Mantém o ambiente de desenvolvimento (quer hardware quer software) e é responsável pela administração do sistema, pela realização de <i>backups</i> regulares, etc.
<i>Technical Writer</i>	Produz material de suporte a utilizadores finais, como manuais de utilizador, textos de ajuda, notas da distribuição, etc.
<i>Tool Specialist</i>	É responsável pelas ferramentas de suporte ao projecto, incluindo a selecção e aquisição das mesmas, após identificar as necessidades existentes. Adicionalmente, prepara e configura as ferramentas e verifica que estas se encontram funcionais.
Additional	
<i>Any Role</i>	Qualquer outro papel pode, com os necessários privilégios de acesso, efectuar o <i>check-in</i> e <i>check-out</i> dos artefactos do projecto para proceder à sua manutenção no sistema de controlo de configurações. Adicionalmente, também pode submeter pedidos de alteração e actualizar os pedidos de alteração da sua responsabilidade.
<i>Stakeholder</i>	Qualquer elemento que seja materialmente afectado pelo resultado do projecto. Este é outro papel genérico que cobre, dependendo do contexto do projecto, cliente, utilizadores finais, adjudicante, accionista, etc.

Anexo II Diagramas de Apoio ao Processo

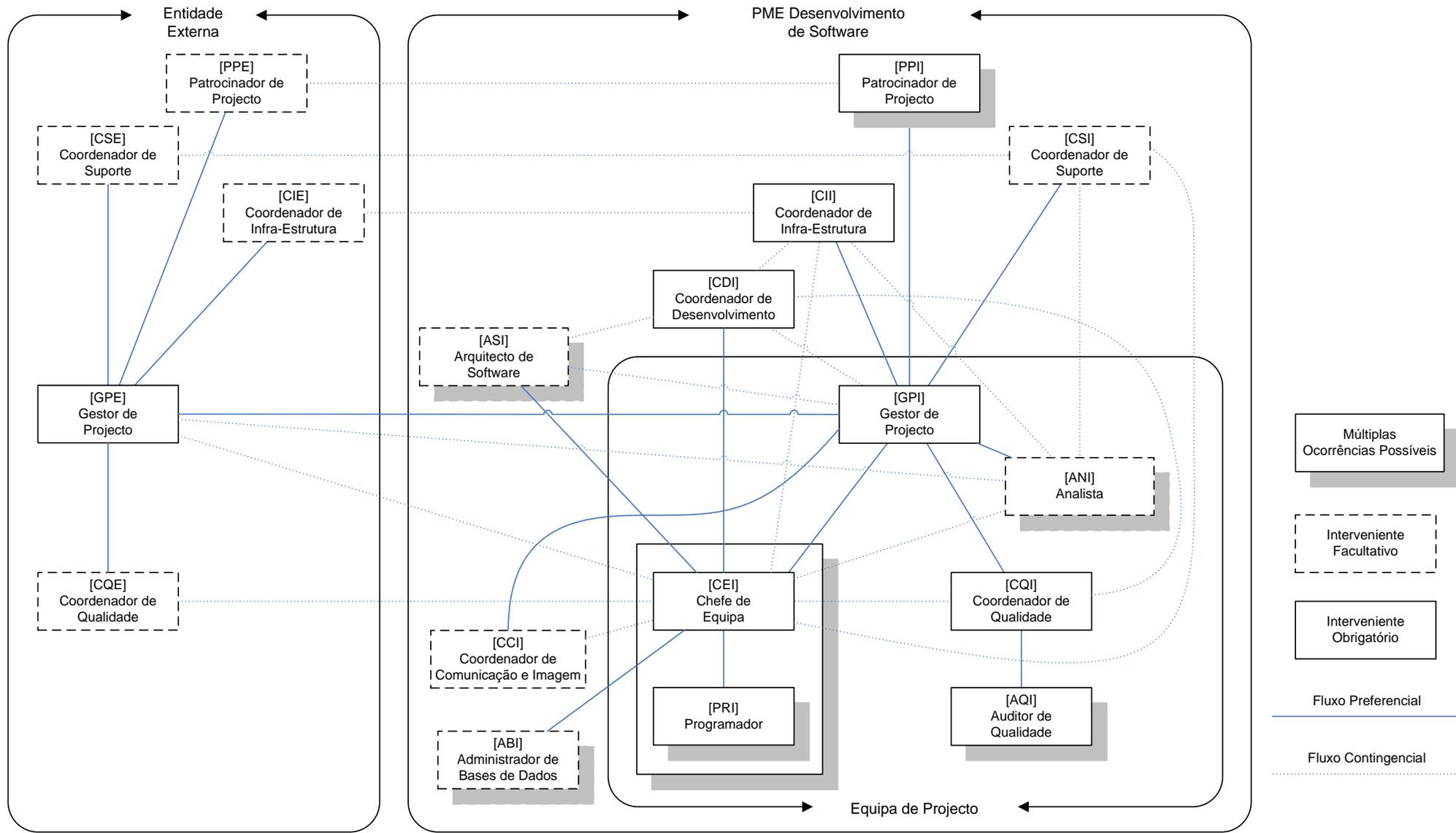
Modelo Base



Modelo Reduzido



	Modelo Base	Modelo Reduzido																				
Gestor de Projecto	<table border="1"> <tr> <td><i>Project Manager</i></td> <td><i>Business-Process Analyst</i></td> <td><i>Change Control Manager</i></td> <td><i>Deployment Manager</i></td> </tr> <tr> <td><i>Requirements Specifier</i></td> <td><i>Review Coordinator</i></td> <td><i>Test Analyst</i></td> <td></td> </tr> </table>	<i>Project Manager</i>	<i>Business-Process Analyst</i>	<i>Change Control Manager</i>	<i>Deployment Manager</i>	<i>Requirements Specifier</i>	<i>Review Coordinator</i>	<i>Test Analyst</i>		<table border="1"> <tr> <td><i>Project Manager</i></td> <td><i>Business-Process Analyst</i></td> <td><i>Change Control Manager</i></td> <td><i>Deployment Manager</i></td> </tr> <tr> <td><i>Requirements Specifier</i></td> <td><i>Review Coordinator</i></td> <td><i>Test Analyst</i></td> <td><i>System Analyst</i></td> </tr> <tr> <td><i>Business Designer</i></td> <td><i>Use Case Specifier</i></td> <td></td> <td></td> </tr> </table>	<i>Project Manager</i>	<i>Business-Process Analyst</i>	<i>Change Control Manager</i>	<i>Deployment Manager</i>	<i>Requirements Specifier</i>	<i>Review Coordinator</i>	<i>Test Analyst</i>	<i>System Analyst</i>	<i>Business Designer</i>	<i>Use Case Specifier</i>		
<i>Project Manager</i>	<i>Business-Process Analyst</i>	<i>Change Control Manager</i>	<i>Deployment Manager</i>																			
<i>Requirements Specifier</i>	<i>Review Coordinator</i>	<i>Test Analyst</i>																				
<i>Project Manager</i>	<i>Business-Process Analyst</i>	<i>Change Control Manager</i>	<i>Deployment Manager</i>																			
<i>Requirements Specifier</i>	<i>Review Coordinator</i>	<i>Test Analyst</i>	<i>System Analyst</i>																			
<i>Business Designer</i>	<i>Use Case Specifier</i>																					
Chefe de Equipa	<table border="1"> <tr> <td><i>Integrator (ou System Integrator)</i></td> <td><i>Capsule Designer</i></td> <td><i>Code Reviewer</i></td> <td><i>Designer</i></td> </tr> <tr> <td><i>Integration Tester</i></td> <td></td> <td></td> <td></td> </tr> </table>	<i>Integrator (ou System Integrator)</i>	<i>Capsule Designer</i>	<i>Code Reviewer</i>	<i>Designer</i>	<i>Integration Tester</i>				<table border="1"> <tr> <td><i>Integrator (ou System Integrator)</i></td> <td><i>Capsule Designer</i></td> <td><i>Code Reviewer</i></td> <td><i>Integration Tester</i></td> </tr> <tr> <td><i>Software Architect</i></td> <td><i>Design Reviewer</i></td> <td><i>Database Designer</i></td> <td><i>Course Developer</i></td> </tr> </table>	<i>Integrator (ou System Integrator)</i>	<i>Capsule Designer</i>	<i>Code Reviewer</i>	<i>Integration Tester</i>	<i>Software Architect</i>	<i>Design Reviewer</i>	<i>Database Designer</i>	<i>Course Developer</i>				
<i>Integrator (ou System Integrator)</i>	<i>Capsule Designer</i>	<i>Code Reviewer</i>	<i>Designer</i>																			
<i>Integration Tester</i>																						
<i>Integrator (ou System Integrator)</i>	<i>Capsule Designer</i>	<i>Code Reviewer</i>	<i>Integration Tester</i>																			
<i>Software Architect</i>	<i>Design Reviewer</i>	<i>Database Designer</i>	<i>Course Developer</i>																			
Patrocinador de Projecto	<table border="1"> <tr> <td><i>Project Reviewer</i></td> <td><i>Business Reviewer (ou Business-Model Reviewer)</i></td> <td><i>Management Reviewer</i></td> <td><i>Requirements Reviewer</i></td> </tr> </table>	<i>Project Reviewer</i>	<i>Business Reviewer (ou Business-Model Reviewer)</i>	<i>Management Reviewer</i>	<i>Requirements Reviewer</i>	<table border="1"> <tr> <td><i>Project Reviewer</i></td> <td><i>Business Reviewer (ou Business-Model Reviewer)</i></td> <td><i>Management Reviewer</i></td> <td><i>Requirements Reviewer</i></td> </tr> </table>	<i>Project Reviewer</i>	<i>Business Reviewer (ou Business-Model Reviewer)</i>	<i>Management Reviewer</i>	<i>Requirements Reviewer</i>												
<i>Project Reviewer</i>	<i>Business Reviewer (ou Business-Model Reviewer)</i>	<i>Management Reviewer</i>	<i>Requirements Reviewer</i>																			
<i>Project Reviewer</i>	<i>Business Reviewer (ou Business-Model Reviewer)</i>	<i>Management Reviewer</i>	<i>Requirements Reviewer</i>																			
Analista	<table border="1"> <tr> <td><i>System Analyst</i></td> <td><i>Business Designer</i></td> <td><i>Use Case Engineer</i></td> <td><i>Use Case Specifier</i></td> </tr> </table>	<i>System Analyst</i>	<i>Business Designer</i>	<i>Use Case Engineer</i>	<i>Use Case Specifier</i>																	
<i>System Analyst</i>	<i>Business Designer</i>	<i>Use Case Engineer</i>	<i>Use Case Specifier</i>																			
Coordenador de Desenvolvimento	<table border="1"> <tr> <td><i>Process Engineer</i></td> <td><i>Architecture Reviewer</i></td> <td><i>Tool Specialist</i></td> </tr> </table>	<i>Process Engineer</i>	<i>Architecture Reviewer</i>	<i>Tool Specialist</i>	<table border="1"> <tr> <td><i>Process Engineer</i></td> <td><i>Architecture Reviewer</i></td> <td><i>Tool Specialist</i></td> </tr> </table>	<i>Process Engineer</i>	<i>Architecture Reviewer</i>	<i>Tool Specialist</i>														
<i>Process Engineer</i>	<i>Architecture Reviewer</i>	<i>Tool Specialist</i>																				
<i>Process Engineer</i>	<i>Architecture Reviewer</i>	<i>Tool Specialist</i>																				
Programador	<table border="1"> <tr> <td><i>Implementer</i></td> <td><i>Component Engineer</i></td> </tr> </table>	<i>Implementer</i>	<i>Component Engineer</i>	<table border="1"> <tr> <td><i>Implementer</i></td> <td><i>Component Engineer</i></td> <td><i>Designer</i></td> <td><i>User-Interface Designer</i></td> </tr> <tr> <td><i>Graphic Artist</i></td> <td><i>Technical Writer</i></td> <td></td> <td></td> </tr> </table>	<i>Implementer</i>	<i>Component Engineer</i>	<i>Designer</i>	<i>User-Interface Designer</i>	<i>Graphic Artist</i>	<i>Technical Writer</i>												
<i>Implementer</i>	<i>Component Engineer</i>																					
<i>Implementer</i>	<i>Component Engineer</i>	<i>Designer</i>	<i>User-Interface Designer</i>																			
<i>Graphic Artist</i>	<i>Technical Writer</i>																					
Arquitecto de Software	<table border="1"> <tr> <td><i>Software Architect</i></td> <td><i>Design Reviewer</i></td> </tr> </table>	<i>Software Architect</i>	<i>Design Reviewer</i>																			
<i>Software Architect</i>	<i>Design Reviewer</i>																					
Coordenador de Infra-Estrutura	<table border="1"> <tr> <td><i>System Administrator</i></td> <td><i>Configuration Manager</i></td> </tr> </table>	<i>System Administrator</i>	<i>Configuration Manager</i>	<table border="1"> <tr> <td><i>System Administrator</i></td> <td><i>Configuration Manager</i></td> </tr> </table>	<i>System Administrator</i>	<i>Configuration Manager</i>																
<i>System Administrator</i>	<i>Configuration Manager</i>																					
<i>System Administrator</i>	<i>Configuration Manager</i>																					
Coordenador de Qualidade	<table border="1"> <tr> <td><i>Test Manager</i></td> <td><i>Test Designer</i></td> </tr> </table>	<i>Test Manager</i>	<i>Test Designer</i>	<table border="1"> <tr> <td><i>Test Manager</i></td> <td><i>Test Designer</i></td> <td><i>Use Case Engineer</i></td> </tr> </table>	<i>Test Manager</i>	<i>Test Designer</i>	<i>Use Case Engineer</i>															
<i>Test Manager</i>	<i>Test Designer</i>																					
<i>Test Manager</i>	<i>Test Designer</i>	<i>Use Case Engineer</i>																				
Coordenador de Comunicação e Imagem	<table border="1"> <tr> <td><i>User-Interface Designer</i></td> <td><i>Graphic Artist</i></td> </tr> </table>	<i>User-Interface Designer</i>	<i>Graphic Artist</i>																			
<i>User-Interface Designer</i>	<i>Graphic Artist</i>																					
Coordenador de Suporte	<table border="1"> <tr> <td><i>Course Developer</i></td> <td><i>Technical Writer</i></td> </tr> </table>	<i>Course Developer</i>	<i>Technical Writer</i>																			
<i>Course Developer</i>	<i>Technical Writer</i>																					
Administrador de Bases de Dados	<table border="1"> <tr> <td><i>Database Designer</i></td> </tr> </table>	<i>Database Designer</i>																				
<i>Database Designer</i>																						
Auditor de Qualidade	<table border="1"> <tr> <td><i>System Tester</i></td> </tr> </table>	<i>System Tester</i>	<table border="1"> <tr> <td><i>System Tester</i></td> </tr> </table>	<i>System Tester</i>																		
<i>System Tester</i>																						
<i>System Tester</i>																						



Múltiplas Ocorrências Possíveis

Intervente Facultativo

Intervente Obrigatório

Fluxo Preferencial

Fluxo Contingencial