

Universidade do Minho

Sandra Ribeiro de Almeida Brandão

**Validação de Software
Dedicado à Gestão Documental**

Tese de Mestrado
Engenharia e Gestão de Sistemas de Informação

Trabalho efectuado sob a orientação do
**Professor Doutor Ricardo Jorge Silvério Magalhães
Machado**

Outubro de 2009

Agradecimentos

Ao Professor Doutor Ricardo Jorge Silvério Magalhães Machado, coordenador desta dissertação, por todo o empenho demonstrado e tempo dispendido.

À minha família, em particular aos meus pais, irmão e avós, pelo apoio incondicional, incentivo e compreensão nas horas mais difíceis.

Validação de Software dedicado à Gestão Documental

Resumo

O presente documento surge como uma introdução ao projecto de Dissertação, que emerge no âmbito da unidade curricular com o mesmo nome, leccionada no decorrer do quarto semestre do Mestrado em Engenharia e Gestão de Sistemas de Informação (MEGSI).

Nos dias que correm, a informação assume um papel privilegiado no contexto organizacional, razão pela qual a utilização de sistemas de gestão documental para o seu tratamento, organização e correcta utilização, é cada vez mais frequente. Neste contexto, a validação surge como uma vantagem competitiva, assegurando que os sistemas estão optimizados para as necessidades do cliente, e garantindo o seu correcto funcionamento. No entanto, este não é um procedimento comum. E como tal, não existe ainda uma metodologia de validação adaptada ao contexto específico do software dedicado à gestão documental. O propósito deste projecto é, então, o estudo desta problemática, com vista ao desenvolvimento de um modelo global de validação de sistemas de gestão documental.

Palavras-chave: software, sistemas de gestão documental, validação.

Validation of Document Management Software

Abstract

This document appears as an introduction to the Dissertation project, which comes under the curricular unit with the same name, taught during the fourth semester of the Master in Engineering and Management Information Systems (MEGSI).

Nowadays information takes a key role in the organizational context, which is why the use of document management systems for treatment, organization, and proper use of it is increasing. In this context, the validation comes as a competitive advantage, ensuring that systems are optimized to customer needs as well as ensuring their proper functioning.

However, this is not a common procedure. And that's why there isn't any validation methodology adapted to the specific software dedicated to document management. The purpose of this project is the research on the issue with a view to developing a global model of validation adapted to document management systems.

Keywords: software, document management systems, validation.

Índice

1.Introdução	1
1.1.Validação de software dedicado à gestão documental	1
1.2.Estruturação do documento	2
1.3.Abordagem de investigação	4
2.Validação e teste de software	6
2.1.Validação vs. Teste.....	8
2.2.Vantagens da validação de software	11
2.3.Princípios de validação	12
2.4.O processo de validação	14
2.4.1.Enquadramento	14
2.4.2.Técnicas de validação.....	21
2.4.3.Etapas do processo de validação:	22
2.5.A norma ISO/IEC 9126	27
2.5.1.Modelos de qualidade	29
2.5.2.Métricas	31
3.Sistemas de Gestão Documental.....	33
3.1.Documentos: o que são?.....	33
3.2.Porquê gestão documental?.....	35
3.3.SGD – definições.....	37
3.4.SGD – Definição de uma estratégia	38
3.5.SGD – Vantagens e Desvantagens.....	40
3.6.Características dos SGD	42
3.6.1.Importância do ciclo de vida dos documentos.....	42
3.6.2.Respeita os fluxos de trabalho.....	46

3.6.3.Tem por base um repositório.....	47
3.6.4.Implementa funcionalidades genéricas	48
3.6.5.Podem ser sistemas colaborativos.....	50
3.6.6.Privilegiam a segurança.....	52
3.7.Validação de SGD.....	54
3.8.SGD - Conclusões.....	57
4.Modelo conceptual de validação	59
4.1.Definição de um modelo conceptual de validação	59
4.2.Validação durante a fase de desenvolvimento de SGD.....	61
4.3.Validação durante a fase de conclusão do desenvolvimento de SGD ..	78
4.4.Validação durante a fase de utilização de SGD	95
4.5.Modelo conceptual de validação – conclusões	101
5.Conclusão	103
Anexos.....	111
Métricas de validação durante a fase de desenvolvimento de SGD	111
Funcionalidade.....	111
Confiabilidade.....	114
Usabilidade	117
Eficiência	121
Manutenção	123
Portabilidade	125
Métricas de validação durante a fase de conclusão do desenvolvimento de SGD	130
Funcionalidade.....	130
Confiabilidade.....	134
Usabilidade	138
Eficiência	143

Manutenção	146
Portabilidade	151
Métricas de validação durante a fase de utilização de SGD	155
Efectividade	155
Produtividade	156
Segurança.....	157
Satisfação	157

Índice de Figuras

Figura 1: validação e teste	9
Figura 2: ciclo de vida de software	15
Figura 3: modelos de qualidade	28
Figura 4: modelo de qualidade interna e externa	30
Figura 5: ligação entre os diferentes tipos de qualidade	32
Figura 6: ciclo de vida de um documento	43
Figura 7: estrutura do modelo conceptual de validação	60
Figura 8: validação interna no ciclo de vida do software.....	62
Figura 9: <i>outputs</i> validação interna	62
Figura 10: validação interna	64
Figura 11: Métricas de funcionalidade.....	66
Figura 12: métricas de confiabilidade	68
Figura 13: Métricas de usabilidade	71
Figura 14: métricas de eficiência	72
Figura 15: métricas de manutenção	74
Figura 16: métricas de portabilidade	75
Figura 17: validação externa no ciclo de vida do software	78
Figura 18: <i>outputs</i> validação externa	79
Figura 19: validação externa	80
Figura 20: métricas de funcionalidade	85
Figura 21: métricas de confiabilidade	86
Figura 22: métricas de usabilidade.....	88
Figura 23: métricas de eficiência	91

Figura 24: métricas de manutenção	92
Figura 25: métricas de portabilidade	94
Figura 26: validação em uso no ciclo de vida do software	96
Figura 27: validação em uso	97
Figura 28: métricas de efectividade.....	98
Figura 29: métricas de produtividade	99
Figura 30: métricas de segurança	100
Figura 31: métricas de satisfação	101
Figura 32: actividades de validação.....	104

Lista de Acrónimos

URS – *User Requirements Specification*

FRS – *Functional Requirements Specification*

DRS – *Design Requirements Specification*

ERP – *Enterprise Resource Planning*

FDA - *Food and Drugs Administration*

RV - *Relatório de Validação*

TR – *Test Report*

TS – *Test Specification*

SCORE - *Solution for Compliance in a Regulated Environment*

VMP - *Validation Master Plan*

1.Introdução

O projecto de Dissertação surge no segundo e último ano do Mestrado em Engenharia e Gestão de Sistemas de Informação, como o derradeiro desafio para os alunos que aspiram à obtenção do grau de Mestre.

No entanto, este grau exige de nós algo mais do que foi explorado até ao momento – um Mestre numa determinada área deverá ter a capacidade de resolver problemas que se apresentem fora do ambiente com que está familiarizado, com acesso a informação incompleta ou limitada, de forma autónoma e eficaz (Arnhold *et al.*, 2002).

É então, tendo em vista os objectivos acima mencionados, que surge o projecto de Dissertação, no âmbito da unidade curricular com o mesmo nome, que propõe aos alunos escolher um tema acerca do qual elaborarão uma dissertação de natureza científica, com tendo em vista a resolução de um dado problema, devidamente enquadrado dentro da área em questão.

1.1. Validação de software dedicado à gestão documental

O tema escolhido para esta Dissertação foi a *Validação de sistemas computadorizados dedicados à gestão documental*. Nos dias que correm, a informação ganha uma importância cada vez maior no seio organizacional. Desta forma, a sua organização, tratamento e acesso são alguns dos pontos-chave para o sucesso empresarial no século XXI. E esta é a razão pela qual os sistemas de gestão documental tem ganho

cada vez mais relevo no ramo das soluções informáticas para organizações. A optimização do seu desempenho é, portanto, uma questão que tem todo o interesse em ser explorada, e uma das vias para que isso aconteça é a implementação do processo de validação a este tipo de sistemas.

Actualmente, a validação de sistemas de gestão documental é um processo pouco conhecido, e utilizado maioritariamente por empresas do ramo da saúde e alimentação, por estarem sujeitas a legislação que a tal obriga. Mas não seria uma mais-valia a implementação deste processo aos sistemas de gestão documental?

A resposta a esta questão é um dos propósitos desta dissertação. Para isso, está previsto o cumprimento dos seguintes objectivos:

- Estudo e compreensão do que é, de facto, a validação de um sistema de gestão documental;
- Identificação de qual a utilidade e vantagens subjacentes à validação de sistemas de gestão documental;
- Desenvolvimento de um modelo de validação geral, adaptado ao contexto particular dos sistemas de gestão documental;
- Colmatar a lacuna existente no que diz respeito à literatura referente ao assunto em questão, já que não existe ainda qualquer modelo de validação dirigido para este contexto específico.

1.2. Estruturação do documento¹

O presente documento seguirá a seguinte estrutura: vimos já a contextualização da dissertação no âmbito do Mestrado em Engenharia e

¹ Todas as citações presentes no documento são de tradução livre, sendo efectuadas pela autora.

Gestão de Sistemas de Informação, bem como a descrição do tema escolhido e problema que lhe está implícito. Nas páginas seguintes será apresentada uma secção dedicada às abordagens de investigação seguidas no decorrer deste projecto.

Terminada a introdução e descrição da tese, seguir-se-á um capítulo dedicado à problemática da validação. Neste capítulo serão apresentados os fundamentos teóricos relativos a este tema, de uma forma geral. O objectivo subjacente a este capítulo é a contextualização do leitor na temática da validação de software. Adicionalmente, será ainda apresentada a norma ISO/IEC 9126, que estará na base do modelo conceptual de validação, a desenvolver em capítulos posteriores.

O capítulo que se seguirá a este, dirá respeito aos sistemas de gestão documental – aqui será explicado ao leitor o que são estes sistemas, qual a sua utilidade e especificidades a estes associadas. Para além disto, será feita a relação entre a validação – apresentada no capítulo anterior – e estes sistemas em particular, entrando então no tema que dá nome a esta dissertação – a validação de sistemas de gestão documental.

O modelo propriamente dito, não será apresentado até ao capítulo 4. Apenas depois de compreender devidamente os conceitos de validação, sistemas de gestão documental, e a junção dos dois, se poderá dar início ao desenvolvimento de um modelo de validação geral, aplicável a este tipo de sistemas em particular. Esta tarefa será executada então no capítulo que se segue.

Para terminar, será apresentado um capítulo de conclusão, que conterà as elações obtidas através dos resultados verificados no capítulo anterior, bem como as perspectivas futuras acerca do tema em questão e quaisquer considerações finais que se revelem apropriadas.

1.3. Abordagem de investigação

"The use of a systematic method is the soul of research"

(Berndtsson *et al.*, 2008)

Qualquer projecto, para ser bem sucedido, exige um bom planeamento. Este caso não é excepção, e uma das componentes desse planeamento foi a definição das abordagens de investigação a adoptar no decorrer da dissertação.

Tendo em conta a natureza desta dissertação, foram seleccionados dois métodos de pesquisa: análise de literatura e entrevista.

A análise de literatura é, a base na qual assenta todo o trabalho. Para compreender a temática da validação de sistemas de gestão documental, foram realizadas várias pesquisas acerca dos conceitos-chave subjacentes ao assunto, de forma a compreender o que pensam os autores acerca do mesmo, e permitindo elaborar uma contextualização do tema no âmbito da comunidade científica.

Sendo este um trabalho de investigação, a componente de análise literária esteve sempre presente, pois foi através da mesma que foi possível compreender as bases para desenvolver as teorias que serão apresentadas mais à frente neste documento. À medida que a dissertação foi sendo desenvolvida, foram sempre surgindo novos conceitos, novas metodologias, e novas necessidades de contextualização, razão pela qual este método não foi nunca abandonado.

Para além da análise de literatura, foi ainda utilizado o método de entrevista, numa fase inicial desta dissertação. De facto, nos primeiros meses, foi realizado um estágio numa empresa do ramo da saúde, onde foi possível, não só obter contacto com um sistema de gestão documental em particular – IBM SCORE – como também realizar entrevistas com os

responsáveis pela validação do mesmo, permitindo-me desta forma compreender de uma forma mais clara algumas das etapas que compõe o processo, recursos envolvidos e dificuldades experimentadas. Estas informações, apesar de serem de extrema utilidade para a compreensão acerca do que é a validação e o que são sistemas de gestão documental, dizem apenas respeito a um tipo de sistema específico, não estando, portanto, reproduzidas no presente documento.

2. Validação e teste de software

Prendendo-se o tema desta dissertação com a validação de Sistemas de Gestão Documental, é necessário, antes de mais, enquadrar o leitor no que diz respeito à validação de software, de modo geral. Só depois de compreendermos claramente o que é a validação de software é que poderemos avançar para a contextualização desta no campo desse software específico que são os sistemas de gestão documental. Desta forma, o presente capítulo será dedicado ao estudo da validação. Para tal, foi dividido em várias secções: para começar, e ainda dentro desta secção, será feita a descrição acerca do que é a validação, de uma perspectiva geral. Na secção seguinte será feita a distinção entre este conceito e o conceito de teste. Por último, serão apresentadas, numa nova secção, as características gerais de validação, bem como as metodologias mais aplicadas neste âmbito.

Antes de mais, é necessário questionar: o que é, efectivamente, o processo de validação?

Paz (n.d.), define validar como *"o acto de comprovar, de acordo com as normas e padrões previamente estabelecidos, que os processos de facto conduzem aos resultados esperados e projectados. A validação consiste em estabelecer evidências documentadas, com alto nível de segurança, de que um processo específico terá desempenho efectivo e produzirá consistentemente um resultado que atenda às suas especificações e características previamente determinadas."*

O processo de validação de software é principalmente regulado pela FDA (*Food and Drug Administration*). Isto deve-se ao facto de este ser um processo obrigatório na indústria farmacêutica, alimentar e cosmética (Odegaard, 2006). Esta obrigatoriedade deve-se a dois factores: em

primeiro lugar, e sendo estas áreas de elevado impacto junto dos consumidores por lidarem directamente com a sua saúde, como forma de garantir a qualidade do produto. Por outro lado, devido à necessidade de expandir o negócio a nível global, é indispensável garantir que o sistema respeita um determinado conjunto de normas qualitativas impostas por agências reguladoras internacionais, como por exemplo a FDA. No entanto a validação poderá ser aplicada a quaisquer ramos de negócio. É então pelo motivo de grande parte da informação produzida ser da responsabilidade da FDA, que, apesar de o sistema aqui abordado não ser específico para organizações nas áreas da saúde e alimentação, estarão presentes no documento referências a esta instituição.

Outra definição, publicada pela FDA (2002), descreve validação como sendo uma ferramenta utilizada para assegurar a qualidade do software e operações automatizadas, uma vez que permite aumentar a usabilidade e fiabilidade do sistema, resultando em menores taxas de erro. Para além disto, o processo de validação permite identificar áreas críticas e efectuar possíveis melhorias, de modo a causar reduções nos custos, a longo prazo.

É comum, por vezes, depararmo-nos com uma outra designação para este conceito – V&V, ou *Verification and Validation*. V&V refere-se também ao processo de garantir que um dado produto, serviço ou sistema cumpre os requisitos para o qual foi concebido. No entanto, apesar de o conceito ser semelhante, é importante referir a diferença, que, de resto, é óbvia: a presença da palavra *Verification* – verificação - nesta designação.

Verificação e validação não são a mesma coisa. Enquanto a primeira designação diz respeito a um processo de controlo de qualidade, de carácter avaliativo, a segunda refere-se ao processo de assegurar a qualidade do produto, serviço ou sistema em questão. Por outras palavras, a verificação foca-se nas especificações necessárias para que

um sistema seja correctamente implementado, e a validação foca-se no utilizador, e na satisfação das suas necessidades.

A definição de validação como processo de garantir que um sistema cumpre todos os requisitos estabelecidos é sólida e coerente, não deixando margem para grandes dúvidas. Todas as definições encontradas no decorrer da pesquisa referem que o processo de validação tem por objectivo assegurar que um sistema faz aquilo para o qual foi criado, dando especial ênfase à documentação de todo este processo, que deve ser rigorosa e específica.

No entanto, se não existem grandes dúvidas quanto ao que é a validação, o mesmo não acontece relativamente às metodologias necessárias para que o processo de validação, propriamente dito, seja correctamente executado. Este assunto será abordado mais à frente neste capítulo.

2.1. Validação vs. Teste

O conceito de validação é, tendencialmente, confundido com o conceito de teste. Tratando-se a validação de um processo que procura assegurar a qualidade de um dado sistema informático, pode ser confundido com o processo de teste, o que não é, de todo, verdade.

O dicionário comum descreve o teste como sendo uma *"prova para verificar o bom funcionamento"*. Se nos embrenharmos um pouco mais na área técnica, deparamo-nos com definições mais elaboradas, como é o caso de Hetzel (1988) , que define o teste de software *como "qualquer actividade que tem como objectivo avaliar um atributo ou capacidade de um programa ou sistema e determinar que este cumpre os requisitos exigidos"*. Myers (1979) define ainda teste como *"o processo de executar um programa ou sistema com o intuito de encontrar erros"*.

Ora, se a validação procura garantir a qualidade, e o teste procura encontrar erros, podemos desde já estabelecer uma diferença entre os dois conceitos. E, a partir daqui, é fácil compreender qual o sentido do teste no âmbito da validação de software. A validação de software pode assumir várias metodologias, cada uma delas contendo um conjunto de etapas específicas. Mas uma etapa que será, certamente, comum a todas elas, é o teste.

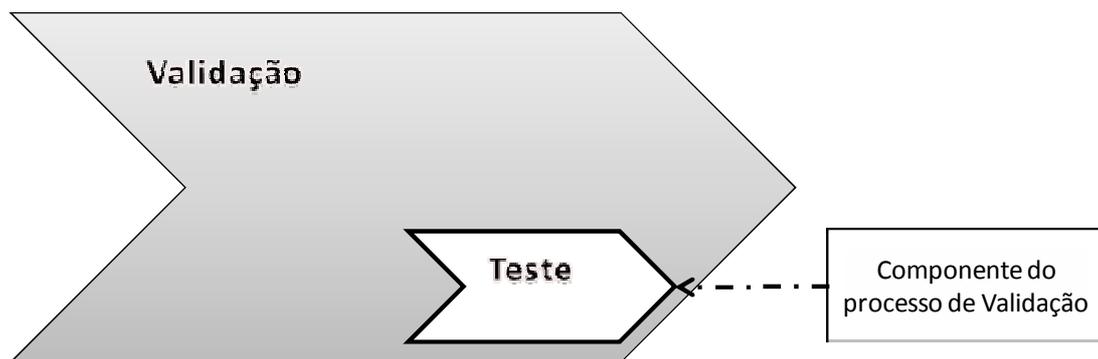


Figura 1: validação e teste

Um teste implica um conjunto de procedimentos que são efectuados no sistema. Poderão ser recriações de situações normais, ou de situações pouco prováveis, tudo com o propósito de verificar se o software responde correctamente em qualquer situação.

Mas, porquê testar? A resposta é simples. Cada organização possui o seu ambiente específico. O ambiente no qual determinado software vai ser inserido não é igual àquele no qual foi testado pelo fabricante. Daí que o processo de teste seja necessário como forma de prevenção, para evitar falhas na segurança, perdas de dados, ou outros problemas que poderão advir de *bugs* ou “buracos” no sistema (Muegge, 2007). Mas o processo de validação não ocorre somente aquando da implementação de um sistema, muito pelo contrário. A validação deve ser periódica. Aliás, todas as empresas que actuam no ramo da saúde e alimentação são, como já foi referido acima, obrigadas a isso. E então porquê testar um sistema que já foi testado anteriormente? A resposta a esta questão é muito

simples: as organizações não são entidades estáticas – vão sofrendo alterações ao longo do tempo, assim como os seus sistemas. O propósito destas validações e testes periódicos ao sistema é, portanto, garantir que, com o passar do tempo e evolução dos processos na organização, a qualidade continua a ser constante.

Portanto, de acordo com o que vimos nos parágrafos anteriores, o processo de validação pressupõe várias etapas, sendo que uma delas – talvez a menos controversa de todas – consiste na execução de testes ao sistema. Segundo a FDA (2002), durante os testes a executar no decorrer deste processo, devem ser realizados os seguintes procedimentos:

- Verificar se o ambiente de teste, dados de entrada e material de suporte estão conforme o descrito;
- Seguir todos os passos do procedimento;
- Caso seja necessário, é permitido rasurar o documento, desde que esteja presente a data e rubrica do testador;
- Todos os resultados, incidentes e imprevistos devem ser registados;
- Todas as folhas devem ser verificadas, datadas e assinadas.

Um teste pressupõe ainda a existência de uma testemunha – alguém que assiste a todo o processo e atesta a veracidade do documento no final.

Em jeito de conclusão, é possível então explicar que a razão para se chamar validação, e não teste a este processo, é bastante simples: estes dois conceitos não se referem ao mesmo processo. Desta forma, os testes estão definidos como sendo parte do processo de validação. Ou seja, para que um sistema possa ser considerado como validado, é necessária, entre outras coisas, a execução de diversos testes. Esses testes, em conjunto com a execução dos outros procedimentos, que serão referidos mais à frente neste documento, possibilitam a conclusão do processo de validação.

Ou seja, é absolutamente errado substituir o conceito de validação por teste, e vice-versa, pois, apesar de poderem existir testes sem validação, não pode nunca haver validação sem testes.

2.2. Vantagens da validação de software

Como foi referido acima neste documento, a validação pode ocorrer, tanto durante o desenvolvimento do produto de software, como depois da implementação deste, como forma de assegurar que todos os requisitos são preenchidos. A FDA define que, uma vez que o software está, geralmente, integrado num sistema de hardware de maior dimensão, a validação deverá comprovar que todos os requisitos do software foram implementados. E este facto constitui, só por si, uma vantagem – o controlo.

Associadas à vantagem de ter um maior controlo acerca do funcionamento do sistema, estão ainda outras duas vantagens – a usabilidade e fiabilidade. A validação procura garantir a qualidade do sistema. Quando um sistema é periodicamente validado, os possíveis erros são corrigidos, e, no final do processo de validação, o sistema deverá estar a funcionar de acordo com os requisitos estabelecidos. Desta forma, as taxas de falha serão reduzidas, exigindo um menor número de acções correctivas e um menor risco para a organização que utiliza o sistema. Isto traduzir-se-á em ganhos, não só monetários, mas também temporais, para a própria empresa.

Também a longo prazo se tem verificado uma redução dos custos das organizações que adoptam este processo, uma vez que *"tornam mais fácil e menos dispendioso modificar e revalidar alterações ao software"* (FDA, 2008), isto porque os custos com a manutenção de software são, por norma, bastante elevados.

2.3. Princípios de validação

Para cada tipo de sistema em questão, o processo de validação vai sofrer alterações e adaptações. Não obstante, existe um conjunto de princípios gerais a serem considerados quando falamos de validação de software. Esses princípios são, segundo a FDA (2002), os seguintes:

Requisitos. Antes de mais, é necessária a criação de um documento para especificação de requisitos de software. Este documento constituirá a base para o processo de validação, uma vez que será através dele que será feita a comparação entre os requisitos definidos e os resultados obtidos.

Prevenção de defeitos. Os testes efectuados ao sistema poderão, de facto, detectar erros no mesmo. Contudo, os testes não são 100% fiáveis, uma vez que nenhum software pode ser testado de forma exhaustiva, garantindo que não existem quaisquer erros. O teste é uma actividade indispensável, de facto. Mas aliado a uma prevenção eficaz de defeitos, os resultados serão bem mais satisfatórios.

Tempo e esforço. Um bom projecto de validação exige, inevitavelmente, tempo e esforço. O planeamento deste processo deverá começar antecipadamente, sendo que, para a validação de software acabado de desenvolver, deverá começar ainda na fase de desenho e planeamento do sistema. Isto para que, no final, a conclusão acerca da validação do sistema seja sólida e bem sustentada.

Ciclo de vida do software. Cada software tem o seu ciclo de vida específico. E o processo de validação acontece numa determinada fase desse ciclo. Assim, é necessário ter em conta esse aspecto, e as suas condicionantes.

Planeamento. O processo de validação de software deve ser definido e controlado através de um plano. Desse documento devem constar informações, como quais os objectivos a alcançar, qual a abordagem a seguir, recursos a utilizar, tarefas, entre outras informações relevantes. O plano de validação é uma ferramenta muito importante para a qualidade do processo de validação.

Procedimentos. O processo de validação de software é executado através de procedimentos. Estes são o meio de estabelecer como é que o processo vai ser executado, identificando acções que devem ser executadas para esse propósito.

Validação de software após uma alteração. Um produto de software é, por si só, bastante complexo, e uma alteração, por muito pequena que seja, numa das suas componentes, poderá ter impacto em todo o sistema. Assim, um sistema que estava validado antes de uma alteração, deixa de o estar após esta. Torna-se então necessário efectuar uma análise de validação, com o propósito de validar a alteração que ocorreu, bem como determinar o impacto que esta teve no sistema global.

Alcance da validação. O alcance que o processo de validação vai ter, deverá ser definido unicamente com base na complexidade do software. Não importa o tamanho da organização, ou as restrições a nível de recursos. O que importa é unicamente o software, e o risco associado à sua utilização para os efeitos propostos. É, tendo em conta estes factores, que são seleccionadas as actividades, tarefas e itens de trabalho que estarão presentes ao longo do processo. Quanto maior for o risco associado à utilização do software, maior deverá ser o alcance da validação, de forma a prevenir danos cada vez maiores.

Revisão independente. É recomendado que a validação seja efectuada por entidades externas, ou seja, que a equipa responsável pela validação do sistema não seja a mesma que o utiliza, particularmente

para software de maior risco, e cuja validação deverá ser mais extensa. Algumas organizações contratam uma equipa externa para validar o sistema, mas esta não é a única solução. Mesmo dentro da própria empresa, poderá definir-se uma equipa de validação, desde que os seus membros não estejam envolvidos na concepção ou implementação do sistema.

Flexibilidade e responsabilidade. Tal como já foi referido no início desta secção, o processo de validação sofre alterações consoante o contexto em que é aplicado, daí resultando que a implementação específica destes princípios possa também ser diferente consoante a aplicação à qual se adapte. O software é desenvolvido num conjunto muito abrangente de ambientes, e a sua utilização pode variar consoante o contexto em que é aplicado, razão pela qual não é possível definir um modelo de validação que se adapte a todos os sistemas e em todas as circunstâncias. Em alternativa, são definidas linhas orientadoras, com algum grau de flexibilidade, para se poderem adaptar a todos os casos. É importante referir, no entanto, que, e devido aos factos apresentados acima, cada organização é, em última instância, responsável pelo seu processo de validação.

2.4. O processo de validação

2.4.1. Enquadramento

Para melhor compreendermos o processo de validação, é necessário, antes de mais, compreender onde é que este se enquadra, no que diz respeito ao ciclo de vida do sistema em questão.

Não é possível definir um ciclo de vida específico para todos os sistemas. Cada caso é um caso, e cada organização opta pelos produtos que mais se adequam às suas necessidades. O ciclo de vida de um

software deverá acompanhar todo o seu desenvolvimento, desde a criação até que seja considerado obsoleto, e inclui um conjunto de actividades, representadas na figura abaixo.

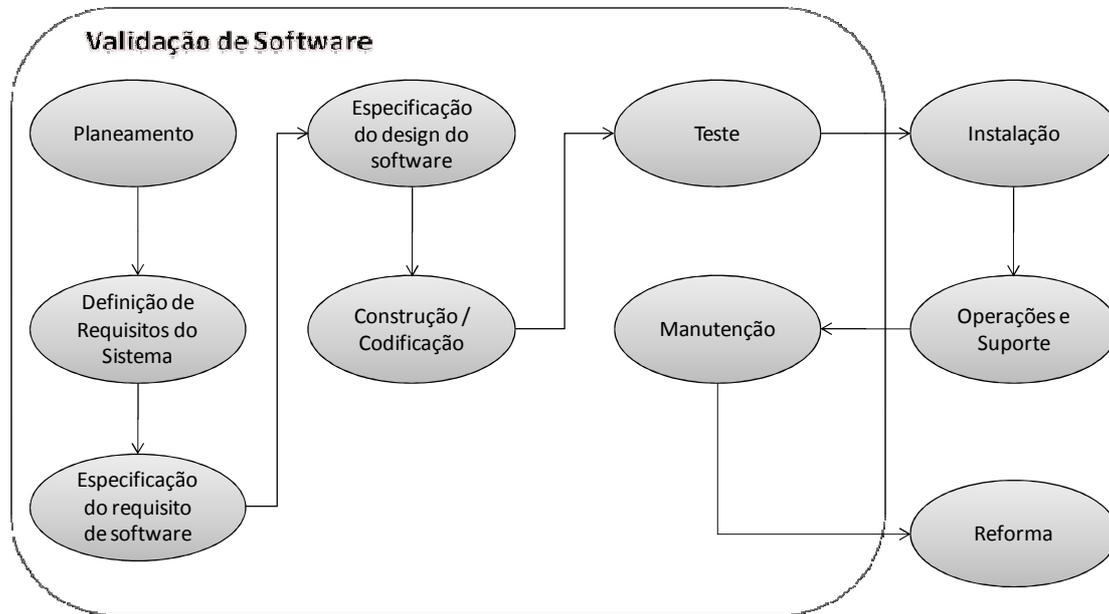


Figura 2: ciclo de vida de software

De entre as actividades acima apresentadas, as seleccionadas correspondem, segundo a FDA (2002), às fases do ciclo de vida do software durante as quais o processo de validação ocorre. De salientar que estas são apenas orientações genéricas, não dizendo, portanto, respeito a nenhum tipo de sistema em particular, e sendo, como já foi referido acima, flexíveis, consoante o contexto de aplicação.

Comecemos então pela fase de **planeamento**. Será fácil perceber pela designação dada a esta etapa que a validação do planeamento resultará num plano que identifique os recursos necessários, revisão dos requisitos, tarefas e procedimentos para relatório de anomalias e suas resoluções, modelo do ciclo de vida do software e actividades associadas a este. Para além disto, o plano deverá conter:

- As tarefas associadas a cada actividade;
- Uma lista de factores qualitativos a ter em atenção;
- Métodos e procedimentos para cada tarefa;

- Critérios de aceitação;
- Critérios para definir se os *outputs* estão de acordo com os requisitos de *Input*;
- *Inputs* e *outputs* para cada tarefa;
- Recursos e suas responsabilidades para cada tarefa;
- Potenciais riscos;
- Documentação acerca das necessidades do utilizador.

A cada uma destas tarefas estão associados recursos físicos e humanos, que deverão ser claramente identificados no plano. Desse mesmo plano deverão constar também os potenciais riscos associados à tarefa em questão, e respectivos planos de contingência.

Para esta fase, deverão ainda ser criados procedimentos, para relatar e resolver anomalias identificadas no decorrer do processo de validação. A validação, na fase de planeamento, resume-se à execução da tarefa de desenvolvimento de um plano de verificação e validação. Isto inclui:

- Definição de tarefas de verificação e validação;
- Definição de critérios de aceitação;
- Planeamento temporal e alocação de recursos;
- Relatório de requisitos formais de concepção;
- Relatório de outros requisitos técnicos.

Seguem-se as fases de **definição de requisitos do sistema e especificação detalhada dos requisitos de software**. Nestas fases, deverá ser feita a identificação, análise e documentação acerca do software e o propósito para o qual será utilizado. É neste momento que deverá ser desenvolvido o documento de especificação de requisitos do sistema, que, como veremos mais à frente, é crucial para o processo de validação – de facto, não é possível validar um sistema sem saber quais os requisitos que se pretendem satisfazer com a sua utilização. Em termos gerais, este documento contém informações acerca de:

- *Inputs* recebidos pelo sistema;
- Funções que o software irá executar;
- Performance esperada;
- Definição de interfaces (internas e para com o utilizador);
- De que forma irão os utilizadores interagir com o sistema;
- O que será considerado um erro, e de que forma é que o sistema reagirá a estes;
- Tempos de resposta esperados;
- Todos valores específicos, limites e predefinições que o sistema aceitará;
- Especificações de segurança a serem implementadas no software.

A fase de **especificação dos requisitos de concepção do software** consiste na passagem dos requisitos anteriormente definidos para uma representação lógica e física do software a ser implementado, ou seja, uma descrição do que o software deve fazer, e de que forma o deve fazer, tornando-se esta etapa na última preparação antes da codificação. É neste momento que deverão ser desenvolvidos gráficos, diagramas de estado, ferramentas de prototipagem e planos de teste. Ainda nesta fase, deverão ser efectuadas análises de tarefas e funções, de risco, bem como testes de usabilidade com a participação de alguns dos futuros utilizadores do sistema.

No final desta etapa deverá ser produzido um documento referente à especificação dos requisitos de concepção do software, documento esse que deverá conter:

- A especificação dos requisitos de software, que foram definidos na etapa anterior;
- Análise de risco do software;
- Orientações ao nível de processos de codificação e desenvolvimento;

- Documentação relativa ao sistema, descrevendo o seu contexto de utilização;
- Hardware a utilizar;
- Parâmetros de medida;
- Estrutura lógica e algoritmos de programação;
- Diagramas de fluxo e estrutura de dados;
- Definição e descrição de variáveis;
- Definição de software de suporte;
- Medidas de segurança, tanto físicas como lógicas.

Toda esta documentação é relevante para o processo de validação, uma vez que é, de todo, impossível, validar um sistema sem ter um conhecimento exacto acerca do contexto em que este funciona. Tal como já foi referido neste documento, a informação contida neste capítulo é apenas uma orientação de nível geral, podendo, no entanto, existir casos em que uma tarefa específica não se aplique.

A definição da arquitectura do software reveste-se de uma elevada importância para o processo de validação, pois poderá significar um maior ou menor esforço neste âmbito quando são implementadas mudanças no sistema.

A fase de **Construção/codificação**, tal como o nome indica, consiste na passagem de todas as especificações anteriormente definidas para código fonte. A validação do sistema nesta fase é efectuada através da elaboração uma análise de rastreabilidade do código fonte, de modo a verificar que:

- Cada elemento contido no documento de especificação de software foi, efectivamente, implementado, e, por outro lado, cada função implementada no código existe na especificação de software e análise de risco;
- Os testes para as funções implementadas estão definidos no documento de especificação de software e análise de risco.

Ainda dentro do ciclo de vida de um produto de software, segue-se a fase de **Teste**, que poderá ter duas vertentes: o teste efectuado pelo desenvolvedor do sistema e o teste efectuado pelo lado do utilizador.

O teste propriamente dito consiste em executar um conjunto acções, com um conjunto de *Inputs* associados e verificar que os *outputs* obtidos estão de acordo com as expectativas predefinidas na documentação elaborada nas fases anteriores.

A importância do teste por parte do desenvolvedor reside no facto de ser produzida documentação que comprova a validação inicial do sistema, e que servirá de auxílio em futuras validações. Para que nada seja esquecido, este tipo de testes devem ser planeados com a maior antecedência possível, e cada plano de teste deverá conter informações acerca do ambiente de teste, metodologias, *Inputs*, procedimentos, resultados esperados, recursos a utilizar e documentação associada.

O teste efectuado do ponto de vista do utilizador comum é essencial à validação do software, uma vez que as autoridades responsáveis pela regulação da qualidade de sistemas informáticos exigem a aplicação de processos que demonstrem a correcta instalação e inspecção do sistema. Desta forma, é necessário que a instalação do sistema em questão cumpra determinados requisitos, ou seja, é necessária a sua validação, através do teste sob a óptica do utilizador.

Tal como acontecia para os testes efectuados pelo desenvolvedor do sistema, também no caso do utilizador comum o teste deverá seguir um plano pré-definido. Desse plano deverá constar o resumo do teste e condições nas quais este será efectuado. Aquando da execução do teste, deverá ser elaborado um relatório do mesmo, contendo os dados de *Input* e resultados obtidos. Durante o processo de teste, deverá ainda ser assegurado que todos os componentes do sistema estão conforme o especificado no guião de teste, e funcionam correctamente.

Após a fase de teste, seguem-se a instalação e operações de suporte. No decorrer destas fases não são efectuadas quaisquer operações de validação. Este processo só reaparece no ciclo de vida de um sistema aquando da fase de **Manutenção**. A manutenção de software inclui toda e qualquer acção de correcção, aperfeiçoamento e adaptação de uma ou várias componentes do sistema. Quando uma alteração, de qualquer um dos tipos acima mencionados, é feita ao sistema, é necessário assegurar, não só que essa mudança preenche os requisitos definidos, mas também que não interfere com as restantes componentes do sistema. O esforço de validação necessário para que isto aconteça é determinado pelo tipo de mudança, produtos afectados e seu impacto no propósito global do software. Como já tinha sido referido anteriormente neste documento, um conjunto de documentação bem estruturado e completo, não só no que diz respeito à documentação relativa ao sistema propriamente dito, mas também no que concerne à documentação respeitante à validação inicial e, caso existam, anteriores validações no sistema, poderá ser bastante útil nestes casos.

Para além das tarefas *standard* de verificação e validação, há ainda um conjunto de tarefas específicas desta etapa que deverão ser executadas:

- **Revisão do plano de validação.** Nesta fase, o plano anteriormente proposto deverá ser revisto, de modo a que este possa manter-se actual e abranger todas as componentes do sistema. Caso não exista um plano de validação, deverá ser criado.
- **Evolução de anomalias.** É frequente nas organizações serem mantidos registos dos problemas experienciados pelo software em utilização. Nestes registos, normalmente, são descritas as anomalias, possíveis causas e acções correctivas aplicadas. Por vezes, no entanto, essas causas não são descobertas, e, apesar de serem tomadas medidas de correcção, as anomalias podem

reaparecer. É por esta razão que, nesta fase, deverá ser analisada a sua evolução, permitindo assim identificar tendências e adoptar acções correctivas e preventivas.

- **Identificação de problemas.** Como foi referido acima, aquando do processo de validação, todos os problemas encontrados, assim como as suas causas e soluções, deverão ser registados.
- **Avaliação de propostas de mudança.** Todas as propostas relativas a mudanças no sistema deverão ser avaliadas, tendo em conta o efeito que estas terão na globalidade no sistema.
- **Actualização de documentação.** Para concluir o processo de validação nesta etapa, toda a documentação deverá ser cuidadosamente revista, de forma a garantir que todos os documentos sob os quais as alterações tiveram algum tipo de impacto se mantêm actuais.

2.4.2. Técnicas de validação

Como podemos depreender da leitura da secção anterior, o processo de validação de software é composto por um conjunto de actividades e tarefas, que poderão ser executadas uma ou várias vezes ao longo do ciclo de vida de um sistema. Antes de passar à identificação e descrição dessas actividades, faz todo o sentido explicar quais as técnicas dentro das quais elas se enquadram. Segundo Tran (1999), existem então sete técnicas de validação, sendo elas:

Métodos formais. Por métodos formais, entende-se a verificação, ou seja a utilização de técnicas lógicas e matemáticas para investigar e analisar o sistema em questão, não só ao nível da sua concepção e especificações, mas também no que diz respeito a documentação e ao seu comportamento.

Injecção de faltas. Tal como o nome indica, esta técnica consiste na introdução propositada de faltas no sistema, com o propósito de observar o seu comportamento em circunstâncias adversas.

Injecção de faltas de hardware. Esta técnica é uma versão da anterior, com a ligeira diferença de que neste caso as faltas introduzidas no sistema são físicas.

Injecção de faltas de software. Também nesta técnica o sistema deverá ser sujeito, de forma propositada, a faltas. No entanto estas são injectadas na memória do computador, através de técnicas de software, com o propósito de simular as falhas de hardware referidas no parágrafo anterior.

Análise de confiabilidade. Esta técnica é composta por duas etapas: numa primeira fase, é feita uma identificação dos perigos associados à utilização do software em questão. Na segunda fase, são propostas metodologias com o propósito de reduzir a probabilidade disso efectivamente ocorrer.

Análise de perigo. A técnica de análise de perigo envolve a identificação de possíveis ameaças, suas causas e medidas de contingência.

Análise de risco. Esta técnica vai um bocado além da anterior, pois, para cada ameaça anteriormente identificada, identifica quais as possíveis consequências e a probabilidade de estas ocorrerem.

2.4.3. Etapas do processo de validação:

Tal como foi referido anteriormente neste documento, não existe, na comunidade científica, uma opinião unânime acerca de quais as etapas que constituem o processo de validação. De facto, os diversos autores

que se debruçam sobre o tema defendem várias abordagens diferentes. Uma vez que este capítulo pretende fazer um estudo geral acerca da temática da validação, será apresentada, não a abordagem que considero mais correcta, mas antes um resumo das várias etapas defendidas pelos entendidos no assunto, dando assim a conhecer ao leitor os diversos pontos de vista relativos a esta temática. Mais à frente neste documento, será dedicado então um capítulo à definição do modelo de validação que considero mais apropriado para o tipo de sistema em questão.

Uma das etapas que provoca unanimidade entre os autores da área é a compreensão do sistema. Todos os autores consultados até ao momento concordam quem mais importante do que saber como fazer, é saber "*para o que estamos a olhar*" (Schousboe, 2005). Só assim é possível identificar o âmbito, requisitos e abordagens subjacentes ao processo de validação.

Outra abordagem defendida pela grande maioria dos autores consiste na criação de um documento de especificação de requisitos de utilizador (URS). Este documento consistirá numa "*lista de frases de uma ou duas linhas, identificando a funcionalidade que é necessária para o negócio*" (Odegaard, 2006). Segundo Reid e Strause (2007), este documento deve definir:

- O que queremos que o sistema faça;
- O que não queremos que o sistema faça;
- Porque queremos ou não isso;

O propósito deste documento é analisar e definir quais os objectivos que o sistema deve alcançar, do ponto de vista do utilizador (Rodrigues, 2007).

A etapa que se segue consiste na criação de um plano de validação que identifique quem será responsável pela validação, onde e quando é que esta irá ocorrer. É nesta etapa que é definida uma equipa de

validação, que inclui, no mínimo, um responsável funcional, um líder de equipa, um gestor de documentos e um coordenador de testes (Rodrigues *et al.*, 2007).

Neste plano estarão detalhadas as responsabilidades de todos os membros desta equipa. Segundo Odegaard (2006), é neste documento que deve ser apresentada a descrição do sistema, o seu âmbito, especificações, entre outros detalhes considerados importantes para o processo de validação.

Reid e Strause (2007), defendem ainda que deste plano deve constar também a análise de risco, *milestones* definidas para cada etapa, e um conjunto de critérios que permitam medir o sucesso da implementação.

A justificação para a criação deste documento é dada por Schousboe (2005), quando define que não é possível que um indivíduo execute uma actividade e obtenha resultados positivos sem que possua uma meta bem definida.

Apesar de aparentemente simples, esta é uma etapa que não está ainda totalmente clarificada. O plano de validação é requerido pela FDA. No entanto, o nível a que ocorre este planeamento não é uniforme em todas as organizações. De facto, notam-se diferenças nomeadamente quando comparámos empresas dos EUA com empresas pertencentes à UE. Neste último caso, é norma a utilização de uma ferramenta específica, denominada *Validation Master Plan* (VMP), que permite alcançar um nível de controlo específico, e presumivelmente mais adequado. Isto não acontece no primeiro caso, sendo então que uma das controvérsias que se verificam no tema da validação de sistemas computadorizados consiste nesta dualidade de conceitos. Será que estes dois tipos de plano, claramente diferentes, deveriam ter a mesma designação? Qual deles será o mais correcto? Não deveria existir um modelo base, para que existisse consenso? Na opinião da autora, deveria,

de facto, existir um modelo base, que especificasse quais as camadas da organização deveriam ser abrangidas, pois, sendo o processo de validação requerido para determinados sectores empresariais, não faz sentido que o nível de exigência seja o mesmo. Aliás, a mesma é da opinião de que todo o processo deveria estar sujeito a um conjunto de passos específicos predefinidos e iguais para qualquer organização, independentemente da sua actividade, ou local onde se encontrasse.

A fase do processo de validação será mencionada de seguida consiste na criação de um documento no qual estejam descritas as especificações funcionais (*Functional Requirements Specification – FRS*). Este documento deverá ser uma descrição acerca de como deverá funcionar o sistema, de forma a que todos os requisitos de utilizador sejam atingidos (Rodrigues *et al.*, 2007). No entanto, esta documentação não está presente em todas as abordagens. De facto, se alguns dos autores consideram que esta está incluída na etapa anterior, outros, como é o caso de Odegaard (2006), consideram esta etapa como uma extensão do documento de URS, com o propósito de detalhar os requisitos anteriormente definidos.

O mesmo autor, defende como passo seguinte a criação de um Protocolo de Instalação, no qual é definido como é que o software deve ser instalado e quais as características típicas do pacote fornecido pelo vendedor. Este documento pode também incluir especificações de hardware. É nesta fase que é elaborado o documento de especificação de requisitos de concepção (*Design Requirements Specification – DRS*), com o propósito de determinar "o enquadramento técnico necessário à execução das funcionalidades especificadas" (Rodrigues *et al.*, 2007). Tal como no passo anteriormente mencionado, a criação deste documento não é obrigatória, sendo que, se algumas abordagens a consideram essencial, outras não lhe dão a mesma importância.

Odegaard (2006), sugere a criação de um Relatório de Instalação, como sendo um dos passos fulcrais para um processo de validação correctamente efectuado. Isto porque, este documento será *"a prova de que o software foi correctamente instalado, de acordo com as recomendações e requisitos de concepção no vendedor"*.

Uma das abordagens que não gera controvérsia entre os diversos autores, consiste na elaboração de um relatório de testes. Este *"avalia as actividades de teste de um sistema computadorizado face ao respectivo plano de testes e justifica os desvios verificados"* (Rodrigues *et al.*, 2007). Ou seja, é um sumário do que foi feito, resultados obtidos, medidas tomadas e conclusões finais. É extremamente importante, não só no que diz respeito à validação em que esse teste se encontra inserido, mas também para futuros testes e processos de validação. É importante que exista um historial de validações e testes ao sistema.

Segue-se a fase de lançamento do sistema, ou seja, a fase em que é permitido que o software comece a ser utilizado para a produção. Nesta fase, não deverão existir dúvidas acerca da fiabilidade e segurança do sistema. (Odegaard, 2006)

Terminados estes passos, na sua globalidade ou apenas em parte, consoante as preferências da organização em questão, o processo de validação está completo. Mas isso não significa que o trabalho termine aí. O final do processo de validação é apenas o início de uma nova etapa. De facto, a validação de um sistema só termina efectivamente quando este for definitivamente descontinuado, portanto, o sistema deve manter-se validado (Rodrigues *et al.*, 2007). E isto significa que deve ser executado um conjunto de procedimentos standard para gestão e resolução de problemas, controlo da mudança, entre outros. Caso se verifique a necessidade de efectuar alterações ao sistema já validado, estas devem ser revistas e avaliados os potenciais riscos. Toda e qualquer mudança requerida no sistema deve ser devidamente autorizada, documentada,

testada, aprovada, e, só depois disto, implementada. Dadas estas necessidades, equipa de validação deve manter-se e estar envolvida nas tarefas anteriormente mencionadas.

O final do processo é acompanhado de um Relatório de Validação – RV. Este documento deverá descrever todas as actividades realizadas no decorrer deste processo, desvios ao planeamento, soluções encontradas e, para finalizar, a decisão final relativa ao estado do sistema (validado ou não validado), obtida por comparação entre os critérios estabelecidos inicialmente e os resultados finais (Rodrigues *et al.*, 2007).

2.5. A norma ISO/IEC 9126

A norma ISO/IEC 9126, apesar de não se relacionar explicitamente com a validação de software, tem como propósito a garantia da sua qualidade. E é por este motivo que o desenvolvimento do modelo conceptual da validação subjacente a esta dissertação, terá por base a referida norma. É por esta razão que, ao longo dos parágrafos que se seguem, será descrita a norma ISO/IEC 9126, e suas especificidades.

A norma mencionada no título acima teve a sua primeira publicação no ano de 1991, e abrange todas as características que definem um produto de software de qualidade, podendo ser aplicada a qualquer produto de software – para este caso interessa apenas considerar o software dedicado à gestão documental.

O modelo de qualidade definido pelo conjunto de características contidas na norma em estudo divide-se em duas categorias:

- Qualidade interna e externa
- Qualidade em uso

Com o passar do tempo, a norma foi sofrendo alterações e melhorias, e, nos dias que correm, subdivide-se em quatro partes:

- ISO/IEC 9126-1: Modelo de Qualidade;
- ISO/IEC 9126-2: Métricas Externas;
- ISO/IEC 9126-3: Métricas Internas;
- ISO/IEC 9126-4: Métricas de Qualidade em Uso.

As métricas externas, descritas na parte 2 da norma, e tal como o nome indica, dirão respeito à avaliação de características de qualidade externa, características essas que se baseiam nas necessidades do utilizador.

As métricas internas, descritas na parte 3 da referida norma, serão utilizadas para avaliação de características de qualidade interna.

Por último, a parte 4 da norma conterá as características de qualidade em uso.

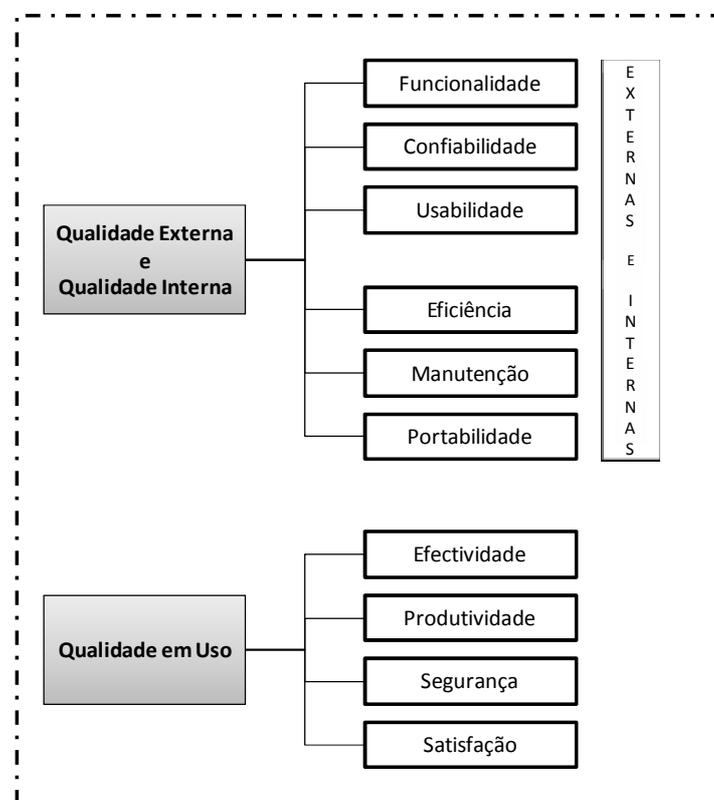


Figura 3: modelos de qualidade

Mas o que é, então, a qualidade interna, externa e em uso?

A **qualidade interna**, segundo Machado e Souza (n.d.) diz respeito à "*totalidade de características do produto de software na visão interna*", ou seja as suas características serão especificadas numa fase intermédia no desenvolvimento do produto de software.

Já a **qualidade externa**, como se torna fácil de perceber, diz respeito às características do produto na visão externa, sendo, portanto, avaliada, quando o produto é já final.

A **qualidade em uso**, por sua vez, diz respeito à medição da qualidade do ponto de vista do utilizador do software em questão, ou seja, para este tipo de qualidade, não interessam as propriedades do software propriamente dito, mas antes em que medida é que o utilizador conseguirá alcançar os seus objectivos através da sua utilização.

2.5.1. Modelos de qualidade

Apesar de possuírem métricas distintas, existe apenas um modelo para a **qualidade externa e interna**. Como pudemos visualizar na imagem acima, este modelo define seis características básicas para que um produto de software possa ser considerado de qualidade, sendo elas:

- Funcionalidade – existência de um conjunto de funções que satisfazem os requisitos definidos para o sistema em questão;
- Confiabilidade – possibilidade de o sistema manter o seu desempenho sob condições estabelecidas, durante um dado período de tempo;
- Usabilidade – capacidade de utilização do software, e medida do esforço necessário para que este possa ser utilizado de forma correcta;

- Eficiência – relação entre o desempenho do produto em questão e a quantidade de recursos utilizados;
- Manutenção – avaliação do esforço necessário para manter o software em funcionamento, bem como para efectuar alterações no mesmo;
- Portabilidade – capacidade de o software ser transportado entre diferentes ambientes.

Cada uma destas características divide-se num conjunto de sub-características, que podemos observar na imagem abaixo.



Figura 4: modelo de qualidade interna e externa

As sub-características apresentadas acima não serão desenvolvidas aqui, mas numa fase posterior deste documento.

O modelo para a **qualidade em uso**, tal como pudemos observar na Figura 3: modelos de qualidade, é constituído por quatro características, sendo elas:

- Efectividade – capacidade de o produto de software permitir ao utilizador efectuar as funções para as quais foi desenhado;
- Produtividade – relação entre a efectividade e os recursos dispendidos;

- Segurança – níveis de danos a pessoas, negócios, *software*, propriedade ou ambiente considerados satisfatórios;
- Satisfação – capacidade de o produto em questão satisfazer, de forma global, os seus utilizadores.

2.5.2. Métricas

Na secção anterior foram apresentadas as características referentes aos vários modelos de qualidade abrangidos pela norma ISO/IEC 9126. Mas estas características de nada valem se não houver forma de as medir, razão pela qual foram estabelecidas métricas.

Duarte e Falbo (n.d.), afirmam que *"para saber o valor de uma determinada característica de qualidade, tem que se criar uma métrica para quantificá-la e fazer uma medição para determinar a medida, que é o resultado da aplicação na métrica"*.

Como vimos já anteriormente, a norma ISO/IEC 9126 define, nas suas diferentes partes três tipos de métricas, para os três tipos de qualidade – interna, externa e qualidade em uso. Todas estas métricas estão relacionadas entre si, pela simples razão de que a qualidade interna é um pressuposto para a qualidade externa, e esta para a qualidade em uso.

As **métricas externas**, descritas na norma ISO/IEC 9126-2, são, como foi referido anteriormente, utilizadas para avaliar o produto final. Para cada uma das sub-características englobadas nas características que vimos na Figura 4: modelo de qualidade interna e externa deverá existir pelo menos uma métrica associada.

As **métricas internas**, por outro lado, têm a sua utilidade na avaliação interna do produto, ou seja tratam directamente com o seu código-fonte e têm um elevado impacto na qualidade externa e de uso.

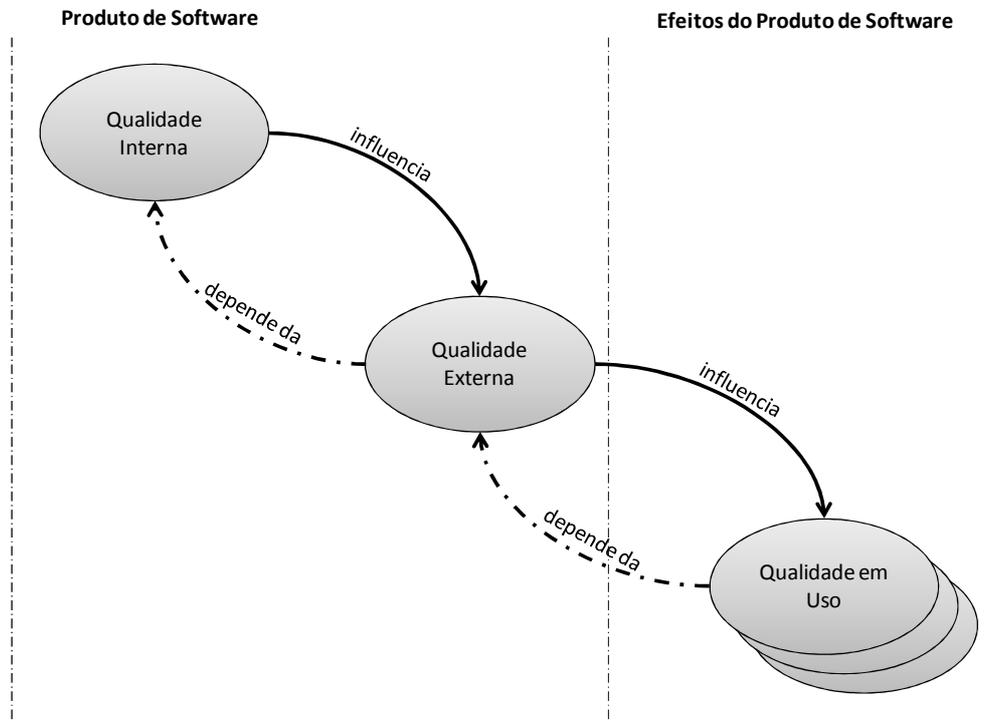


Figura 5: ligação entre os diferentes tipos de qualidade

Por último, as **métricas de qualidade** em uso deverão medir o quanto o produto de software em questão satisfaz as necessidades do utilizador. Estas medidas poderão ser obtidas tanto pela observação de utilização do produto como pela simulação de um ambiente real.

3. Sistemas de Gestão Documental

No capítulo anterior, ficamos a conhecer o conceito de validação de software, e algumas das características que lhe estão associadas. Mas a validação será a mesma coisa quando aplicada a sistemas de gestão documental e quando aplicada a ERP, por exemplo? A resposta é não.

Podemos estudar o processo de validação em termos gerais, mas quando o objectivo é enquadrar um modelo de validação num sistema específico, é necessário fazer algumas adaptações. O facto de o sistema em estudo nesta tese ser um sistema de gestão documental, vai mudar toda a perspectiva de validação, sujeitando-a às suas especificidades. Assim, e antes de prosseguir para a elaboração do referido modelo, faz todo o sentido dedicar o presente capítulo à descrição, sistematização das características associadas aos sistemas de gestão documental e análise de produtos no mercado.

3.1. Documentos: o que são?

Antes de prosseguirmos com a definição de sistemas de gestão documental é necessário compreender as bases em que este conceito assenta. Sendo assim, comecemos pelo essencial: sistemas de gestão documental manuseiam documentos. Mas o que são documentos?

Tradicionalmente, eram considerados documentos os registos textuais (Buckland 1997; Liu 2004). Mas isso era antes de entrarmos na era digital. Chen *et al* (2007) salientam que, nos dias que correm, as tecnologias de informação têm sofrido um crescimento imenso, alargando assim os limites no que concerne à definição de documento.

Em concordância com o que foi dito anteriormente, já Calderon *et al* (2004) haviam definido, anos antes, documentos como sendo o "*conjunto informações registadas em suporte*", mencionando ainda que estas informações podem ter vários tipos de funções: sociais, administrativas, técnicas, jurídicas, culturais, etc., e que, para que estas sejam correctamente executadas, é necessário que sejam preservados e acessíveis.

Também Omar (2005), concorda que os "*métodos de comunicação modernos, tais como e-mail, fóruns de internet, e ficheiros digitais de vídeo e som, aceleraram os processos de negócios, e deram ao termo "documento" uma versão mais nova e expandida*".

A gestão documental é regulada pela norma IEC 82045, sendo que esta abrange não só os documentos em formato papel, como os documentos em formato electrónico. A referida norma, define que os documentos abrangidos por um sistema de gestão documental se poderão dividir em categorias distintas:

- Documentos simples
- Documentos Compostos
- Agregação de documentos.

No caso dos **documentos simples**, e tal como o próprio nome indica, existe um único documento, ao qual é associado um conjunto de metadados específicos, ou seja, um conjunto de dados que o acompanham ao longo do seu ciclo de vida, com o propósito de o identificar e/ou descrever, oferecendo ao documento um valor acrescentado, já que permitem a sua gestão, pesquisa e recuperação mais eficiente dentro de um repositório.

Documentos compostos são constituídos por um conjunto de documentos simples, que, na sua globalidade, formam um único

documento, sendo esse documento associado a um conjunto de metadados.

No caso da **agregação de documentos**, tal como no anterior, existe um conjunto de vários documentos, com a diferença de que estes são todos independentes entre si, sendo cada um associado aos seus próprios metadados. Para além disto, a agregação em si contém os seus próprios metadados, que poderão ou não ter um documento associado.

3.2. Porquê gestão documental?

"A gestão documental tem gerado um grande interesse no mundo dos negócios recentemente. Permite às organizações exercer um maior controlo sobre a produção, armazenamento e distribuição de documentos, produzindo uma maior eficiência na capacidade de reutilização de informações." (Chen et al, 2007)

O mundo dos negócios actual está em constante mutação. As transacções ocorrem a uma velocidade relâmpago, e, nas organizações actuais, ter a capacidade de gerir informações e tomar decisões de forma rápida, eficiente e eficaz, é um factor determinante de sucesso.

Davenport (2000), salienta que a tomada de decisão baseada em informações incorrectas chegam a custar biliões de dólares às organizações, traduzindo-se em investimentos que não se justificam, produtos desnecessários, processos que não funcionam.

Também Furtado (1982) defendia, já há mais de vinte anos atrás, que a eficiência na organização de recursos se traduziria na eficácia de resultados.

O que se verifica ainda em algumas organizações são *"grandes massas documentais acumuladas, sobretudo em suporte papel,*

guardadas sem tratamento adequado” (Calderon *et al*, 2004). Isto vai contra os princípios acima defendidos: como é possível tomar decisões de forma acertada e com a rapidez necessária, quando a informação não está organizada? De facto, o processamento manual de documentos contribui para a sua desorganização – nem sempre o documento se encontra disponível quando requisitado, por vezes informações importantes estão em falta, já para não mencionar os casos em que, com o passar do tempo, os documentos se deterioram, tornando-se a sua informação ilegível. (Omar, 2005)

Os problemas acima mencionados, associados a uma crescente evolução nas tecnologias de informação, levaram a que, nos dias que correm, os documentos em formato electrónico passem a assumir um papel cada vez mais relevante no seio das organizações (Omar, 2005). Mas desengane-se quem pensa que o facto de os documentos passarem a ser manuseados de forma digital implica um menor esforço na sua organização. Assim como acontece com os documentos em formato papel, *“também os digitais precisam de ser arquivados, controlados e circular de forma segura dentro da organização”* (Omar, 2005), motivo pelo qual se impõe a criação e adopção de aplicações informáticas que integrem esses documentos com as áreas funcionais e fluxos de trabalho da organização – os Sistemas de Gestão Documental (SGD)

Antes de prosseguirmos para as definições de sistema de gestão documental, apenas uma ressalva: a designação de sistema de gestão documental não se aplica apenas a sistemas informáticos desenhados para esse propósito. No entanto, e dado que este projecto se enquadra no âmbito das tecnologias de informação, deste momento em diante, o termo Sistema de Gestão Documental fará apenas referência aos sistemas electrónicos para a gestão de documentos.

3.3. SGD – definições

Após clarificação do conceito de documento, e da necessidade que se impõe, para que estes sejam alvo de uma gestão prática e eficaz, passamos então à definição de Sistema de Gestão Documental, analisando as opiniões de alguns nomes relevantes na área:

Omar (2005), define sistema de gestão documental como um *"sistema de controlo e gestão utilizado para regular a criação, utilização e manutenção de documentos criados electronicamente"*.

Já para Lucca (2007), o enfoque da dos sistemas de gestão documental está em *"possuir, de forma electrónica, informações sobre os documentos, independentemente da forma ou suporte em que se encontram"*.

Na opinião de Conarq (2006) um sistema de gestão documental consiste num *"conjunto de tecnologias utilizadas para organização da informação não estruturada de um órgão ou entidade, que pode se dividido nas seguintes funcionalidades: captura, gestão, armazenamento e distribuição."*

Segundo Andrade (2002), *"é ao mesmo tempo um método um sistema e uma tecnologia para a conversão e processamento de documento como informação electrónica digital."*

Conforme pudemos observar acima, não existe uma definição universal para Sistemas de Gestão documental – cada autor define sistema de gestão documental de uma forma única, sendo que alguns dão uma maior relevância à criação e manutenção de documentos, outros valorizam o armazenamento de informação acerca destes, outros ainda as metodologias utilizadas, consoante a sua área de interesse.

Na opinião da autora, definição mais abrangente não é nenhuma das apresentadas acima, mas sim o conceito apresentado pela UNESCO², que declara que a gestão documental é uma parte do processo administrativo, que acompanha o ciclo de vida dos documentos, desde a sua criação até à sua eliminação, procurando sempre que estes sejam utilizados com a maior eficácia possível (Herrera, 1993).

Acrescentaria ainda a esta definição um aspecto para o qual nos alerta Lawrence Burnet³, que consiste no facto de os sistemas de gestão documental terem como finalidade "reduzir selectivamente a proporções manipuláveis a massa de documentos, que é característica da civilização moderna, de forma a conservar permanentemente os que têm um valor cultural futuro sem menosprezar a integridade substantiva da massa documental para efeitos de pesquisa".

De salientar que a definição acima fala na redução dos documentos em formato papel, mas não da sua eliminação. Chen *et al* (2007), alerta-nos para o facto de não ser, de todo, viável eliminar o papel nas organizações. Desta forma, a definição de gestão documental deve abranger não só documentos em formato electrónico, mas também em papel.

3.4. SGD – Definição de uma estratégia

A decisão de implementação de um sistema de gestão documental implica a definição de uma estratégia. Uma organização tem todo um conjunto de factores subjacentes que tornam esta decisão bastante complexa, e obrigam a reflexão. É necessário coordenar políticas

² United Nations Educational Scientific and Cultural Organization

³ Historiador norte-americano

empresariais, fluxos de trabalho, tecnologia, e toda uma variedade de documentos, para poder definir uma estratégia de gestão documental, antes da implementação do sistema informático propriamente dito. Obviamente, não haverá uma estratégia universal que sirva a todas as organizações. No entanto, e segundo Craine (n.d.), existe um conjunto de características associadas ao processo de gestão documental que deverão estar presentes aquando da definição da referida estratégia:

- O processo de definição da uma estratégia documental deverá ser abrangente, cobrindo todas as áreas de interesse da organização em questão. No entanto, é importante não esquecer que deverá também ser possível efectuar a sua gestão. Ou seja, apesar de ser um projecto extenso, será necessário garantir que possui os meios necessários para que nada seja esquecido.
- Outra característica a ter em conta quando se define uma estratégia documental são as metas organizacionais. Estas devem ser claras e bem definidas, assim como a forma de medir resultados e de os comparar com as metas acima mencionadas.
- A cultura organizacional reveste-se também de uma elevada importância aquando da definição da estratégia de gestão documental. Este é um factor muitas vezes menosprezado, mas no entanto muito importante: as políticas internas de cada empresa, bem como a falta de suporte e resistência à mudança por parte daqueles que nelas trabalham são por vezes factores que levam ao insucesso de uma estratégia documental.
- Para terminar, é necessário ter em conta a facilidade de implementação acima mencionada, bem como a avaliação de resultados já referida anteriormente. De facto, a estratégia documental pode ser muito bem concebida, mas, se não

puder ser executada, não terá qualquer valor para a organização! Por outro lado, de que servirá uma estratégia excelente, se não for possível medir os resultados com ela obtidos?

3.5. SGD – Vantagens e Desvantagens

Um sistema de gestão documental poderá conter um número variado de funções, entre as quais registo documental, *workflow*, arquivo digital, entre outros. Esta é também uma solução que apresenta variados benefícios (Foo, 2003), de entre os quais se salientam:

- Eliminação ou diminuição do circuito de papel: a aposta em documentos em formato electrónico é cada vez mais frequente, e apresenta vantagens, entre as quais se destacam a preservação dos documentos com o passar do tempo, a possibilidade de um documento ser consultado em simultâneo por várias pessoas, e até mesmo a preservação ambiental.
- A Redução do espaço físico destinado ao armazenamento de documentos (Lucca, 2007), é outra das vantagens decorrentes da utilização de sistemas de gestão documental. Estando os documentos armazenados no sistema, deixa de ser necessário o espaço físico que seria, caso toda a documentação da organização estivesse contida em ficheiros de papel.
- Gestão automatizada de processos: através da definição de fluxos de trabalho, é possível uma melhor gestão dos processos dentro da organização, conduzindo a um aumento de produtividade (Lucca, 2007);
- Geração automatizada de documentos: consoante o sistema em questão, esta é uma funcionalidade que poderá ser

implementada, traduzindo-se numa vantagem para a organização.

- Possibilidade de acesso simultâneo a documentos por parte de vários utilizadores do sistema, devido à integração com a Web (Lucca, 2007) – esta é uma dupla vantagem uma vez que, ao disponibilizar os documentos online diminui também os custos com cópias, já para não falar na preservação ambiental, uma vez que diminui a utilização de papel nas organizações.
- Integração com outros sistemas, como é o caso do e-mail, fax, entre outros.
- Possibilidade de realizar pesquisas mais rápidas e eficientes, já que os documentos estão armazenados electronicamente num repositório global;
- Auxílio no processo de tomada de decisão (Lucca, 2007);
- Minimização de perdas, já que os documentos são armazenados em arquivo digital;
- Possibilidade de indexação, ou seja, catalogação e categorização dos documentos electrónicos. Esta fase é em tudo equivalente ao processo de arquivo físico mas com todos benefícios dos sistemas de informação.

"A questão da aplicação da gestão de documentos envolve paralelamente a máxima utilização da informação e a mínima utilização de tempo, pessoal e dinheiro, assim, garantindo a eficiência no âmbito interno e externo às empresas." (Cláudio, 2005)

Claro que nem tudo são benefícios, e, como todos os sistemas informáticos, este traz também as suas complicações. As questões relativas à segurança são uma delas. É impossível obter um sistema 100% seguro, a tecnologia está em constante evolução, e com a ela as formas de quebrar barreiras de segurança. Aliás, não é preciso ir tão longe para verificar uma desvantagem dos sistemas electrónicos de

gestão documental: numa empresa em que a documentação circule, maioritariamente, por via electrónica, caso haja um corte na electricidade não há condições para prosseguir o seu funcionamento normal. Isto não aconteceria caso o trabalho fosse efectuado em papel.

Obviamente que todas as decisões têm contrapartidas, e não há uma solução infalível. Apesar disto, na opinião da autora, a utilização de sistemas de gestão documental tem uma grande utilidade, e é e será cada vez mais comum no contexto organizacional.

3.6. Características dos SGD

Após a explicação sobre o que são os Sistemas de gestão documental, ficamos a conhecer qual o seu propósito e quais os tipos de documentos que manuseiam. É tempo, então, de avançarmos para o domínio técnico. Neste capítulo, serão apresentadas as funcionalidades que um sistema deste tipo poderá assumir, e de que forma poderá a sua validação um problema. Antes de prosseguir é, no entanto, importante, referir que estas funcionalidades variam consoante o sistema em questão, podendo, para alguns casos, não se verificar.

3.6.1. Importância do ciclo de vida dos documentos

Todas as funcionalidades de um sistema de gestão documental têm por detrás um conceito essencial – o ciclo de vida de um documento. De facto, este é o “esqueleto” de um sistema de gestão documental, já que é nele que assentam todas as funcionalidades do mesmo.

Segundo a norma IEC 82045, o ciclo de vida de qualquer documento deverá ser composto por sete fases, sendo elas a iniciação, preparação, estabelecimento, utilização, revisão, anulação e supressão.

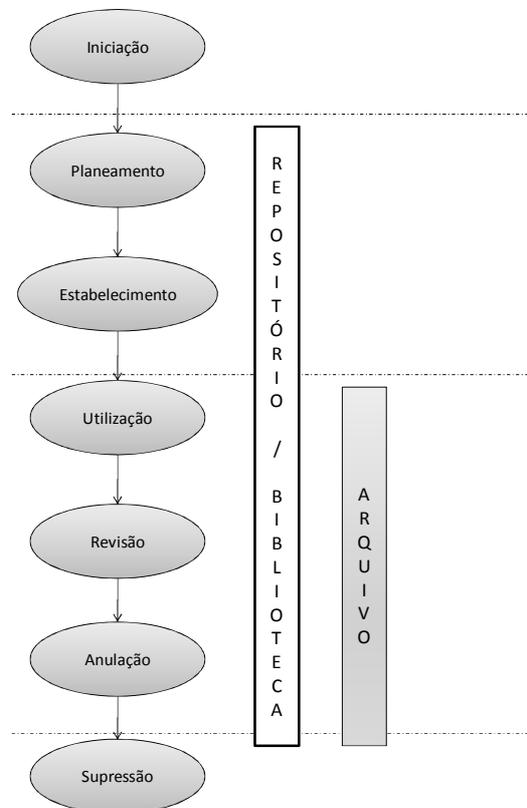


Figura 6: ciclo de vida de um documento

A **iniciação** consiste na entrada do documento no sistema. É nesta fase que ele deverá ser identificado, de forma inequívoca e estável dentro do sistema, ou seja, a identificação deverá ser única, e não depender de quaisquer factores (como por exemplo a localização física do documento).

A fase que se segue, tal como o nome indica, consiste na **preparação** do documento, ou seja, o desenvolvimento do seu conteúdo.

O **estabelecimento** de um dado documento poderá variar de acordo com os procedimentos de controlo de documentos definidos para o sistema em questão. Por norma, o documento deverá ser submetido a verificação, revisão, correcção e aprovação, junto das entidades

responsáveis. Após concluídas estas etapas, o documento será publicado, concluindo assim a fase de estabelecimento.

Após o estabelecimento, o documento estará apto a ser utilizado para os fins pretendidos, dando assim início à fase de **utilização**, durante a qual os documentos estarão disponíveis a quem de direito. É importante ressaltar que, no decorrer desta fase, os documentos em questão deverão estar acessíveis através de um repositório seguro e controlado.

Com o passar do tempo, poderá ser necessário **rever** os documentos criados. Uma das funcionalidades básicas oferecidas por um sistema de gestão documental é a criação de versões de um mesmo documento. A norma IEC 82045 define que, ao longo do ciclo de vida de um dado documento, este poderá sofrer alterações. Estas alterações poderão enquadrar-se em dois tipos:

- Alterações ao seu conteúdo;
- Alterações ao seu aspecto.

Ainda que as segundas não exijam, por norma, uma nova versão, por cada alteração efectuada ao conteúdo de um documento deverá ser publicada uma nova versão do mesmo. A mesma norma define ainda a possibilidade de existência de ligações entre diferentes documentos aquando do período de preparação de uma nova versão. No entanto, após passagem para a fase de aprovação ou publicação, deverá deixar de ser permitido efectuar ligações deste tipo, uma vez que isso poderia conduzir a alterações de conteúdo.

Havendo documentos que permanecem no sistema por longos períodos, outros há que, após algum tempo, perdem a sua utilidade. Dá-se então início ao processo de **anulação** do mesmo. Por questões legais, o mesmo documento deverá permanecer em arquivo durante um período

mínimo de tempo⁴, antes de se proceder à sua **supressão** – momento no qual o documento é, definitivamente, eliminado do sistema, deixando de ser rastreável.

A validação do sistema de gestão documental, na vertente que diz respeito ao ciclo de vida de documentos, deverá garantir que os seguintes aspectos estão adequados ao especificado na URS:

- Correção no formato de entrada do documento: os dados de entrada no sistema deverão estar de acordo com o especificado na especificação de requisitos. Ou seja, deverão ser efectuados testes para verificar se o sistema aceita os dados especificados, e recusa quaisquer outros não previstos na sua documentação;
- Validade da identificação do mesmo: Cada documento deverá ser identificado inequivocamente. Mas será que isso, de facto acontece? Uma falha neste aspecto traduzir-se-ia em danos elevadíssimos, e portanto este é um aspecto a ter em conta. Será necessário então testar se o sistema atribui uma identificação única a cada documento, e se a mesma é fiável;
- Garantia de que o documento é introduzido/modificado/eliminado por utilizadores com permissões para tal: outro aspecto que é necessário validar são as permissões de utilizadores. Será necessário, para isso, testar dois cenários possíveis: se um utilizador com permissões para executar um determinado processo o consegue, de facto, fazer, e se um utilizador que não tenha essas permissões, estará realmente impedido de o fazer. Para isto, vários casos de teste deverão ser executados, nomeadamente para testar este aspecto em actividades de

⁴ Em Portugal, este período é, actualmente de 10 anos.

introdução, modificação e supressão de documentos, em diversos níveis de segurança.

- Garantia da confidencialidade dos documentos: directamente relacionado com o ponto acima, este aborda a questão da confidencialidade e privacidade, ou seja, para que este aspecto seja efectivamente validado, será necessário responder a questões como: estará garantida a confidencialidade dos documentos? Poderá um utilizador sem permissões para tal consultar um dado documento? Quais os mecanismos de segurança que garantem esta confidencialidade?

3.6.2. Respeita os fluxos de trabalho

Como pudemos observar acima, os documentos manuseados por uma organização não estão parados. Eles são "*criados, modificados, distribuídos, em rotas claramente definidas*" razão pela qual os sistemas de gestão documental devem conter "*fluxos de trabalho que estabeleçam precisamente para onde será enviado um documento, se ele chegou ao seu destino, quando é que foi redireccionado e qual o seu estado num determinado momento*" (Mehedintu et al, 2007),

Da afirmação acima, depreende-se então que, para além do ciclo de vida do documento, o sistema deverá também ter fluxos de trabalho claramente definidos, de modo a que seja possível definir a sequência de tarefas a serem executadas pelo mesmo.

Sendo os fluxos de trabalho uma das características fundamentais dos sistemas de gestão documental, é crucial que a sua validação não só seja executada, como também o seja de uma forma que abranja todos os aspectos necessários. Assim, o principal desafio inerente à validação dos fluxos de trabalho de um sistema de gestão documental consiste em

garantir que todos os passos desse mesmo fluxo são abrangidos pelo sistema, e executados correctamente, de acordo com o especificado. Para tal será necessário testar um conjunto de cenários, que abranjam os seguintes aspectos:

- Criação de documentos;
- Modificação de documentos;
- Circulação de documentos.

Será, portanto, necessário validar se estes processos – criação, modificação e circulação – ocorrem de acordo com o previsto. Esta validação deverá incluir cenários de teste em que existam vários destes processos a decorrer em simultâneo, com vista a observar o comportamento do sistema nesta situação. Outra questão importante que deverá ser estudada é a execução do mesmo processo por diferentes actores de um mesmo sistema, com diferentes permissões. Para além disto, será necessário testar se o fluxo decorre conforme o esperado ou se existe a possibilidade de avançar etapas.

3.6.3. Tem por base um repositório

À definição do ciclo de vida dos documentos e fluxo de trabalho que deverá ser seguido pelo sistema, junta-se outra funcionalidade essencial para o correcto funcionamento de um sistema deste tipo: a existência de uma biblioteca de documentos. De facto, não faz sentido pensar num sistema de gestão documental sem pensar no seu elemento central – a biblioteca/ repositório. Aqui serão armazenados todos os documentos, durante as várias fases do seu ciclo de vida.

A validação desta característica deverá abordar aspectos como:

- Segurança

- Conteúdos
- Acessos

Ao nível da segurança, deverá ser garantido que os documentos no repositório estarão disponíveis apenas a quem tenha permissões para tal. Deverá também ser garantido o registo de operações, e verificada a sua fiabilidade. Outra questão a ser observada são os meios utilizados para garantir esta segurança, e sua adequação com o definido na especificação de requisitos.

Quanto aos conteúdos, a validação desta componente do sistema deverá garantir que os conteúdos permitidos pela mesma correspondem ao definido na especificação de requisitos, nomeadamente no que diz respeito ao seu formato, tamanho e organização.

Outro aspecto sobre o qual deverá incidir a validação do repositório de um sistema de gestão documental são os acessos – esta questão está mais uma vez relacionada com a segurança, e o desafio que se coloca aqui é a garantia de que os acessos ao repositório, tanto na inserção, como na consulta ou eliminação de documentos, são correctamente controlados. Também a questão dos acessos concorrenciais se reveste de uma elevada importância, sendo necessário garantir que o sistema suporta acessos simultâneos de diferentes utilizadores, com diferentes permissões, continuando a assegurar a segurança do repositório.

3.6.4. Implementa funcionalidades genéricas

Portanto, até ao momento, foi possível concluir que um sistema de gestão documental assenta em três pilares:

- O ciclo de vida dos documentos,
- O fluxo de trabalho que estes deverão seguir;

- O seu armazenamento num repositório.

Mas, em termos práticos, quais serão as funcionalidades que um sistema de gestão documental deverá disponibilizar aos seus utilizadores? Os sistemas de gestão documental, com maiores ou menores alterações, assentam sempre num conjunto de informações genéricas. De facto, ao contrário de outros tipos de sistemas informáticos, que disponibilizam uma panóplia de opções e funcionalidades, os sistemas de gestão documental servem essencialmente para uma coisa: gerir documentos. E, por esta razão, existe um conjunto de funcionalidades genéricas que lhes são características. Herrera (1993), salienta as seguintes:

- Normalização de documentos, de forma a simplificar o processo administrativo. A validação desta característica terá como desafio garantir que a normalização é, de facto, efectuada de forma adequada. Ou seja, que o sistema identifica os tipos de documentos manuseados pela organização, e possui mecanismos para efectuar correctamente esta normalização.
- Informatização dos processos de trânsito documental, permitindo assim, não só a redução do fluxo de papel na organização, mas também uma maior agilidade e segurança. Uma das vantagens fundamentais provenientes da utilização de sistemas de gestão documental advém da redução do uso de papel, e conseqüente agilização dos processos. Sendo o tema desta tese a validação de software dedicado à gestão documental, apenas fará sentido validar a questão da agilidade resultante da informatização de processos. Assim, será necessário garantir que a circulação de documentos no sistema corresponde ao esperado, assegurando que são cumpridos os fluxos de trabalho (explicados na secção anterior) e tempos definidos na especificação de requisitos.

Omar (2005), alerta para a importância de não esquecer que os sistemas de gestão documental não trabalham apenas com documentos em formato electrónico, acrescentando uma funcionalidade que considera essencial:

- *Scanning*, como forma de conversão de documentos em formato papel para o formato electrónico. Em termos de validação, o *scanning* é visto como entrada de dados – ou documentos, no caso – no sistema. Como tal, será necessário assegurar que estes dados se encontram no formato correcto, são armazenados conforme o esperado, lhes são aplicadas as permissões e restrições correctas, e respeitam as normas de segurança impostas na especificação de requisitos.

Para além da característica acima mencionada, Herrera (1993), defende ainda que um sistema de gestão documental completo deverá abranger a funcionalidade de indexação. Segundo a autora, uma das funcionalidades essenciais a um bom sistema de gestão documental é a implementação de um algoritmo de indexação, que torne a procura e utilização de um dado documento simples e eficaz. A questão da validação da indexação prende-se com a garantia de que esta é feita correctamente, e que existem meios de aceder, posteriormente, a estes dados.

3.6.5. Podem ser sistemas colaborativos

Uma das vantagens dos sistemas de gestão documental é, segundo Herrera (1993), a possibilidade de uma maior coordenação e colaboração dentro da organização. Ou seja, sistemas de gestão documental são sistemas colaborativos. No entanto esta colaboração pode dar-se em níveis distintos, consoante o tipo de organização em que estão implementados. Pequenas organizações poderão utilizar sistemas de

gestão documental que interliguem apenas dois departamentos. Ou, num caso extremo, uma pequena biblioteca poderá utilizar um sistema contendo apenas um pequeno repositório – não deixa de ser um sistema de gestão documental, no entanto a necessidade de colaboração não se verifica. No outro extremo, existem sistemas de gestão documental que interligam toda uma organização, podendo até existir necessidades de interoperabilidade com outros sistemas.

A validação desta característica é complexa, uma vez que deverá abranger vários aspectos:

- Questões relacionadas com a segurança
- Fluxos de trabalho
- Mecanismos de comunicação
- Integração com outros sistemas

As questões relativas à validação da segurança e fluxo de trabalho são já abordadas neste documento, nas secções apropriadas. Quanto à questão dos mecanismos de comunicação, torna-se necessário garantir que todas as informações chegam a quem delas necessite, de forma correcta e em tempo útil. Este será um dos problemas inerente à validação da funcionalidade de colaboração dos sistemas de gestão documental. Para além desta questão, será necessário avaliar o comportamento do sistema em situações de colaboração de vários utilizadores, ou seja, estando um conjunto de actores a utilizar simultaneamente uma determinada componente do sistema, como reagirá este? Que desafios advirão deste acesso simultâneo?

A validação da integração com outros sistemas revela-se um desafio, uma vez que há um conjunto de situações a ter em atenção: é necessário validar se as trocas de dados estão a decorrer conforme o esperado, respeitando formatos, tempos de execução e consistência dos mesmos, controlos de acesso, assim como se a cooperação entre os sistemas em questão cumpre os requisitos especificados.

3.6.6. Privilegiam a segurança

Para terminar, resta ainda referir uma das características mais valiosas dos sistemas de gestão documental – a segurança. Herrera (1993) defende a importância do controlo de acesso aos documentos presentes no sistema, como forma essencial para garantir a segurança do mesmo. Assim, cada documento deverá apenas estar disponível para quem dele realmente necessite, e o sistema deverá fornecer meios de autenticação e comprovação de identidade seguros e eficazes. De facto, não se justificaria que todos os documentos estivessem disponíveis para todos os utilizadores, e que estes pudessem efectuar alterações nos mesmos. Deverá então existir um conjunto de permissões de visualização, modificação, edição, eliminação, e quaisquer outros tipos que se revelem necessários dado o contexto no qual o sistema é utilizado.

Em termos de validação, esta será talvez a característica mais problemática, por uma razão muito simples: não há forma de garantir a 100% a segurança de qualquer sistema informático. Isto deve-se, em primeiro lugar, ao facto de ser uma área em constante evolução, e que de um momento para o outro se torna obsoleta. Mas a principal razão para a insegurança dos sistemas informáticos em geral, e dos sistemas de gestão documental em particular, é bem mais simples do que isto: são sistemas utilizados por humanos. Numerosas estatísticas provam que a principal causa de falhas em sistemas informáticos é erro humano. Porque não importa o quão avançado é o sistema, em última instância, quem detém o poder é o Homem, que é bastante menos fiável do que a máquina.

Portanto, é sabido que os resultados da validação ao nível da segurança de um sistema de gestão documental não serão 100% fiáveis. Mas isso não significa que não possa ser validado. Para validar a

segurança destes sistemas, será necessário analisar três aspectos distintos:

- Confidencialidade
- Integridade
- Disponibilidade

Para validar a **confidencialidade** de um sistema de gestão documental, será necessário avaliar, antes de mais, as entradas no sistema – é necessário compreender se o processo de *login* ocorre conforme esperado. De seguida, torna-se pertinente avaliar as definições de acessibilidade, ou seja, um dos problemas que se põe quando pretendemos avaliar a confidencialidade de um sistema é verificar que utilizadores têm acesso a que áreas do sistema e a que tipos de dados. Não faria sentido que, por exemplo, numa organização, um funcionário de escritório tivesse permissões para editar documentos elaborados pelos gestores de topo. A validação deverá garantir que todos os cenários são testados, e que existem no sistema os mecanismos apropriados para garantir a sua confidencialidade.

A questão da **integridade** prende-se mais com os dados propriamente ditos, no caso dos sistemas de gestão documental os documentos. É necessário validar a correcção dos mesmos, ou seja, garantir que as informações e documentos transmitidos pelo sistema estão correctos, e de acordo com o esperado.

Para terminar, resta falar da **disponibilidade**. A validação da segurança ao nível da disponibilidade relaciona-se directamente com a questão da confidencialidade, uma vez que visa garantir que os documentos estão disponíveis, quando necessários, para as pessoas com permissões para tal.

Para concluir, deverá ser referido que, após estudo na área, a autora do presente documento partilha da opinião de Dourish *et al.*

(2000), quando estes definem a existência quatro funcionalidades fundamentais de qualquer sistema de gestão documental, dentro das quais se as tarefas que foram apresentadas anteriormente. Essas funcionalidades são as seguintes:

- Preenchimento de documentos, que poderá ser realizado pelos utilizadores com permissão para tal, ou de forma automática, pelo sistema.
- Gestão de documentos. Esta funcionalidade abrange a definição de ciclos de vida, revisão e aprovação de documentos, entre outras.
- Localização: possibilidade de, através da indexação acima referida, localizar com facilidade qualquer documento.
- Partilha: esta é, na opinião da autora, se não a mais importante, uma das mais importantes funcionalidades destes sistemas. Isto porque permite que vários utilizadores acedam simultaneamente a um mesmo recurso, mas geralmente possui mecanismos que evitam que num mesmo documento sejam efectuadas alterações em simultâneo, combatendo assim a redundância e garantindo que o documento é consistente.

3.7. Validação de SGD

Nas secções anteriores, o presente documento debruça-se já sobre a temática da validação e respectivas características. Neste capítulo, foi introduzido o conceito de gestão documental e suas especificidades, bem como aqueles que serão, na opinião da autora, os possíveis problemas de validação associados a cada uma destas características. Mas o que é, para a comunidade científica, a validação de sistemas de gestão documental?

Conforme é possível depreender pela abordagem feita na secção anterior, a validação de um sistema de gestão documental, é então a aplicação das etapas mencionadas no capítulo anterior a um sistema de gestão documental. Isto poderá parecer simples, no entanto não é assim tão linear.

No decorrer das pesquisas efectuadas, não foi possível encontrar uma única descrição geral de validação destes sistemas, apenas casos muito específicos, dentro de contextos muito específicos. Isto deve-se, no ponto de vista da autora, a dois factores:

- Por um lado, a validação de sistemas de gestão documental não é um procedimento comum. De facto, apenas as organizações que se enquadram nas áreas da saúde e alimentação se vêm obrigadas a executar tal procedimento, sendo que uma grande parte das restantes não possui informação suficiente e relevante sobre o processo.
- Por outro lado, as organizações que dedicam tempo e esforço ao desenvolvimento do mesmo processo, para além de o fazerem num contexto único e específico, não sendo, portanto, um modelo geral, não têm qualquer interesse em partilhar o conhecimento fruto do seu trabalho com os seus concorrentes directos.

Impõe-se então questionar: será pertinente aplicar o processo de validação a sistemas de gestão documental? Porquê? E de que forma?

Para responder a estas questões é necessário, antes de mais, identificar a perspectiva sobre a qual ocorrerá este processo. De facto, é possível identificar dois pontos de vista distintos: o de quem desenvolve o sistema e o de quem o adquire e implementa num contexto organizacional específico. O processo de validação será distinto para os dois casos acima mencionados. Como pudemos observar no capítulo anterior, a validação de software poderá ocorrer em várias fases distintas

do seu ciclo de vida. Se relacionarmos a informação adquirida no capítulo do presente documento dedicado à validação com as duas situações acima descritas, poderemos concluir que, quando falamos da validação do ponto de vista do fornecedor/criador do produto, estamos-nos a referir à validação que poderá ocorrer em qualquer uma das etapas do ciclo de vida do produto até ao teste, e quando nos referimos à validação do ponto de vista do comprador, da organização que implementa o sistema, estamos a referir-nos ao processo que ocorre aquando da manutenção do mesmo.

Para ambos os casos, a resposta à questão relativa à pertinência deste processo é afirmativa. Um sistema de gestão documental deverá abranger todas as áreas funcionais de uma organização. Como tal, haverá interesse, não só da parte da mesma como também da parte de quem o fornece, em garantir que este funciona de acordo com os requisitos previamente especificados. É por esta razão que a validação se poderá traduzir numa mais-valia para empresas que utilizem este tipo de sistemas. A forma como essa validação deverá ser feita não é, no entanto, clara. De facto, e tal como foi mencionado acima, não é possível recolher informação abstracta acerca da validação de sistemas de gestão documental.

É, então, neste contexto, que se enquadra o projecto em questão. De facto, esta falha na informação é a principal área de interesse no que concerne ao tema deste projecto, sendo o facto de não existir ainda uma resposta genérica para a problemática em torno da validação de sistemas de gestão documental, o que motiva a execução do referido projecto.

O capítulo que se segue será então dedicado ao estudo das normas actualmente em vigor dentro da área da gestão documental, e elaboração de um modelo conceptual de validação, adaptado aos sistemas de gestão documental.

3.8. SGD - Conclusões

A gestão documental assume, de facto, uma elevada importância, não só no seio das organizações, mas também no dia-a-dia. Basta pensarmos que, se hoje em dia temos conhecimento de alguns factos históricos, foi porque houve documentação que foi preservada com o passar dos anos. É, portanto, importante preservar os documentos, e todos sabemos que, quando se lida com grandes volumes de informação, a tendência é descurar a sua manutenção. Já para não mencionar o facto de se tornar praticamente impossível encontrar a informação pretendida num período de tempo viável, caso esta esteja desorganizada.

Cada vez mais, a gestão documental se assume como uma vantagem competitiva para as organizações que nela investem, permitindo, para além da questão da manutenção referida acima, bem como da diminuição do volume de papel na organização e subsequente redução do espaço físico necessário, também uma optimização dos fluxos de trabalho, e ganhos em tempo e eficiência, já mencionados no capítulo relativo às vantagens.

Assim, a gestão documental surge como uma necessidade fundamental nos dias que correm. É, de facto, muito importante investir nesta área, devido às inúmeras vantagens que foram mencionadas ao longo deste documento. É preciso, no entanto, não esquecer que a implementação de um sistema de gestão documental não traz só vantagens – ela tem contrapartidas que devem ser tidas em atenção. A questão da segurança, já mencionada anteriormente, é uma das mais relevantes. É preciso não esquecer também os avanços tecnológicos a que esta área está sujeita, tanto pela positiva como pela negativa.

No que diz respeito à validação deste tipo de sistemas, é importante salientar a escassez de informação existente. Como foi referido anteriormente, a pouca informação existente sobre esta temática diz

respeito a situações muito específicas, não se adaptando a outros sistemas, ou até ao mesmo sistema num contexto ligeiramente diferente. Este facto surge então como motivação para o desenvolvimento de um modelo conceptual de validação, adaptável aos sistemas de gestão documental, de forma global, assunto que será aprofundado no capítulo que se segue.

4. Modelo conceptual de validação

Como foi descrito nos capítulos anteriores, o propósito deste projecto consiste na elaboração de um modelo conceptual de validação de sistemas de gestão documental. Para isso, é necessário garantir que o mesmo respeita os requisitos normativos referentes à sua qualidade. Para avaliar este aspecto, o modelo a desenvolver terá então por base a norma ISO/IEC 9126, descrita no capítulo 2 do presente documento.

4.1. Definição de um modelo conceptual de validação

Conforme pudemos observar em capítulos anteriores deste documento, a validação poderá ocorrer em várias fases do ciclo de vida de um produto de software. Uma vez que este modelo em particular se baseará na norma ISO/IEC 9126, e tendo em conta os tipos de qualidade nela definida, o modelo em questão vai então definir três momentos para a execução deste processo, podendo estes ser executados em separado ou na sua globalidade. Assim, a validação poderá ocorrer aquando do desenvolvimento do produto, situação na qual se aplicariam métricas de qualidade interna; ou na fase final do seu desenvolvimento, situação na qual as métricas indicadas seriam as correspondentes à qualidade externa; ou ainda durante a sua utilização e manutenção, situação na qual as métricas mais adequadas seriam as de qualidade em uso.

Poderiam, portanto, ser definidos três modelos de validação distintos, consoante esta ocorresse em qualquer uma das etapas anteriormente mencionadas. No entanto, e dada a interligação entre

estas, e o facto de cada uma delas estar dependente do sucesso da anterior, como pudemos observar na Figura 5: ligação entre , fará mais sentido a criação de um modelo global de validação.

Assim sendo, e com o propósito de não descurar qualquer uma das etapas anteriormente mencionadas, será elaborado um modelo de validação que abranja todas elas de forma sequencial, seguindo a estrutura abaixo apresentada.

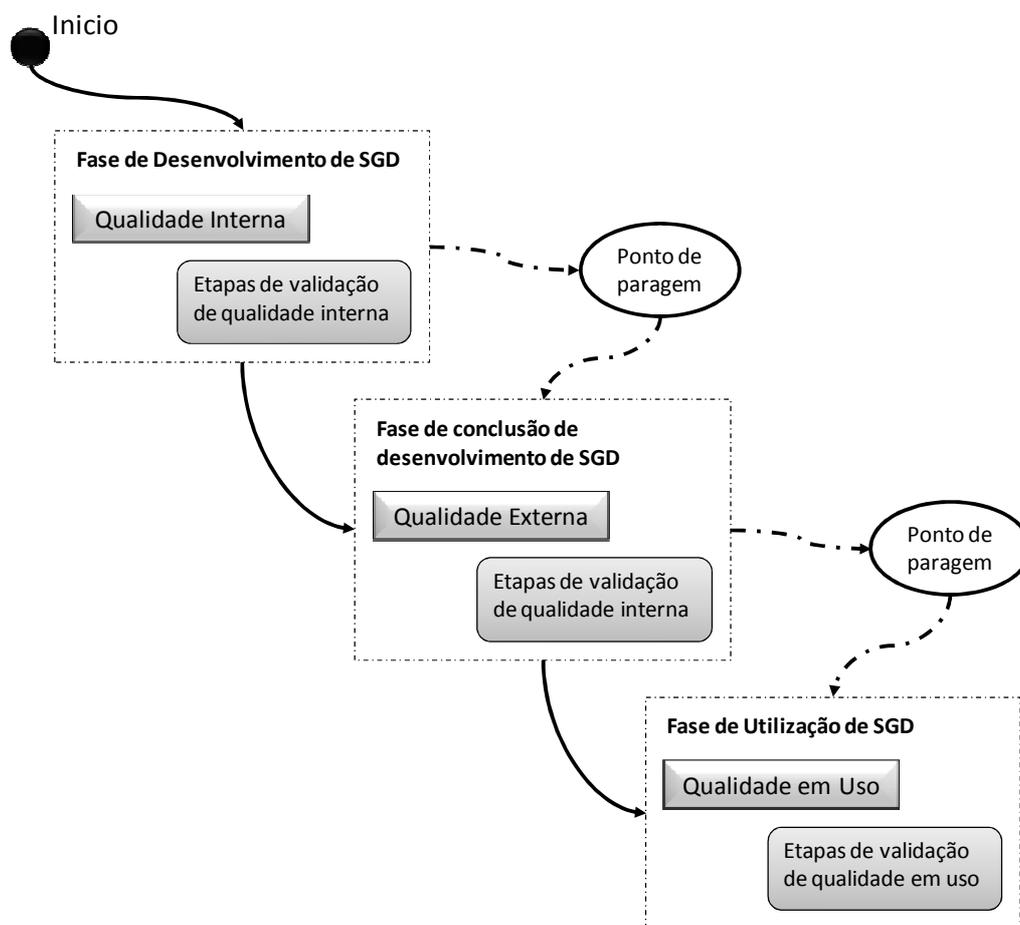


Figura 7: estrutura do modelo conceptual de validação

Para uma melhor compreensão do trabalho desenvolvido, este será apresentado primeiro por módulos, sendo dedicada uma secção a cada uma das três fases presentes no modelo global, e só depois na sua globalidade.

Antes de prosseguirmos para o modelo propriamente dito, apenas uma ressalva: no capítulo 2 deste documento, estudamos as diferentes técnicas de validação. Este modelo em particular terá por base a aplicação de métodos formais, ou seja, a aplicação de diferentes métricas que, através de fórmulas matemáticas, permitem avaliar o desempenho do sistema, e assim validar se corresponde ao expectável.

4.2. Validação durante a fase de desenvolvimento de SGD

A validação durante a fase de desenvolvimento de um sistema de gestão documental ocorre aquando da sua construção/codificação. Recordemos então a imagem que vimos anteriormente neste documento, ilustrativa do ciclo de vida do software. Analisando a imagem abaixo, concluiremos que, antes de procedermos a esta primeira etapa de validação, deverão ter sido executadas as seguintes fases:

- Planeamento;
- Definição de requisitos de sistema;
- Especificação de requisitos de software;
- Especificação dos requisitos de concepção do software.

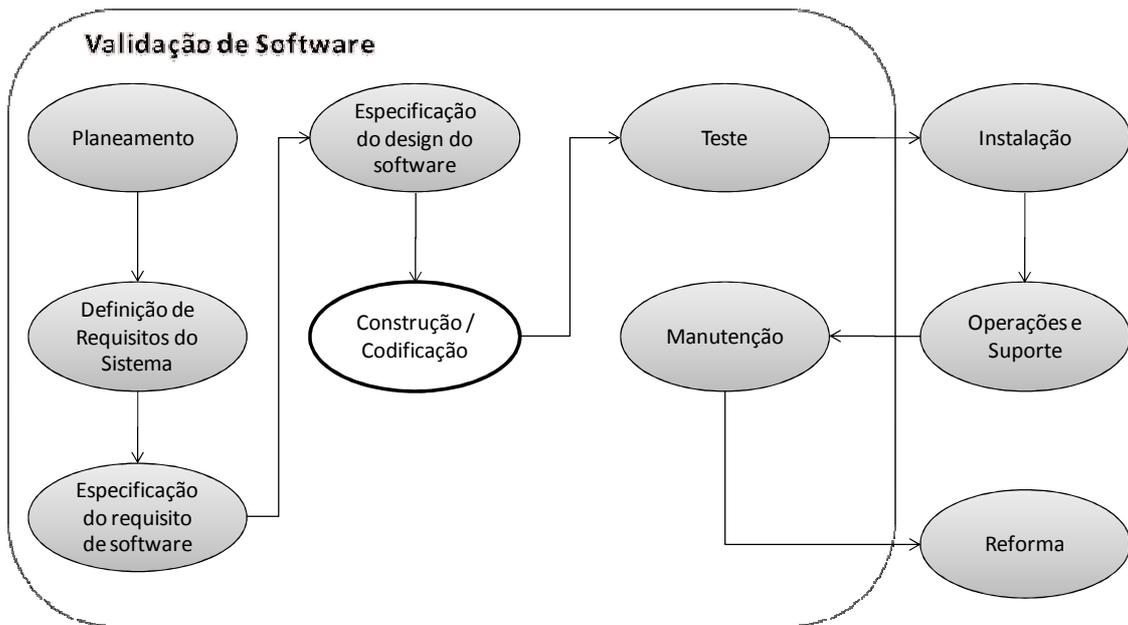


Figura 8: validação interna no ciclo de vida do software

Para cada uma destas fases deverá ter sido produzido um conjunto de *outputs*, neste caso documentos, como poderemos observar no esquema que se segue.

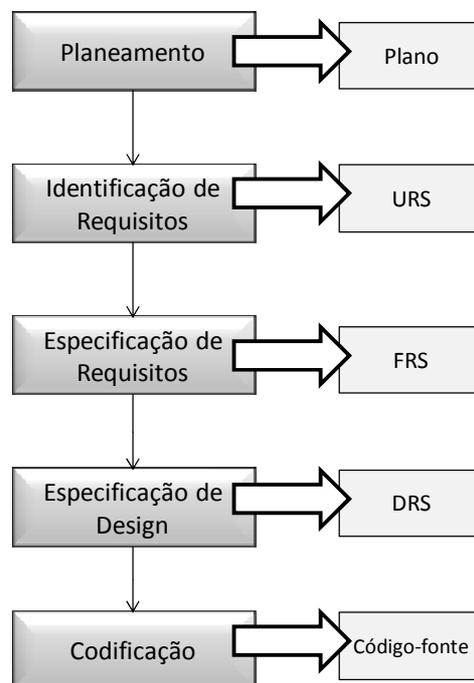


Figura 9: *outputs* validação interna

Da fase de **planeamento** deverá surgir o documento contendo o plano que abrangerá todo o processo de desenvolvimento do software.

Informações tais como objectivos a alcançar, abordagens a seguir, recursos disponíveis, entre outros, deverão constar deste documento. A compreensão do sistema é, como pudemos observar no capítulo relativo à validação, uma componente essencial para o sucesso de todo o processo de validação.

A fase de **identificação de requisitos** tem como *output* o documento de *User Requirements Specification* – URS, que consistirá na enumeração e descrição dos requisitos que se espera que sejam cumpridos pelo sistema, do ponto de vista do utilizador. Já a fase de **especificação de requisitos** produzirá um documento que abordará esses mesmos requisitos, de um ponto de vista funcional – *Functional Requirements Specification* – FRS, descrevendo de que forma deverá o sistema funcionar para que seja possível alcançar os requisitos definidos na URS. Ainda dentro da temática dos requisitos, existe ainda um documento de elevada importância, que é produzido na fase de **especificação de requisitos de concepção** – *Design Requirements Specification*.

Por ultimo, temos a etapa da codificação, que, tal como o próprio nome indica, deverá produzir o código-fonte relativo ao produto em questão.

Portanto, chegados ao momento de validação, temos disponível um conjunto de *outputs* que é necessário validar. Mas como é que pode ser feita essa validação? A norma ISO/IEC 9126, que está na base deste modelo, define um conjunto de métricas enquadradas nas diferentes características do software, com vista à obtenção de dados que permitam concluir quanto à validade ou não do mesmo. Desse conjunto, serão seleccionadas as que se adaptam ao contexto específico dos sistemas de gestão documental, sendo depois aplicadas. O diagrama que se segue demonstra então, em linhas gerais, o procedimento a realizar para efectuar a validação interna de um sistema de gestão documental.

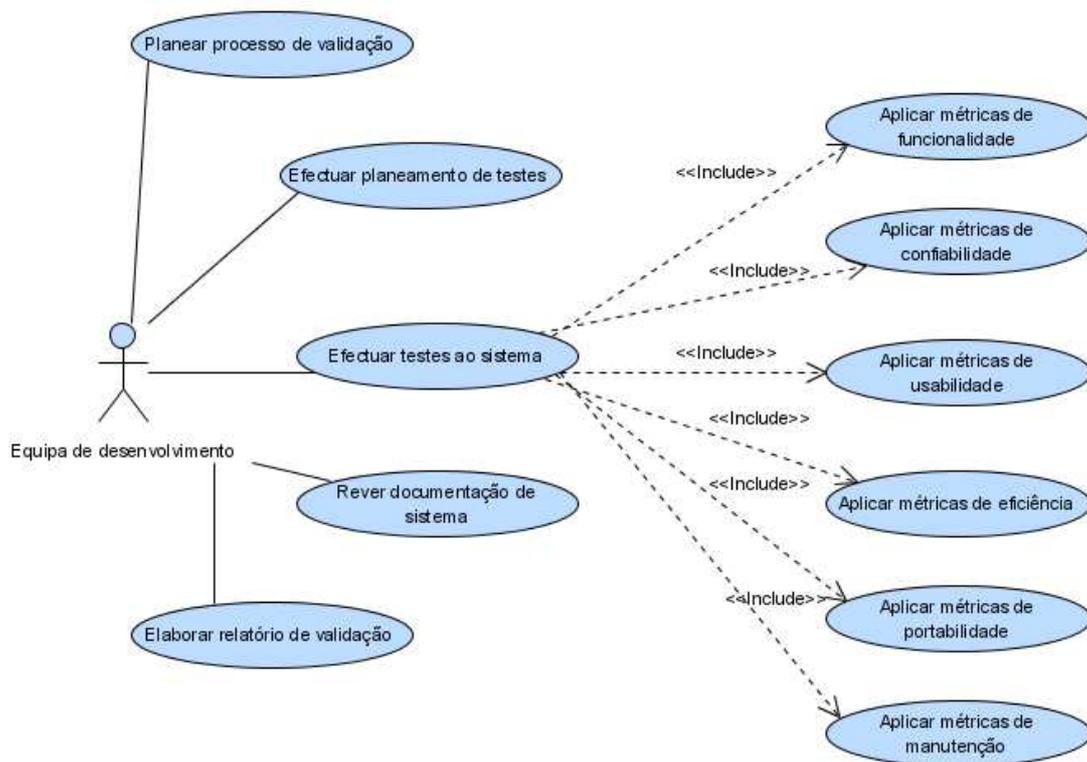


Figura 10: validação interna

Para cada uma das etapas acima apresentadas, será apresentado abaixo um diagrama contendo as actividades que delas fazem parte. Todas as métricas referidas nos diagramas estão descritas em anexo ao presente documento. Esta descrição inclui a sua fórmula e respectiva interpretação, os *outputs* sobre os quais incidem, bem como o objectivo que lhes está implícito.

Comecemos pela validação da **funcionalidade** do sistema. Para validar esta característica, deverão ser definidas métricas de adequação, interoperabilidade, segurança e funcionalidade, com o propósito de verificar em que medida é que as funções e itens implementados até à fase do ciclo de desenvolvimento de software em que nos encontramos, foram correctamente implementadas. No contexto específico dos sistemas de gestão documental, faz sentido utilizar métricas de todas estas categorias, no entanto, para situações específicas, algumas delas poderão ser dispensáveis. Consideremos o exemplo de um sistema de gestão documental bastante simples, que não interage com quaisquer outras

aplicações. Para este caso, as métricas de interoperabilidade não se justificarão. Portanto, não só para a questão da funcionalidade, mas também para todas as métricas definidas neste modelo, é importante ressaltar que a sua aplicação depende da dimensão e especificidades do sistema sobre o qual são aplicadas.

O diagrama abaixo apresentado, descreve a forma como deverá ser efectuado o processo de validação da funcionalidade do sistema, bem como as componentes avaliadas no seu decorrer. A tabela apresentada em anexo apresenta, tal como foi já referido, o conjunto de métricas definidas pela norma ISO/IEC 9126 para cada um dos aspectos acima mencionados.

A primeira etapa de validação da funcionalidade de um sistema de gestão documental, consiste na avaliação da adequação funcional. Esta métrica incide directamente sobre cada uma das etapas do fluxo de trabalho (descrito no capítulo anterior), uma vez que este é a base de qualquer sistema de gestão documental. Para além deste aspecto, a adequação funcional avaliará também os mecanismos de comunicação implementados pelo sistema, e sua conformidade com o especificado.

Segue-se a medição da complitude da adequação funcional. Mais uma vez, a característica dos sistemas de gestão documental inerente a esta actividade, são os fluxos de trabalho, pelas razões apresentadas acima. Também como no caso anterior, será avaliada a conformidade entre esta medida e os valores especificados como correctos.

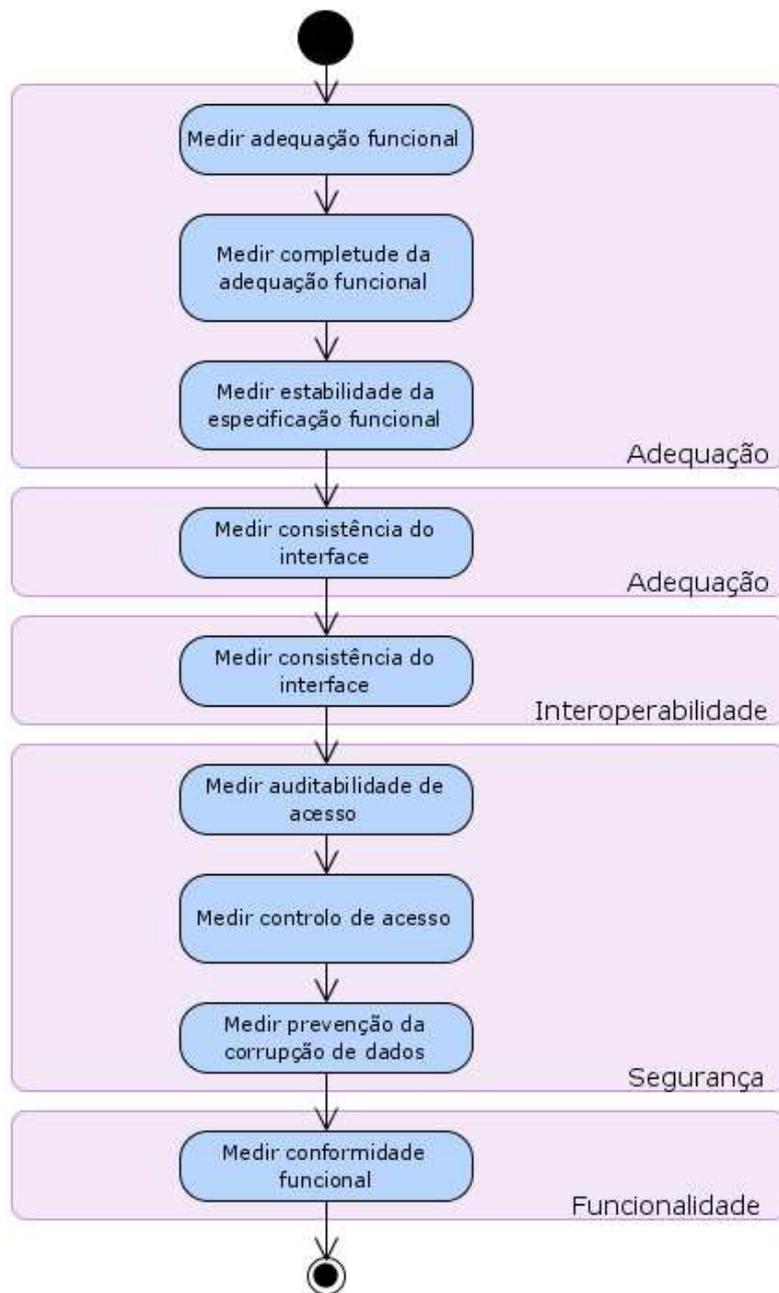


Figura 11: Métricas de funcionalidade

Para terminar a validação da funcionalidade ao nível da adequação, deverá ser medida a estabilidade da especificação funcional, que incidirá também sobre as características mencionadas nos parágrafos anteriores.

Ao nível da interoperabilidade, deverá ser medida a consistência do interface. Por outras palavras, esta medição, que se relaciona directamente com a característica de interoperabilidade dos sistemas de

gestão documental, deverá analisar se os protocolos de interface foram correctamente implementados.

As actividades de validação da segurança, ao nível da funcionalidade, prendem-se essencialmente com o controlo de acessos ao sistema. Tanto a métrica de auditabilidade de acesso, como a de prevenção e controlo de dados, como a do controlo de acesso propriamente dito, visam analisar a existência e eficiência de mecanismos de segurança que garantam não só a integridade do sistema, mas também que as suas diferentes componentes, e os documentos nele armazenados, são acedidos e/ou modificados apenas por quem de direito.

Para terminar, é necessário também validar a conformidade funcional do sistema, ou seja, garantir, através da análise de registos, fluxos de trabalho e integração com outros sistemas, que as características funcionais do produto estão de acordo com as normas e convenções legais aplicáveis.

Terminada a análise da funcionalidade, segue-se a verificação da **confiabilidade** relativa ao sistema em causa. A utilização de métricas de confiabilidade – maturidade, tolerância a falhas, recuperabilidade e confiabilidade – visa, aquando da validação interna, prever se o produto final será capaz de alcançar as metas previamente definidas no que diz respeito às necessidades de confiabilidade inerentes ao mesmo. O processo de validação da confiabilidade é executado da forma que podemos observar na figura apresentada abaixo.

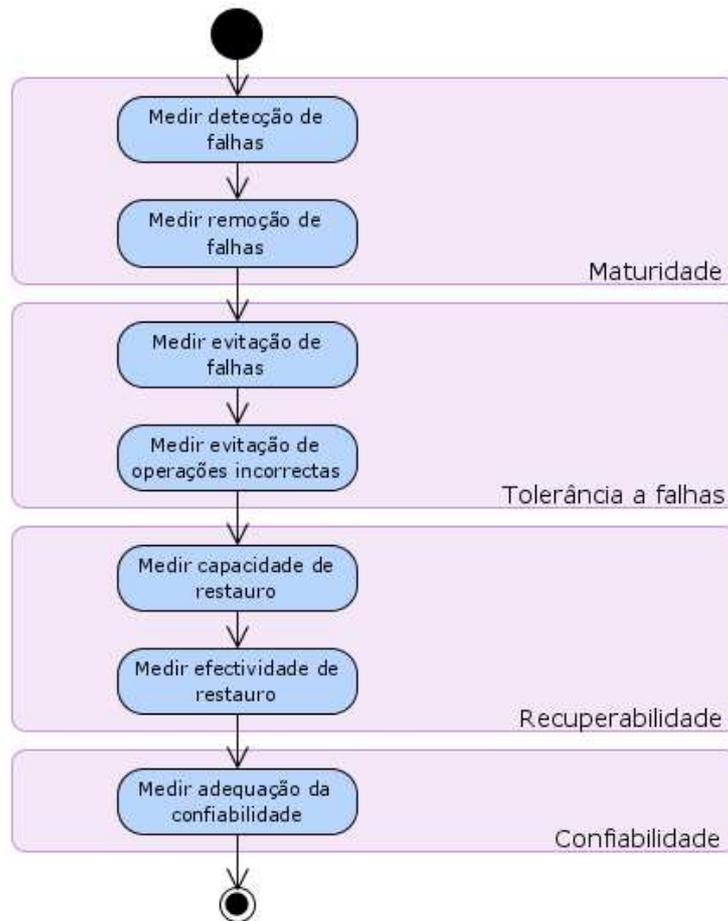


Figura 12: métricas de confiabilidade

A validação interna da confiabilidade do sistema é iniciada ao nível da sua maturidade. Para esta característica, deverá ser analisado todo o sistema, e não apenas uma componente específica, no que diz respeito à detecção e remoção de falhas.

As métricas para avaliação da tolerância a falhas, por suas vez, abordam a integridade do sistema, procurando contabilizar a evitação de padrões de falha e operações incorrectas.

A análise da recuperabilidade prende-se directamente com uma das necessidades de validação abordadas no capítulo anterior: o registo de operações. A medição da capacidade e efectividade de restauro visam, precisamente, garantir a integridade desta componente, presente em qualquer sistema de gestão documental.

Por último, e tal como aconteceu para as características funcionais, será necessário validar a adequação da confiabilidade, garantindo que esta está em conformidade com os requisitos legais aplicáveis.

A utilização de métricas para avaliar a **usabilidade** do sistema, por sua vez, diz respeito à previsão de se o sistema estará apto a ser compreendido, entendido, atractivo e utilizado, bem como se estará em concordância com os protocolos e normas requeridos. As métricas de usabilidade abrangem a compreensão, aprendizagem, operabilidade, atractividade e usabilidade do sistema, conforme poderemos ver no diagrama abaixo, e descrito com um maior pormenor nas tabelas em anexo.

A validação da usabilidade de um sistema é efectuada por meio de medições a diferentes níveis. No que diz respeito à compreensão, deverá, antes de mais, ser medida a complitude de descrição, procurando-se, com isto, garantir a conformidade entre a URS e o código-fonte, dois dos *outputs* mencionados acima nesta secção. Para além disto, e tratando-se esta de uma métrica que trata directamente com o utilizador típico do sistema, deverá também ser medida a compreensão e evidência das funções, prendendo-se estas medições directamente com a análise da utilização do sistema.

Ao nível da aprendizagem, mais uma vez, o utilizador terá um papel fulcral, uma vez que a medição da complitude da documentação para o utilizador e/ou meios de ajuda visa então garantir a conformidade entre esta documentação e os requisitos de utilização do sistema.

A validação das características de operabilidade de um sistema de gestão documental passa pela efectuação de um conjunto de medições:

- Verificação da validade do *Input*: no capítulo anterior foi já mencionado que os dados de entrada e respectivos formatos seriam uma das componentes de um sistema de gestão

documental a ser validada. Esta métrica é responsável por esse processo.

- Personalização: a medição da personalização, por sua vez, prende-se directamente com a questão das permissões de utilizadores, característica também relevante no contexto dos sistemas de gestão documental.
- Capacidade de monitorizar o estado de operações: este valor será utilizado para validar o registo de operações, já referido como uma das características fundamentais de um sistema de gestão documental.
- Consistência operacional: a medição da consistência operacional, tal como se pode depreender pelo nome, deverá garantir a integridade das operações do sistema, e sua conformidade com os requisitos previamente estabelecidos.
- Clareza de mensagens; clareza dos elementos do interface: estas duas medições, analisam mais uma vez a perspectiva do utilizador e visam garantir a usabilidade do sistema em análise.
- Recuperabilidade de erros operacionais: esta última medição, no contexto de operabilidade de um sistema, prende-se mais uma vez com a análise da sua integridade.

Por último, e à semelhança do que aconteceu para as características anteriores, também a adequação da usabilidade do sistema deverá ser validada, com vista à garantia de que este aspecto está em concordância com as regras e normas legais em vigor.

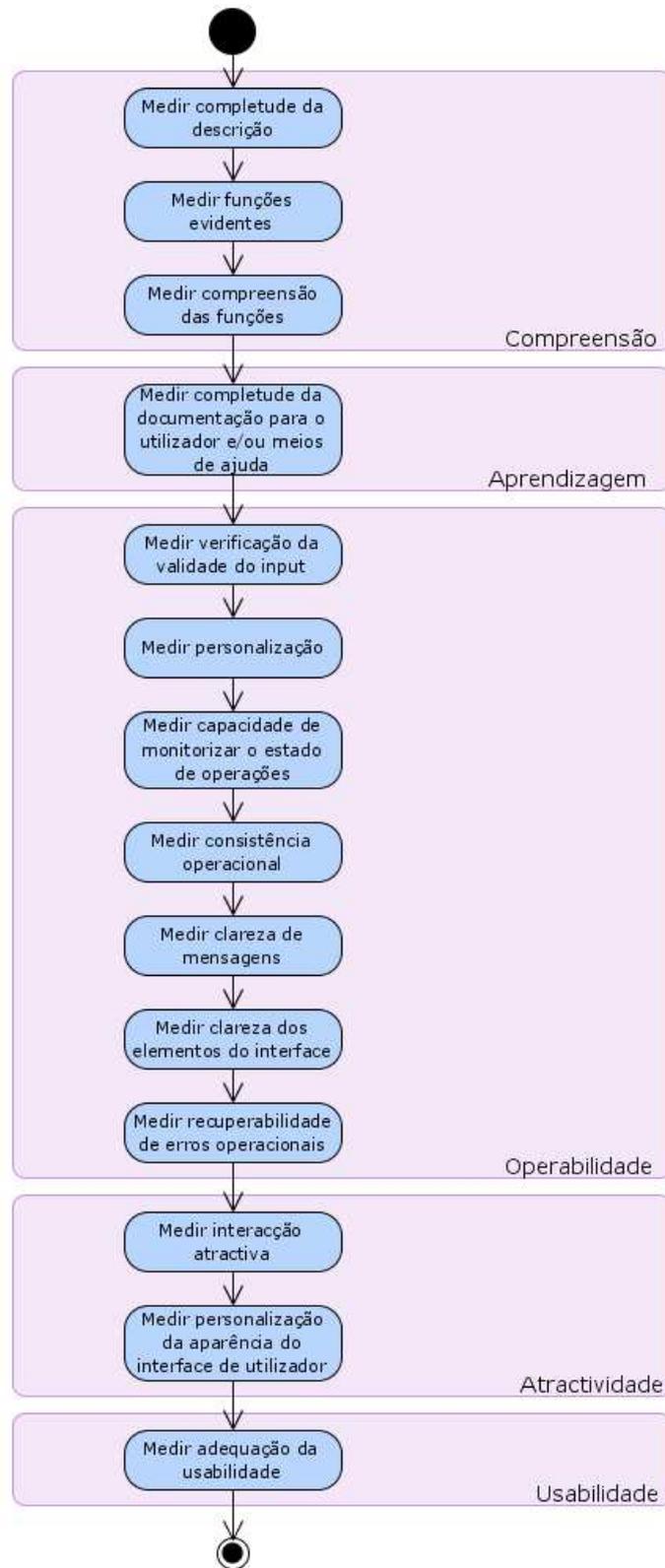


Figura 13: Métricas de usabilidade

As métricas de **eficiência**, tal como o nome indica, procuram prever a eficiência do sistema. Para tal, deverão ser efectuados testes ao

mesmo, em condições que reproduzam o seu funcionamento normal, e aplicadas métricas temporais e de eficiência, com vista à obtenção de valores que permitam concluir então acerca da eficiência do produto, conforme podemos observar na figura que se segue.

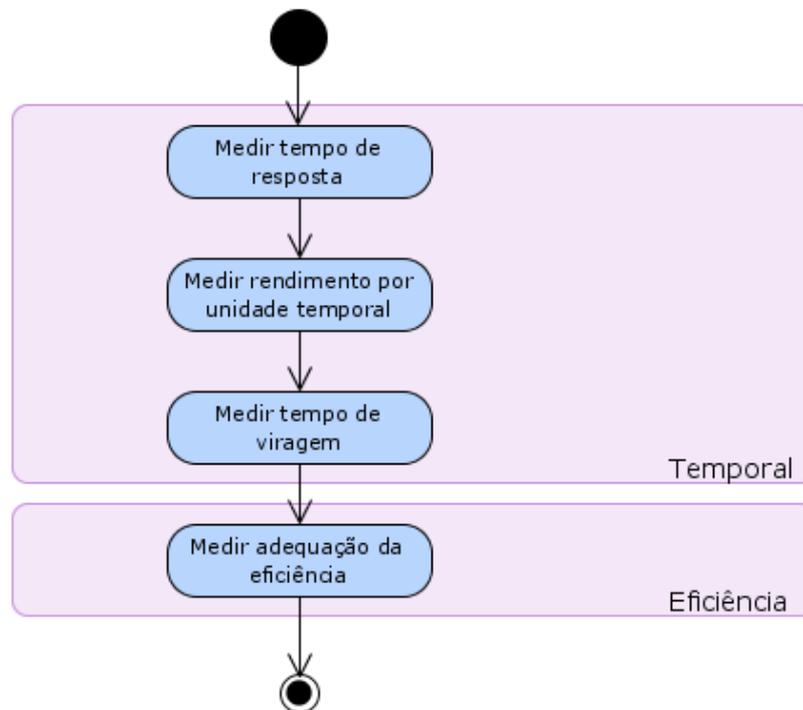


Figura 14: métricas de eficiência

A validação da eficiência de um sistema de gestão documental é efectuada ao nível temporal. No capítulo anterior vimos já que os tempos de execução e resposta são um dos aspectos fulcrais na validação de um sistema de gestão documental – não faria sentido manter um sistema cujos tempos não fossem satisfatórios para o seu propósito final. Assim, deverão ser medidos os tempos de resposta e rendimento por unidade temporal, bem como o tempo estimado para executar um conjunto de tarefas, sendo que esta última abordará também a questão da comunicação (mensagens internas ao sistema, e entre sistemas) e fluxos de trabalho (seguimento de tarefas).

Para além disto, e tal como aconteceu para as características anteriores, deverá também ser validada a adequação da eficiência no contexto legal e normativo, garantindo a conformidade entre os tempos de execução e as normas estabelecidas neste âmbito.

Segue-se a verificação da **manutenção** do produto. Ou seja, nesta etapa, serão aplicadas métricas, abordando questões relacionadas com a capacidade de análise, alterabilidade, estabilidade, capacidade de teste e manutenção, com o propósito de prever o esforço necessário para efectuar possíveis alterações ao sistema, no futuro.

A validação da manutenção de um sistema de gestão documental começa com a avaliação da sua capacidade de análise, através da medição da complitude do registo de actividades. Esta métrica será aplicada sobre uma das características dos sistemas de gestão documental anteriormente abordada neste documento: o registo de operações.

Também a alterabilidade do sistema incide sobre o registo de operações, e pode ser validada através da análise do registo de mudanças.

A estabilidade de um sistema de gestão documental, por sua vez, é validada através da aplicação das métricas de impacto de mudança e localização do impacto de mudança, que avaliam a integridade do sistema.

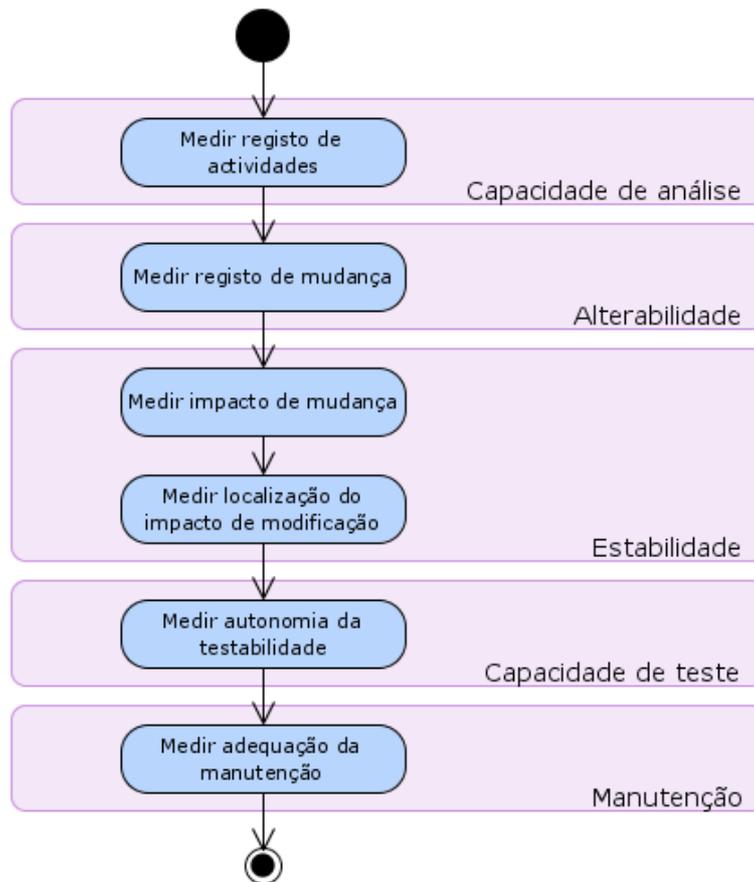


Figura 15: métricas de manutenção

Ainda no contexto da validação da manutenção do sistema, deverá ser medida a sua capacidade de teste, analisando em que medida é que o software pode ser testado de forma independente. Esta análise incide mais uma vez sobre a integridade do produto e sua conformidade com os requisitos especificados.

Por último, resta validar a adequação da manutenção, ou seja, se esta característica se encontra em conformidade com os requisitos legais a ela aplicáveis.

Para terminar o processo de teste inerente à validação interna, é necessário avaliar a **portabilidade** do sistema. Isto significa que será necessário medir e analisar a sua adaptabilidade, coexistência e portabilidade, com o objectivo de concluir acerca do efeito que aquele produto de software específico poderá ter num sistema global, durante a

actividade de transporte. A imagem abaixo descreve então o processo de validação da portabilidade do sistema.



Figura 16: métricas de portabilidade

O primeiro passo para validar a portabilidade de um sistema de gestão documental, consiste na análise da adaptabilidade de estruturas de dados. Esta métrica incidirá directamente sobre os conteúdos do sistema, formato e organização dos seus dados, que, conforme pudemos observar no capítulo anterior, são algumas das necessidades de validação subjacentes aos sistemas de gestão documental.

A esta actividade, seguir-se-á a validação da adaptabilidade ao ambiente organizacional, que deverá garantir a conformidade desta com o especificado.

Também a análise da adaptabilidade ao software se reveste de uma elevada importância no contexto da validação interna de um sistema de gestão documental, por se prender directamente com a característica dos sistemas de gestão documental que é a interoperabilidade com outros sistemas, assim como a normalização de documentos, uma vez que esta é uma exigência da anterior.

A interoperabilidade entre um sistema de gestão documental e outros sistemas deverá também ser analisada através da aplicação da métrica designada por coexistência disponível, que tem como propósito verificar até que ponto é que um produto é capaz de partilhar um conjunto de recursos com outra solução de software. A aplicação desta medição terá ainda repercussões na avaliação das características de normalização de documentos e execução de vários processos em simultâneo.

A validação da portabilidade termina com a análise da sua conformidade com as normas e requisitos legais aplicáveis.

Antes de prosseguir, resta apenas salientar que, conforme pudemos observar no capítulo relativo à validação, todos os testes e procedimentos executados no decorrer do processo de validação, bem como a execução das métricas acima apresentadas, deverão ser registados em relatórios de teste, e, acima de tudo, no relatório de validação. De facto, para além dos *outputs* sobre os quais incidem as métricas definidas na norma ISO/IEC 9126, existe ainda um conjunto de documentos que deverão constar do modelo de validação, seja qual for a parte que está a ser executada – interna, externa ou em uso:

- Plano de validação: este documento deverá incluir componentes como a descrição da equipa de validação e suas responsabilidades (líder de equipa, gestor de documentos, coordenador de testes), análise de riscos, *milestones*, e

quaisquer outras informações consideradas relevantes para o processo.

- Especificação de testes: deste documento deverá conter a descrição dos testes, o objectivo a eles inerente, bem como os próprios guiões de teste.
- Relatório de testes: após execução dos testes, deverá ser redigido um relatório documentando a forma como foram executados, quem os executou, quem testemunhou, e o seu resultado.
- Relatório de validação: após a aplicação do modelo acima definido, deverá ser produzido um documento relativo ao processo. Os testes executados, seus resultados, alterações efectuadas, alterações propostas, equipa de validação que actuou, são apenas os focos essenciais deste documento. Todo o processo deverá estar bem documentado no relatório de validação.

Conforme pudemos depreender das descrições apresentadas em anexo, para cada conjunto de métricas a validação executada nesta fase - interna - tem essencialmente a função de prever. Nesta altura do ciclo de vida de um produto de software, um sistema de gestão documental, neste caso específico, procura-se verificar se o mesmo está a cumprir os requisitos, e se tem potencial para, no final do seu ciclo produtivo, funcionar de acordo com o previsto. Na validação e externa e em uso, os objectivos serão outros, como passaremos a ver nas secções seguintes.

4.3. Validação durante a fase de conclusão do desenvolvimento de SGD

A validação de um sistema não é um processo exclusivo ao seu desenvolvimento. De facto, foi já referido no presente documento que ela poderá ocorrer em três fases distintas do ciclo de vida de um produto de software. O momento da sua codificação foi a primeira etapa que abordamos. Passemos agora então à validação durante a fase de teste, da qual é ilustrativa a imagem abaixo.

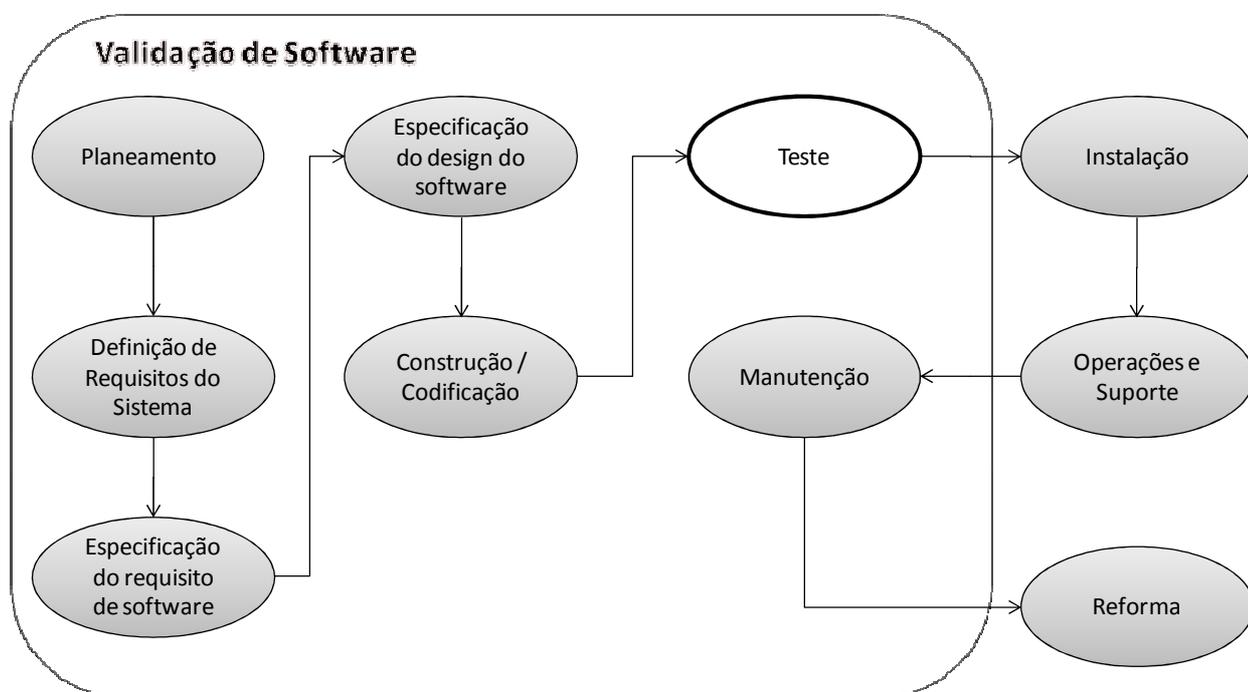


Figura 17: validação externa no ciclo de vida do software

No capítulo 3 do presente documento, dedicado à temática da gestão documental, foi referido que a validação deste tipo de sistema poderá ser executada por dois tipos de utilizador distintos: a equipa de desenvolvimento, e o utilizador final. Sendo a validação externa realizada antes da fase de instalação do sistema, esta será, então, a última validação a ser executada pela equipa de desenvolvimento do produto, e prende-se directamente com a verificação da qualidade externa associada ao mesmo. Se, na validação aquando da codificação, o produto

propriamente dito não existia ainda, nesta fase tal já não acontece. O teste ocorre precisamente após a construção/codificação do software, e antes da sua instalação.

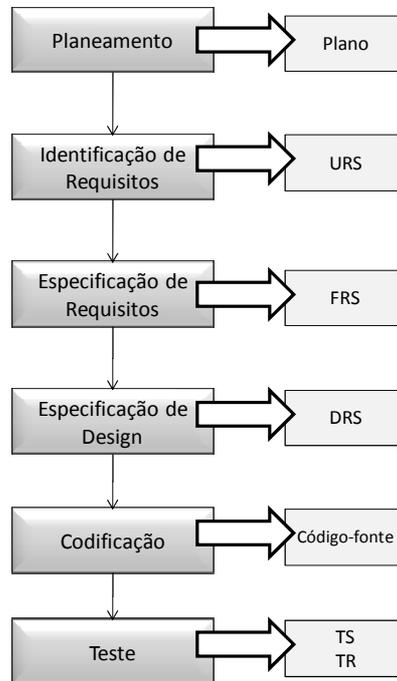


Figura 18: *outputs* validação externa

Se na validação interna existia um conjunto de *outputs* que, através das métricas acima descritas, deveriam ser validados de forma a garantir a sua qualidade, na validação externa, os *outputs* manuseados são semelhantes aos da validação interna, mais os *outputs* decorrentes da etapa que foi acrescentada – o teste. Da etapa de teste decorrem dois novos documentos: **Test Specification** (TS) e **Test Report** (TR). Anteriormente neste documento, havíamos já visto que estes dois documentos eram componentes do processo de validação interna, mas apenas como consequência do mesmo. Aquando da validação externa, eles deixam de ser o produto, para passar a ser o *Input*, ou seja, é sobre eles que, desta vez, irão incidir as métricas definidas na norma ISO/IEC 9126. Para além destes, também os documentos relativos às fases anteriores serão utilizados no processo de validação externa, devendo já reflectir as alterações realizadas após o processo de validação interna. A

validação externa consistirá então na aplicação de um conjunto de métricas, com o propósito de garantir a funcionalidade, confiabilidade, usabilidade, eficiência e manutenção do referido produto, incidindo agora sobre o produto de software já terminado, conforme podemos observar no diagrama apresentado abaixo.

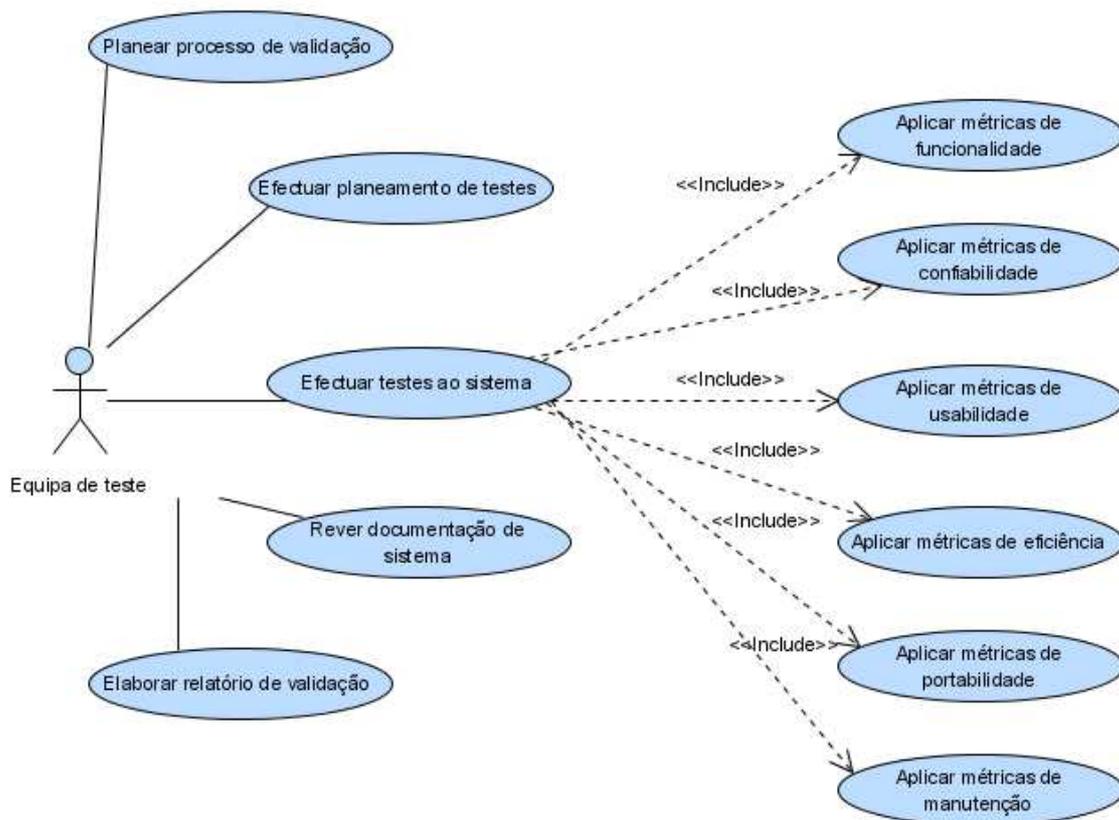


Figura 19: validação externa

Tal como para a validação interna, serão apresentadas em anexo as tabelas que com descrição e explicação das métricas definidas na norma ISO/IEC 9126 que deverão ser aplicadas com vista à execução de um processo de validação externa eficiente. Também como vimos para a questão da validação interna, serão apresentados diagramas contendo a sequência de actividades a executar com vista a validar cada uma das características inerentes à avaliação da qualidade externa de um produto de software de gestão documental.

A primeira característica a ser validada é a **funcionalidade** do sistema, através das métricas de adequação, exactidão, interoperabilidade, segurança e funcionalidade, que podemos observar nas tabelas em anexo. A Figura 20: métricas de funcionalidade ilustra então as actividades subjacentes a este processo.

De seguida deverá ser efectuada a verificação da **confiabilidade** através de métricas externas (Figura 21: métricas de confiabilidade). Este procedimento visa, após a realização de testes, medir o grau de confiabilidade do sistema aquando da sua execução. A tabela apresentada em anexo mostra as métricas definidas pela norma ISO/IEC 9126, adaptadas para o processo de validação externa.

Tal como as métricas de **usabilidade** internas, também as externas procuram determinar em que medida é que o sistema está apto a ser compreendido, entendido, atractivo e utilizado. No entanto, se no primeiro caso era apenas efectuada uma previsão do que seria a usabilidade do sistema, neste caso é já uma medida presente. Uma vez que as questões de usabilidade poderão ser relativas, já que são influenciadas por factores externos ao teste (como por exemplo as capacidades do utilizador que testa), é recomendável que estas medidas sejam efectuadas por uma amostra de pelo menos 8 utilizadores distintos de um mesmo sistema. A Figura 22: métricas de usabilidade apresenta o conjunto de actividades a concretizar no âmbito do teste da usabilidade do sistema.

As métricas de **eficiência**, procuram, tal como o nome indica, medir a eficiência do sistema. Atributos como consumos temporais e de recursos são alguns dos indicadores medidos para concluir se o sistema se adequa ou não aos padrões de eficiência definidos.

De seguida, é feita a verificação da capacidade de **manutenção** do produto. Que consiste, mais uma vez, na aplicação de um conjunto de

métricas com vista a avaliar o comportamento do sistema quando é efectuada a sua manutenção ou uma qualquer alteração.

Por último, deverá ser efectuada a avaliação da **portabilidade** do sistema. As métricas de portabilidade deverão analisar as consequências que actividades de porte terão para o sistema, nomeadamente no que diz respeito à sua adaptabilidade, capacidade de instalação, coexistência e capacidade de instalação, conforme podemos observar na Figura 25: métricas de portabilidade.

Mais uma vez, todas estas características – funcionalidade, confiabilidade, eficiência, manutenção, usabilidade e portabilidade – apesar de incluídas no modelo, poderão ser adaptadas consoante a dimensão e a dinâmica do sistema de gestão documental que se pretende validar. Cada sistema, apesar de pertencer à mesma categoria, terá uma dimensão própria, abrangendo diferentes tipos de dados, diferentes áreas funcionais, e até podendo interagir ou não com outros sistemas. Como vimos anteriormente neste documento, um sistema de gestão documental assenta em três pilares básicos: o ciclo de vida dos documentos, fluxo de trabalho, e o armazenamento. Se o ciclo de vida dos documentos é uma questão bastante linear, o mesmo não se poderá dizer das outras duas. Os fluxos de trabalho, por exemplo, serão distintos consoante o contexto ao qual o sistema seja aplicado. Não será a mesma coisa, por exemplo, validar um pequeno repositório que manuseia apenas documentos simples, ou um sistema que englobe o tratamento e utilização de documentos simples e compostos e interaja com outros sistemas. Assim, será necessário ter em mente que o modelo aqui descrito é apenas traçado em linhas gerais, de forma a que seja possível adaptá-lo a qualquer tipo de sistema de gestão documental, sendo que necessitará certamente de algumas alterações para se adaptar ao contexto específico em que for utilizado.

Segue-se então a apresentação dos diagramas contendo as actividades a realizar para testar a validade de cada uma das características acima apresentadas, e respectivas áreas nas quais estas métricas se encontram.

A validação da funcionalidade de um sistema de gestão documental inicia-se com a aplicação de métricas que visam avaliar a sua adequação:

- Adequação funcional: esta medição actua directamente sobre os fluxos de trabalho e comunicação – características essenciais de um sistema de gestão documental – e visa não só analisar a adequação das funções analisadas, mas também garantir a sua integridade e conformidade de acordo com os requisitos especificados.
- Complitude da implementação funcional: o fluxo de trabalho é mais uma vez a característica do sistema abarcada por esta métrica, que procura garantir a conformidade da complitude da implementação funcional com o especificado.
- Cobertura da implementação funcional: esta medição analisa todo o sistema de gestão documental, com o propósito de verificar a correcção da implementação funcional.
- Estabilidade da especificação funcional: a última das métricas de adequação a aplicar quando se valida a funcionalidade de um sistema de gestão documental, ao nível externo, visa garantir a integridade e conformidade da especificação funcional de acordo com o previsto.
- Ao nível da validação da exactidão de um sistema de gestão documental, podemos definir duas medições a executar: a verificação da exactidão de expectativa e da exactidão computacional. Ambas avaliam a conformidade entre resultados obtidos e resultados esperados, mas a última fá-lo do ponto de vista do utilizador típico do sistema, avaliando assim também a utilização do mesmo.

- Segue-se a avaliação da interoperabilidade do sistema, através da medição da troca de dados, baseada no formato dos dados e nas tentativas de sucesso do utilizador. O primeiro caso analisa características importantes para um sistema de gestão documental, tais como a organização de dados, formato de dados, comunicação e interoperabilidade com outros sistemas, enquanto que o segundo se foca nas definições de utilização, controlo de acessos, comunicação e integração.
- A fase que se segue na validação da funcionalidade de um sistema de gestão documental, é a medição e análise da sua segurança. As métricas definidas no diagrama acima visam, de um modo geral, analisar as características de controlo de acessos, confidencialidade, registo de operações e integridade, inerentes a um sistema de gestão documental genérico.
- Para terminar o processo de validação da funcionalidade de um sistema de gestão documental a nível externo, deverá ser validada a conformidade funcional e conformidade standard do interface do mesmo, de acordo com as normas e requisitos legais aplicáveis.

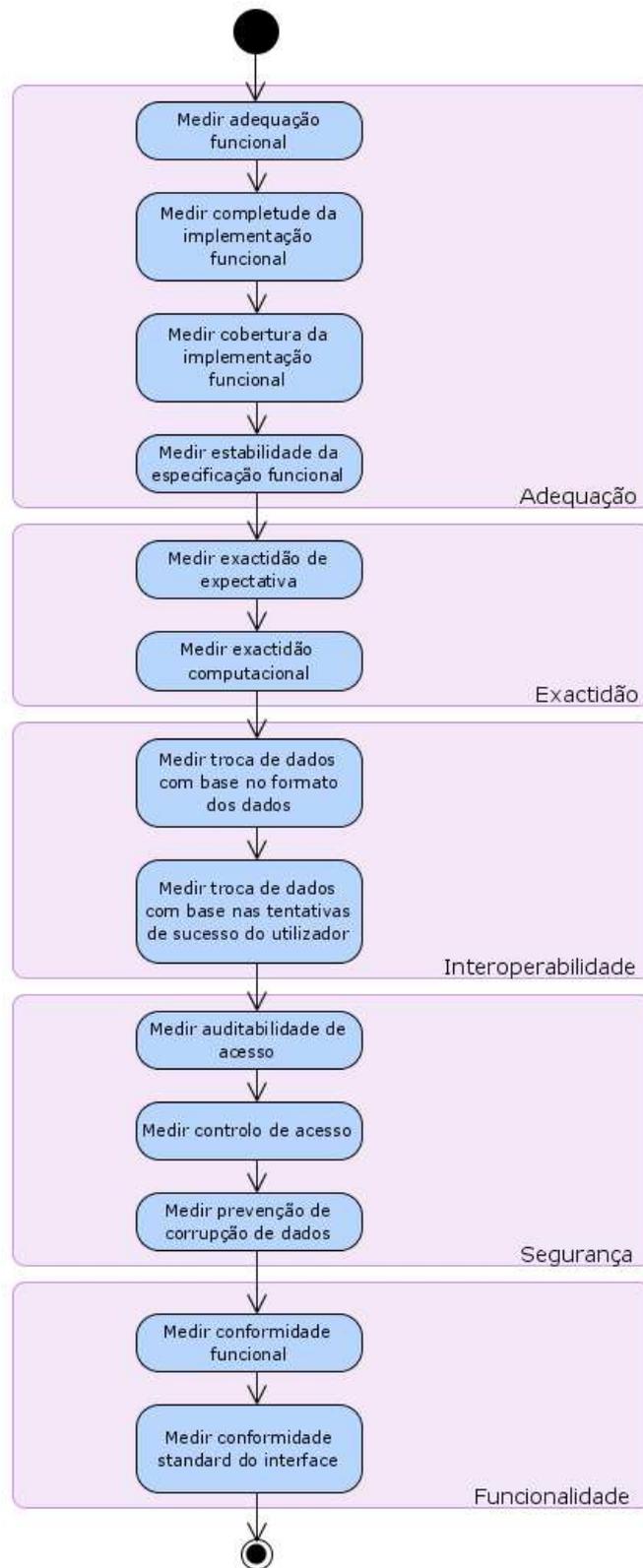


Figura 20: métricas de funcionalidade

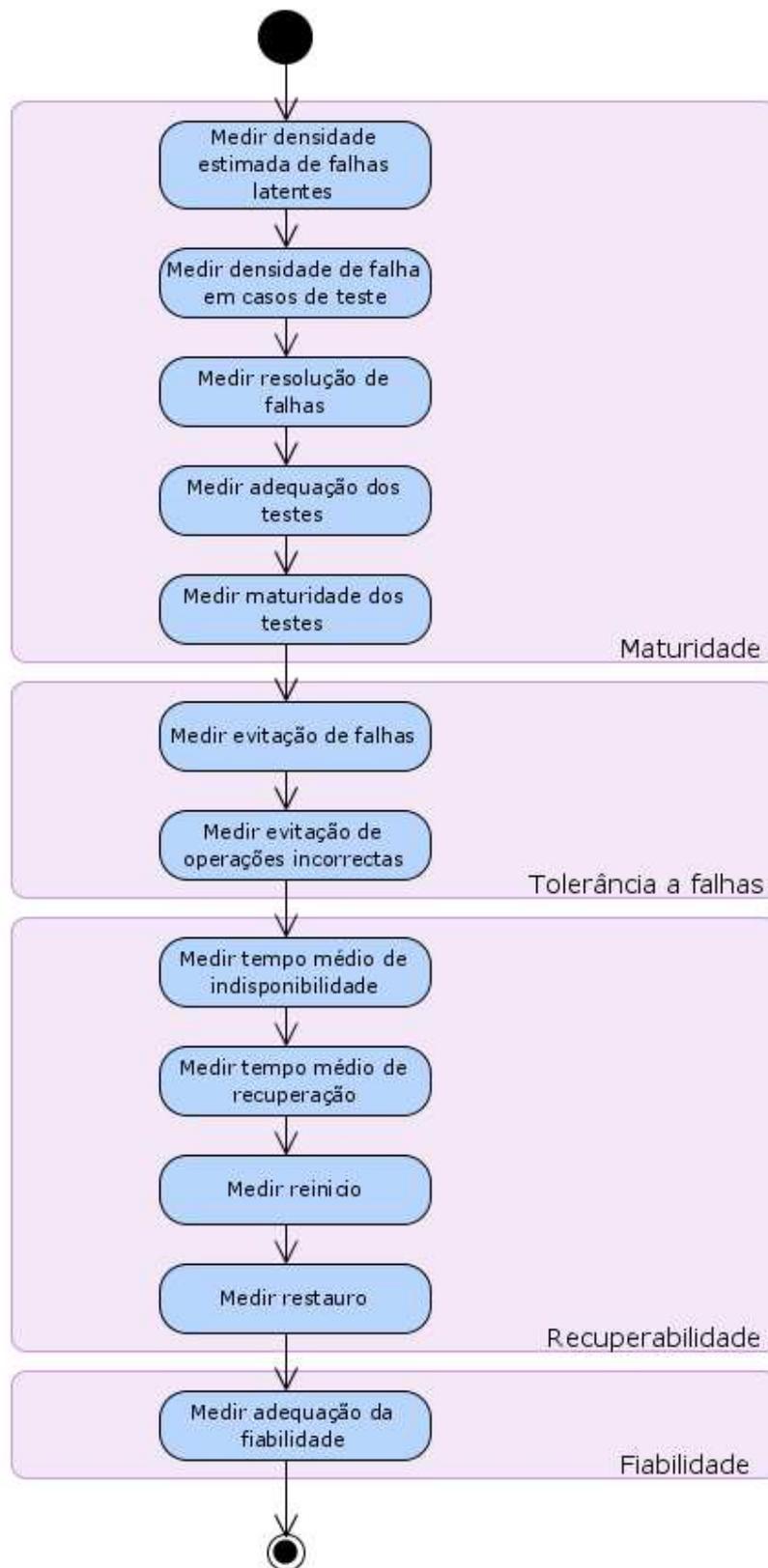


Figura 21: métricas de confiabilidade

Validada a funcionalidade de um sistema de gestão documental, é tempo de prosseguir para a validação da sua confiabilidade. Este processo

ocorre em quatro níveis distintos, como teremos oportunidade de observar na descrição abaixo.

Inicialmente, deverão ser validados os aspectos relacionados com a maturidade do sistema em questão. Para isso será aplicado o conjunto de métricas apresentadas no diagrama acima, todas elas com o propósito de garantir a integridade do sistema, e sua conformidade com os requisitos previamente especificados.

Seguir-se-á a validação no que diz respeito a tolerância a falhas. Neste aspecto, será avaliada a integridade do sistema no que concerne à evitação de falhas e de operações incorrectas.

Segue-se a avaliação da recuperabilidade do sistema. As medições efectuadas neste âmbito têm como propósito garantir que a disponibilidade do sistema se encontra dentro dos padrões apropriados, sendo que as duas primeiras – tempo médio de indisponibilidade e tempo médio de recuperação – também incidem sobre os tempos de execução do sistema, outra das necessidades de validação de sistemas de gestão documental apontada no capítulo anterior.

Para terminar, resta analisar a fiabilidade do sistema, e sua adequação, ou seja, se a fiabilidade do sistema se encontra em conformidade com os requisitos legais e normativos definidos para este.

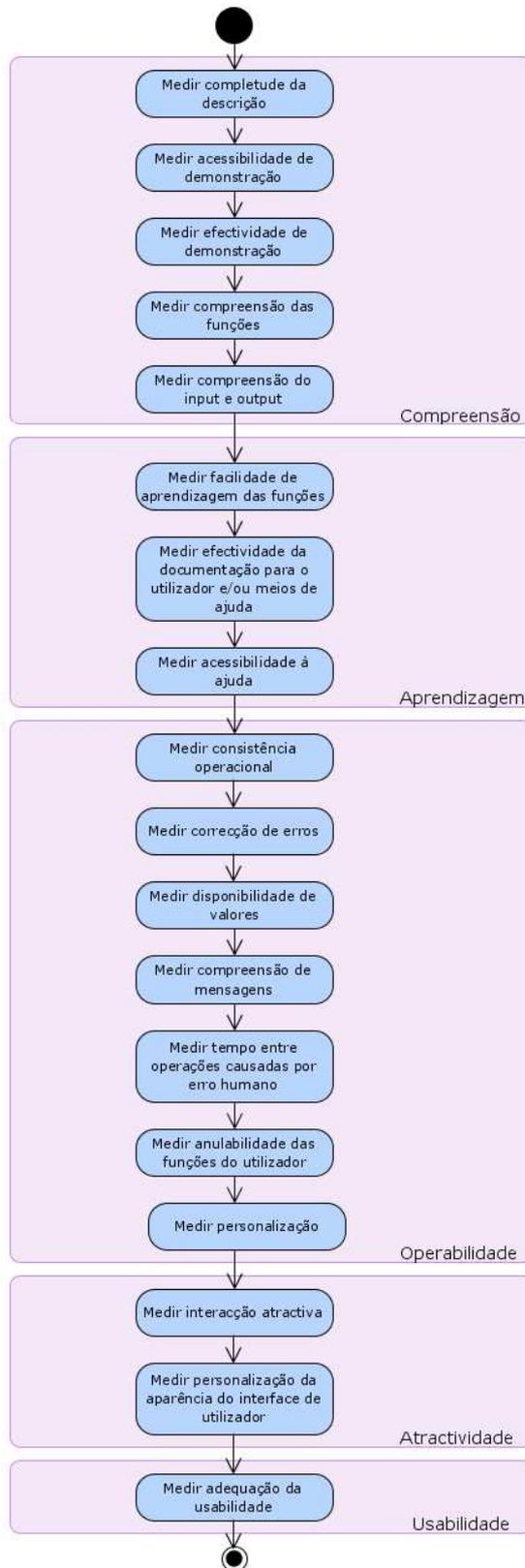


Figura 22: métricas de usabilidade

A validação da usabilidade de um sistema de gestão documental, ao nível externo, é efectuada seguindo o conjunto de actividades que podemos observar no diagrama acima.

Ao nível da compreensão, deverá ser analisada a complitude da descrição, acessibilidade e efectividade de demonstração. Todas estas métricas incidem sobre a avaliação da conformidade da descrição e demonstração, do ponto de vista do utilizador. Para além das referidas, deverão ainda ser aplicadas métricas relativas à compreensão das funções, do *Input* e do *output*, também estas relacionadas com a avaliação da utilização do sistema, e esta última mais especificamente com o formato de documentos, que para o caso dos sistemas de gestão documental são os dados de entrada e saída do sistema.

As métricas definidas no âmbito da validação da aprendizagem, são a facilidade de aprendizagem das funções, efectividade da documentação de apoio ao utilizador e acessibilidade à ajuda, sendo todas elas, de um modo geral, relacionadas com o utilizador, e a segunda, particularmente, relacionada com a conformidade entre esta documentação e os requisitos inicialmente especificados.

No que diz respeito à operabilidade, deverá ser medida a consistência operacional, que abrange todo o sistema; a correcção de erros, que se prende directamente com a utilização do sistema pelos seus actores; a disponibilidade de valores, que se relaciona não só com a componente de utilização, mas também com o formato de dados; a compreensão de mensagens, directamente relacionada com a característica de comunicação que vimos no capítulo anterior como uma das necessidades de validação de um sistema de gestão documental; o tempo entre operações causadas por erro humano, medição que aborda os tempos de execução da perspectiva do utilizador; e ainda anulabilidade das operações do utilizador e personalização, ambas relacionadas com a definição e implementação de permissões de utilizador, outro dos pontos

essenciais a ter em atenção quando se desenvolve um sistema de gestão documental.

Ainda dentro da validação da usabilidade, e ao nível da atractividade do sistema, deverá ser medida a atractividade do interface para o utilizador e capacidade de personalização desta componente. Mais uma vez, está em causa a validação das permissões do utilizador, e sua adequação face ao especificado.

Para terminar, resta efectuar a medida da adequação da usabilidade, procurando com isto garantir a conformidade da usabilidade com os requisitos legais e normativos existentes.

A validação da eficiência é efectuada nas vertentes temporal, utilização de recursos e eficiência.

No primeiro caso, são efectuadas as medições descritas no diagrama acima, que se aplicam directamente aos tempos de execução. A métrica de tempo de resposta aborda também a disponibilidade do sistema. A taxa de transferência, para além dos já referidos tempos de resposta, mede também a comunicação em tempo útil e a capacidade de correr vários processos em simultâneo. O tempo de viragem relaciona-se também com os fluxos de trabalho, e com a característica de comunicação inerente a este tipo de sistema.

No segundo caso, são distinguidas duas métricas: ocorrência média de erros de transmissão e utilização da capacidade de transmissão. A primeira aborda a questão da comunicação interna e entre sistemas, e a segunda relaciona-se com a disponibilidade e tempos de execução.

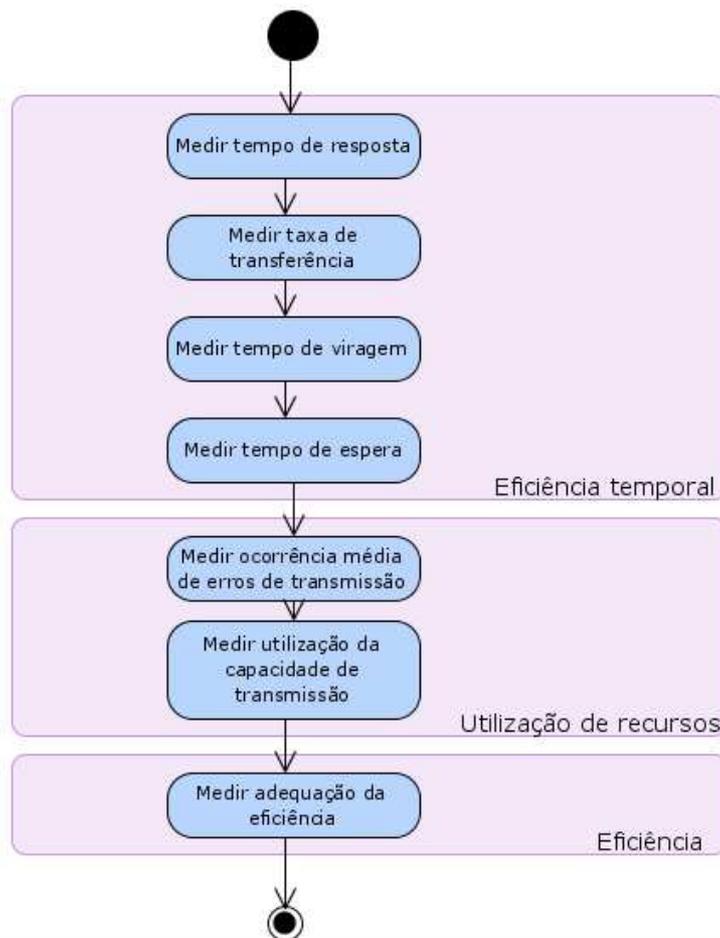


Figura 23: métricas de eficiência

Por último, no terceiro caso, deverá ser medida a conformidade da eficiência com os requisitos e normas legais.

A validação da manutenção de um sistema de gestão documental, ao nível externo, é feita ao nível da capacidade de análise, alterabilidade, estabilidade e manutenção.

No que concerne à capacidade de análise, são aplicadas as métricas “capacidade de análise de falhas”, “funções de diagnóstico” e “capacidade de rastrear auditoria”, que incidem sobre a funcionalidade de registo de operações. A estas medições acresce mais uma – capacidade de monitorização do estado – incidente sobre o registo de falhas.

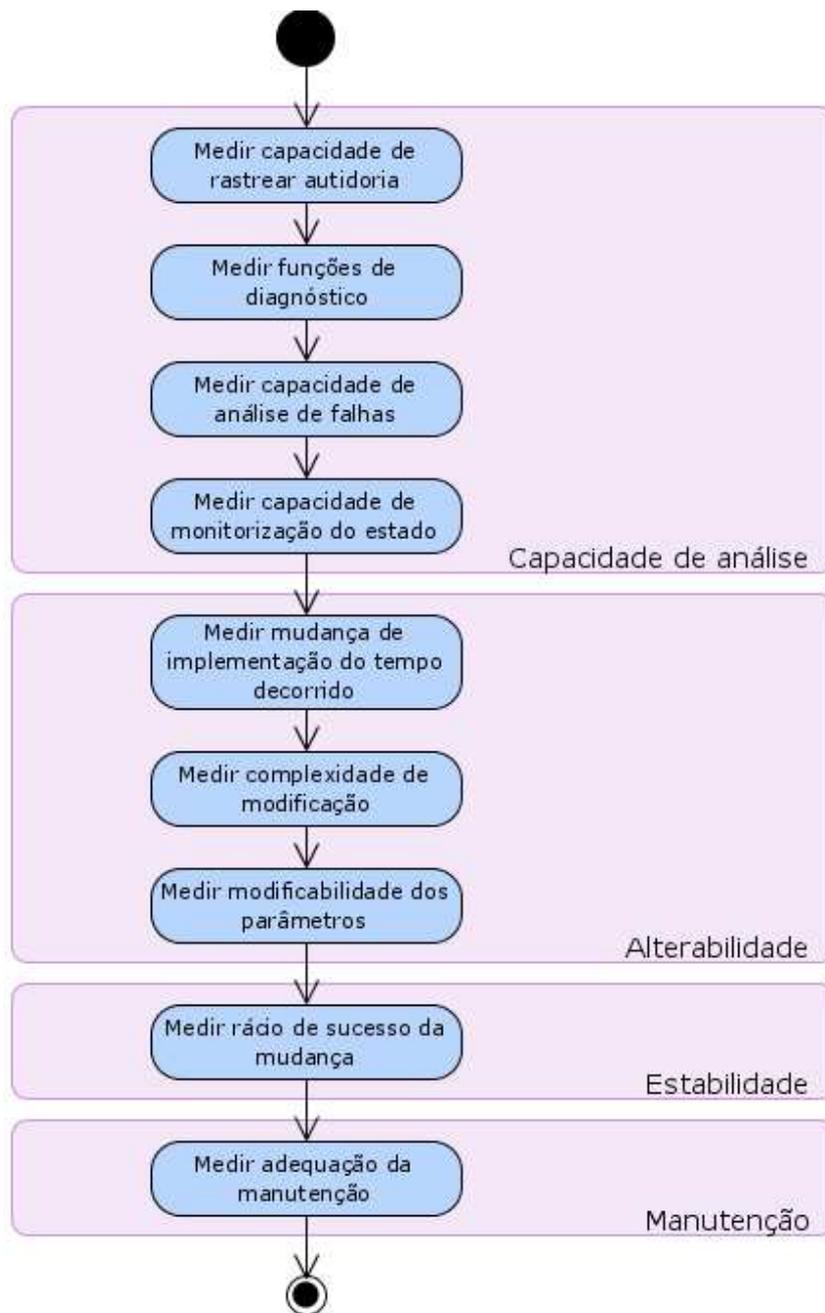


Figura 24: métricas de manutenção

A análise da alterabilidade é feita por via da aplicação de métricas que avaliam a possibilidade de mudar o software em consequência de uma falha, a complexidade de modificação e a modificabilidade de parâmetros, todas elas com impacto directo na integridade do sistema.

Relativamente à estabilidade, deverá ser avaliado o rácio de sucesso na mudança, que deverá estar em conformidade com os valores considerados como aceitáveis.

Para terminar a validação da manutenção, resta apenas verificar a sua adequação relativamente às regras, normas e convenções legais aplicáveis.

A fase de testes da validação externa de um sistema de gestão documental culmina com a validação da sua portabilidade, ou seja, a aplicação de um conjunto de métricas como forma de medir diversas componentes do sistema e garantir que este cumpre os requisitos para ele especificados. A portabilidade é medida sob várias vertentes:

Adaptabilidade: relativamente a este aspecto, são utilizadas três medições: adaptabilidade de estruturas de dados, com incidência sobre conteúdos, formatos e organização de dados; Adaptabilidade ao ambiente organizacional, visando garantir a conformidade entre a adaptabilidade prevista e a efectivamente verificada; Adaptabilidade ao ambiente de software, com incidência sobre o aspecto da integração com outros sistemas e a normalização de documentos, uma vez que esta é uma necessidade que se impõe como consequência da primeira.

Capacidade de instalação: no que diz respeito a esta componente, deverá ser avaliada a facilidade de instalação e reinstalação, do ponto de vista do utilizador, e com o propósito de garantir a sua conformidade com as especificações.

Coexistência: ao nível da coexistência é utilizada apenas uma medição, com o propósito de avaliar a coexistência disponível. Obviamente, esta métrica incidirá sobre a característica de interoperabilidade e integração com outros sistemas, presente nos sistemas de gestão documental.

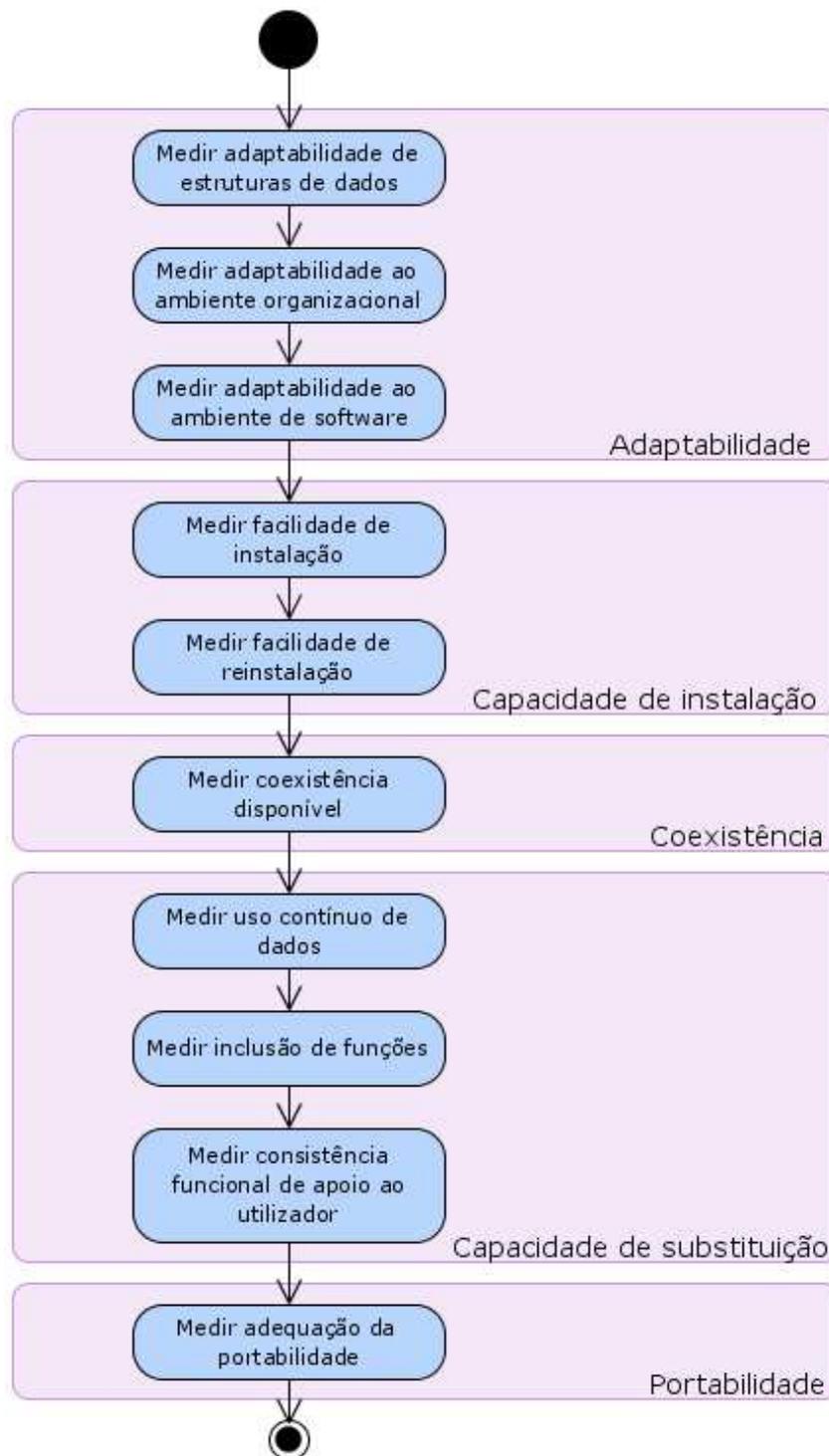


Figura 25: métricas de portabilidade

Capacidade de substituição: a capacidade de substituição é validada com recurso a três métricas: uso contínuo de dados, inclusão de funções e consistência funcional de apoio ao utilizador, sendo que a primeira aborda a problemática da normalização de documentos e formato de

dados, sendo esta última, aliás, também abordada pela segunda métrica, a par da garantia da integridade. A terceira medição engloba questões como utilização, conformidade e integridade.

Portabilidade: Para terminar, resta avaliar a conformidade entre a portabilidade do sistema e os requisitos normativos legais que lhe são impostos.

4.4. Validação durante a fase de utilização de SGD

Nas secções anteriores vimos a primeira e segunda partes do modelo global de validação de sistemas de gestão documental, referentes à garantia da sua qualidade interna e externa, respectivamente. Resta abordar a questão referente à validação de um sistema em uso.

Este tipo de validação ocorre quando o software já foi desenvolvido, testado e instalado, estando já em utilização num dado contexto. E esta é a razão que distingue este tipo de validação da validação externa – o contexto. É impossível garantir que um sistema funcione na perfeição em qualquer contexto, mesmo que tenha sido validado com sucesso anteriormente.

A validação, neste caso, ocorre então na fase de manutenção, como podemos observar na Figura 26: validação em uso no ciclo de vida do software.

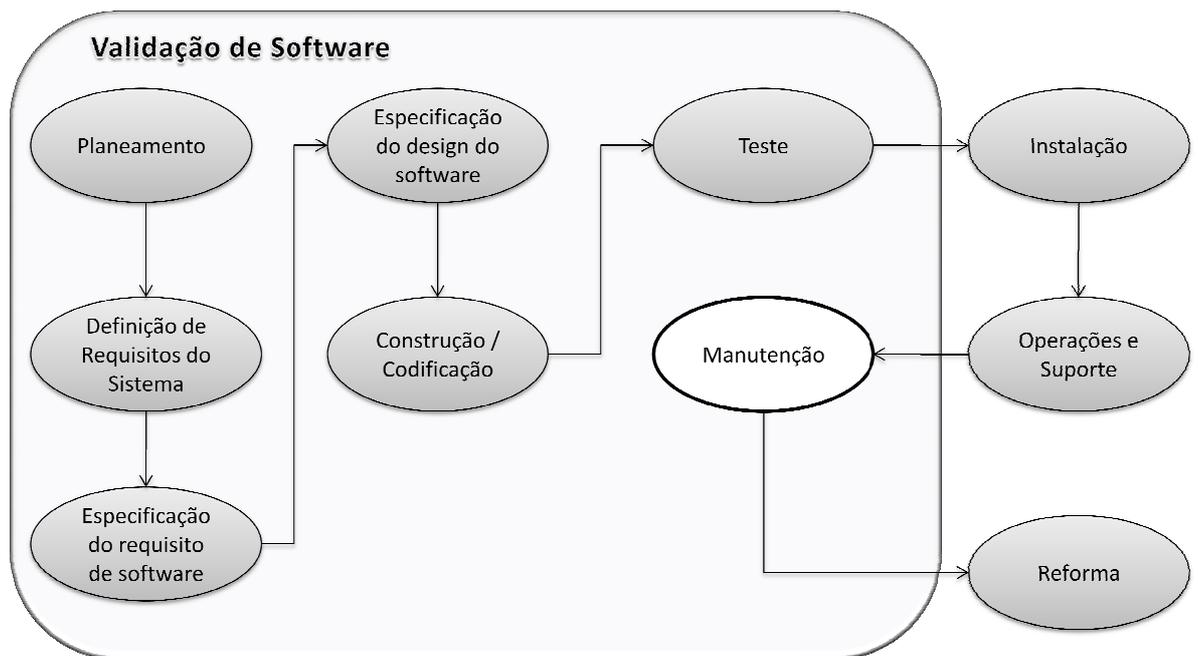


Figura 26: validação em uso no ciclo de vida do software

O processo de validação, executado na fase de manutenção, tem por vista a garantia da sua qualidade em uso, ou seja, espera-se uma certificação de que o sistema funciona tal como seria esperado, com o objectivo de satisfazer as necessidades dos seus utilizadores. Durante esta etapa, deverá ser efectuada a revisão do plano de validação anteriormente executado. Caso este documento não exista, os procedimentos a adoptar no que concerne a documentos de apoio, deverão ser os mesmos referidos na secção relativa à validação interna – deverá ser criado um plano de validação, um documento de especificação de teste, relatório de testes e relatório de validação.

Dentro da validação na fase de manutenção, podemos distinguir ainda duas situações: a primeira validação efectuada ao sistema, do lado do utilizador, e a validação efectuada também pelo utilizador, mas que não é a primeira, faz já parte do processo de manutenção. Nesta situação, impõe-se uma revisão do plano de validação, e identificação de problemas e situações críticas. Espera-se também que seja feita a avaliação acerca das anomalias encontradas anteriormente e sua evolução, bem como o avanço de propostas de mudança, caso tal se

verifique necessário. Estas questões estão implícitas nas métricas abaixo apresentadas.

Antes de prosseguir, resta ainda mencionar que é recomendado que este processo seja executado por uma equipa de validação externa, ou seja, por pessoas que não sejam utilizadores directos do sistema. Tendo em conta a fase do ciclo de vida do produto em que ocorre esta validação, não faria sentido definir *outputs*, como foi feito para os dois casos anteriores, uma vez que a validação em uso incide no utilizador e na sua experiência com o produto final.

Tal como aconteceu para as etapas anteriores, também a validação de um sistema de gestão documental em uso deverá ser efectuada com recurso à aplicação de um conjunto de métricas, definidas na norma ISO/IEC 9126, conforme podemos observar no diagrama abaixo.

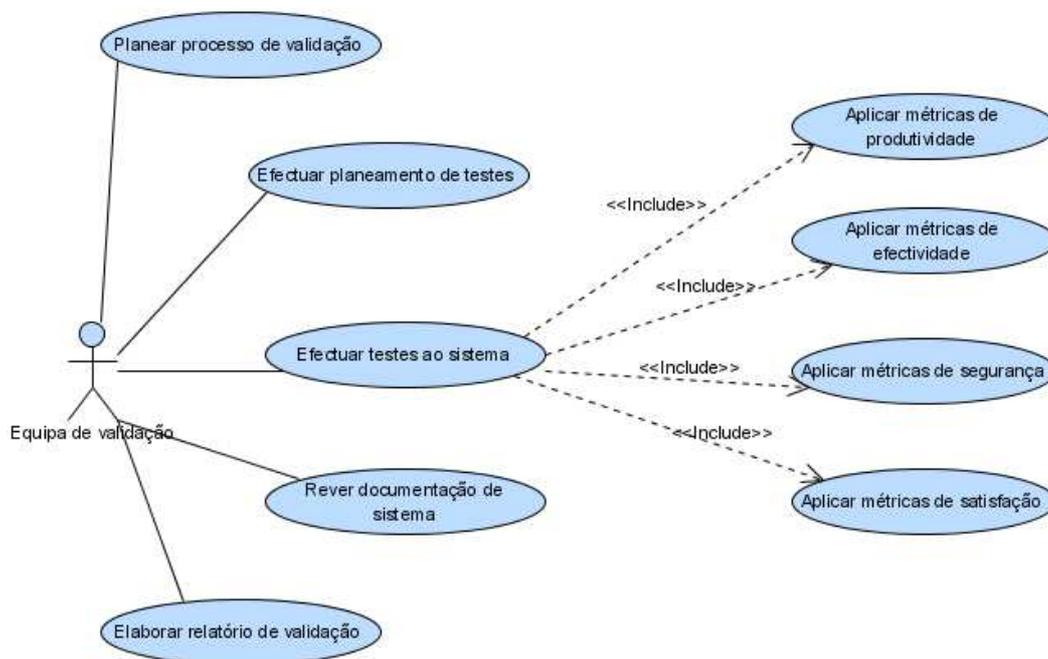


Figura 27: validação em uso

As métricas de **efectividade** têm como propósito garantir que as tarefas executadas pelos utilizadores atingem, de facto, os objectivos pretendidos.

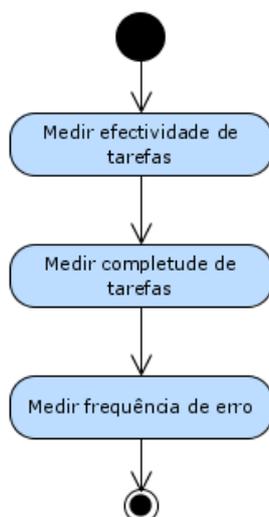


Figura 28: métricas de efectividade

Todas as componentes do diagrama acima apresentado se prendem directamente com uma das características fundamentais de um sistema de gestão documental: a definição de um fluxo de trabalho. Através destas medições, pretende-se avaliar em que medida é que a implementação deste fluxo de trabalho corresponde ao definido no documento de URS. Adicionalmente, a primeira métrica – efectividade de tarefas – deverá também ser aplicada aos fluxos de comunicação, não só entre diferentes utilizadores de um mesmo sistema, mas também entre sistemas colaborativos. Como foi já descrito neste documento, uma das características dos sistemas de gestão documental é a possibilidade de interagir com outros sistemas. A validação da efectividade de comunicação entre estes é, portanto, essencial para a validação do sistema de um modo geral.

A avaliação da **produtividade**, por sua vez, tem por objectivo fazer a relação entre recursos x efectividade, medindo, para tal, aspectos como o tempo de execução de tarefas, da forma que podemos observar na tabela em anexo.

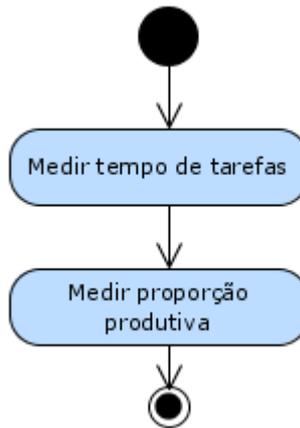


Figura 29: métricas de produtividade

A medição do tempo de tarefas e proporção produtiva têm impacto directo na validação dos tempos de execução do sistema, que, conforme pudemos observar no capítulo anterior, são um dos aspectos importantes a considerar quando se está a validar um sistema de gestão documental. De facto, não faria sentido utilizar este tipo de sistema se os seus tempos de execução não fossem satisfatórios. A medição do tempo de tarefas deverá ainda ser aplicada aos tempos de comunicação, não só a nível interno, mas também no que diz respeito à interoperabilidade entre sistemas. Esta medição tem como propósito validar se o sistema respeita os tempos definidos na especificação de requisitos, e analisar a sua eficiência no âmbito geral.

As métricas de **segurança** tem como função medir os riscos subjacentes à utilização do sistema, sejam eles para com as pessoas, para com o negócio, ou até para com o próprio sistema, dependendo do ambiente em que se encontre inserido.



Figura 30: métricas de segurança

A medição da segurança das pessoas afectadas pela utilização do sistema, prende-se com duas das necessidades de validação identificadas no capítulo anterior: controlo de acessos e confidencialidade. A aplicação desta métrica deverá medir a segurança nestas duas vertentes, garantindo que os acessos ao sistema são correctamente controlados, e que, com isto, a confidencialidade dos documentos manuseados é mantida, uma vez que apenas utilizadores com as permissões adequadas poderão realizar operações, quer sejam apenas de consulta ou também de modificação, sobre estes mesmos documentos.

A medição de danos de software, por sua vez, relaciona-se com a validação da integridade do sistema, procurando garantir, através da aplicação da métrica definida na norma ISO/IEC 9126, a integridade das diferentes componentes do sistema analisadas, conforme os requisitos inicialmente especificados.

Para terminar, a validação em uso aborda ainda a questão **da satisfação**, ou seja, deverão ser aplicadas métricas para garantir que os níveis de satisfação proporcionada pela utilização do sistema estão de acordo com o pretendido.

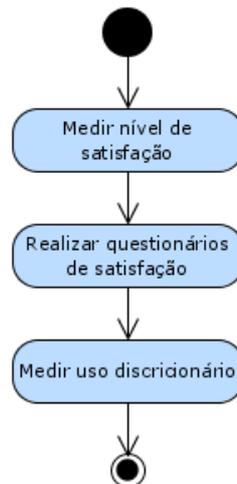


Figura 31: métricas de satisfação

A questão da validação da satisfação não foi abordada anteriormente para o contexto específico dos sistemas de gestão documental, uma vez que esta componente existe em qualquer tipo de sistema, e não apenas neste caso específico. No entanto, e sendo esta validação efectuada quando o sistema está já em uso, pressupõe-se que para que este seja considerado validado é necessário que os utilizadores se sintam satisfeitos com o mesmo, e o vejam como uma mais-valia para o seu trabalho, daí a inclusão, neste modelo, de métricas determinantes da satisfação do utilizador típico do sistema.

4.5. Modelo conceptual de validação – conclusões

Conforme pudemos depreender pela análise das secções anteriores, é possível definir, tendo em conta o ciclo de vida de um produto de software, um modelo de validação composto por três módulos distintos: o primeiro aquando da sua codificação, o segundo aquando do seu teste, e o último aquando da sua manutenção. Para cada um destes momentos as especificidades do produto serão diferentes, bem como os *outputs* sobre

os quais incide o processo de validação. Tendo por base a norma ISO/IEC, e atendendo às funcionalidades características dos sistemas de gestão documental, foi possível identificar um conjunto de métricas apropriadas para suprir as necessidades de validação anteriormente identificadas. A aplicação do modelo apresentado nas secções anteriores, na sua totalidade ou em parte, bem como das referidas métricas, resultará então na validação do sistema de gestão documental em análise.

5. Conclusão

O presente documento tinha como propósito estudar a problemática da validação de sistemas de gestão documental.

Inicialmente, foi feita uma introdução ao tema da validação, em linhas gerais. Foi então possível concluir que a validação é essencialmente o processo de garantir que um sistema faz aquilo para o que foi construído, de forma a satisfazer os requisitos para ele definidos.

Seguiu-se a abordagem à temática dos sistemas de gestão documental, concluindo-se deste capítulo que o tratamento e gestão da informação através de meios informáticos - os sistemas de gestão documental - são uma mais-valia para as organizações actuais, uma vez que a informação é cada vez mais valiosa nos dias que correm. Foi também possível concluir que a validação deste tipo de sistema constituiria, sem sombra de dúvida, uma mais-valia para as entidades que o utilizam, sendo que, apesar disso, poucas são as que realmente o fazem. Esta questão apontava para outro problema: a escassez de informação acerca da validação de sistemas de gestão documental, e a inexistência de um modelo global para este procedimento.

É então, como consequência desse facto, que surge o capítulo 4 do presente documento, dedicado à elaboração do modelo conceptual de validação de sistemas de gestão documental. Como base para este, foi utilizada a norma ISO/IEC 9126, responsável pela regulação da qualidade do software. Interligando a informação contida na referida norma, com a informação anteriormente estudada acerca do posicionamento da validação no ciclo de vida de um produto de software, foi possível identificar uma ligação entre este e os tipos de qualidade definidos na referida norma, levando à identificação de três tipos distintos de validação

- interna, externa e em uso - que, no seu conjunto, compõem o modelo de validação.

Foram então definidas as etapas do ciclo de vida do produto nas quais pode ser efectuada a validação – a fase de construção, a fase de teste e a fase de manutenção – cada uma com um contexto de utilização específico, e podendo ser aplicadas em conjunto ou separado.

De seguida, foram identificados os *outputs* inerentes a cada uma destas etapas, de modo a compreender sobre que artefactos incidiria o nosso modelo. Para além disto, foi ainda necessário definir qual o conjunto de documentação que seria necessário produzir, no âmbito da validação de sistemas de gestão documental, para que o processo fosse o mais completo possível. Foi então definido o seguinte conjunto de actividades de validação:

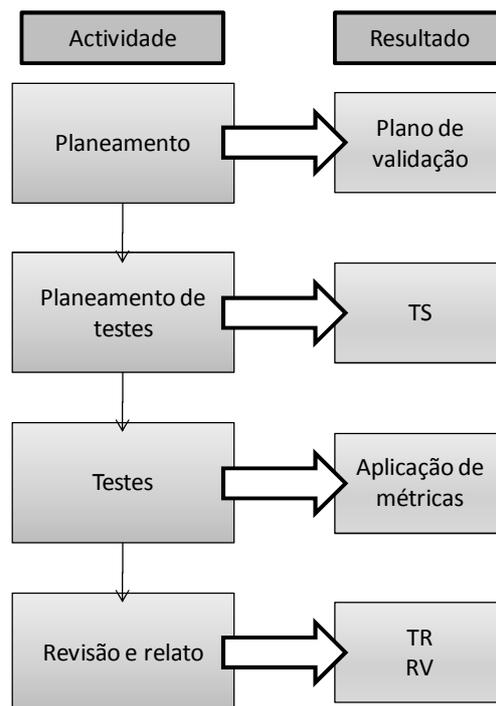


Figura 32: actividades de validação

Dentro da actividade de testes, e com base na norma ISO/IEC 9126, foi definido um conjunto de métricas a aplicar sobre os *outputs* acima referidos, com o propósito de validar o sistema num conjunto de

componentes. Uma vez que o propósito deste modelo é a sua aplicação a sistemas de gestão documental, e não a um sistema em particular, não foi possível delimitar com clareza o alcance da validação. De facto, este baseia-se na complexidade do software ao qual se aplica, e, conforme vimos no capítulo dedicado à gestão documental, os sistemas criados com este propósito poderão ter níveis de complexidade bastante díspares. Assim, o modelo aqui definido inclui métricas que proporcionam uma forma de avaliar todas as componentes que um sistema de gestão documental pode possuir, sendo que, para cada caso específico, deverão ser adaptadas. Dado que este modelo se baseia numa norma de qualidade de software, a sua utilização para validação de outros tipos de sistema não é, de todo, inviável, desde que estas medidas sejam também devidamente adaptadas às suas especificidades.

No que diz respeito a perspectivas de trabalho futuro acerca do tema abordado, a questão mais pertinente reside na aplicação do modelo num ou mais cenários específicos. De facto, um dos pontos que não foi abordado neste documento, mas que teria todo o interesse em ser explorado, seria a aplicação prática do referido modelo, nomeadamente do seu terceiro módulo, respeitante à qualidade em uso. A análise desta aplicação prática teria como propósito verificar se o modelo em questão foca realmente todos os aspectos essenciais no que diz respeito à validação de sistemas de gestão documental, bem como identificar possíveis lacunas, e suas causas.

Portanto, e para concluir, resta referir que o modelo aqui apresentado cumpre os princípios de validação, já que nele são aplicadas métricas que procuram medir a satisfação de **requisitos**, bem como **prevenir problemas futuros**. É também defendido aqui o **planeamento** e documentação do **tempo e esforço** necessários, bem como a definição de *milestones* e atribuição de **responsabilidades**. No que diz respeito ao **ciclo de vida do software**, foi definido que a validação poderia ocorrer em três fases: codificação, teste e manutenção,

sendo que esta última deverá repetir-se após **alterações ao software**, e idealmente efectuada por uma **equipa de revisão independente**. Para terminar, é referido que o modelo é abrangente e **flexível**, podendo ser posteriormente adaptado com base na complexidade do sistema de gestão documental ao qual será aplicado.

Referências

Andrade, M. V. M.(2002). Gerenciamento eletrônico da informação: ferramenta para a gerência eficiente dos processos de trabalho. Em: Seminário Nacional de Bibliotecas Universitárias, 12, 2002, Recife. Anais. Universidade Federal de Pernambuco (UFPE), 2002.

Arnhold, N., S. Delplace, *et al.* (2004). Shared 'Dublin' descriptors for Short Cycle, First Cycle, Second Cycle and Third Cycle Awards. Dublin.

Berndtsson, M., J. Hansson, *et al.* (2008). Thesis Projects. A guide for students in computer science and information systems. Springer.

Bial (2007). Testes Formais a Sistemas Computorizados: 5.

Brasil, C. e-ARQ - Modelo de requisitos para sistemas informatizados de gestão arquivística de documentos - Versão Preliminar para Avaliação. Rio de Janeiro: Conarq, 2006. Acedido em 30-10-2006, 2006, a partir de <http://www.arquivonacional.gov.br>

Buckland, M. K. (1997). "What is a 'document'?" Journal of the American Society for Information Science 48(9): 5.

Calderon, W. R., J. M. Cornelsen, *et al.* (2004). "O processo de gestão documental e da informação arquivística no ambiente universitário." Ci. Inf. 33(3): 7.

Carpenedo, G., M. A. R. Chaves, *et al.* Sistemas de Informação Colaborativos.

Chen, X. H., M. M. M. Snyman, *et al.* (2007). Interrelationship between document management, information management and knowledge management. openUP.

Cláudio, L. and W. A. Sousa (2005). Gestão documental e informação. Brasília, Universidade de Brasília: 6.

Connor, C. and A. S. Louie (2007). IBM SCOREs In Life Science. Health Industry Insights: 16.

Craine, K. (n.d.). "Characteristics of the process." Designing a Document Management Strategy Retrieved 02-09-2009, 2009, from http://www.acom.com/document_management/articles/design_document_strategy_1.html.

Davenport, T. H. (2000). Ecologia da informação: porque só a tecnologia não basta para o sucesso na era da informação, Futura.

Dourish, P., Edwards W. K., *et al.* (2000). Extending Document Management Systems with user-Specific Active Properties. *ACM Transactions on Information Systems*, Vol. 18, No. 2, April 2000, 140-170.

Duarte, K. C. and R. d. A. Falbo Uma Ontologia de Qualidade de Software. Departamento de Informática, UFES. Mestrado: 11.

FDA (2002). General Principles of Software Validation; Final Guidance for Industry and FDA Staff. CDRH: 47.

FDA (2002). General Principles of Software Validation; Final Guidance for Industry and FDA Staff. F. a. D. Administration: 47.

Finlay, P. (1994) *Introducing decision support systems*. Manchester: Blackwell.

Foo, J. (2003). DOCPLAYER - Design Insights from Applying the Non-Hierarchical Media-Player model to Document Management. Department of Computer and Information Science, Linköpings universitet: 75.

Furtado, J. S. (1982). "Informação e organização." *Ciência da Informação* 11(1): 6.

Herrera, A. H. (1993). *Archivística general: teoría y práctica*. Sevilla.

Hetzl, W. C. (1988). *Complete Guide to Software Testing*.

Holos (2007). *Gestão documental, Holos - Soluções avançadas em tecnologias de informação*: 2.

IBM (2005). *Beyond enterprise content management: a compliance-centric architecture*, IBM Healthcare and Life Sciences.

IBM (2006). *IBM Solution for Compliance in a Regulated Environment (SCORE) User's Guide Version 5.1.3* International Business Machines Corporation: 250.

IBM. "IBM SCORE for regulatory compliance." Acedido em 02-01-2009, 2009, a partir de <http://www03.ibm.com/industries/healthcare/us/detail/landing/V302813W72332S02.html>.

IEC 82045-1: 2001 *Gestão de documentos – Parte 1: Princípios e métodos*.

ISO/IEC 9126-1: 2001. *Software engineering–Product quality- Part 1: Quality model*.

ISO/IEC 9126-3: 2003. *Software engineering– Product quality- Part 3: External Metrics*

ISO/IEC 9126-3: 2003. *Software engineering– Product quality- Part 3: Internal Metrics*

ISO/IEC 9126-4: 2004. *Software engineering– Product quality- Part 4: Quality in use metrics*.

- Jardim, J. M. (n.d.) O Conceito e a Prática de Gestão de Documentos. Volume, DOI:
- Jecan, S. (2008). "Document Management vs. Knowledge Management." *Informática Económica* 4(48): 4.
- Joomla (2007) IBM desenvolve projecto para a BIAL. *Computerworld Portugal*, 1.
- Liu, Z. (2004). "The evolution of documents and its impacts. ." *Journal of Documentation* 60(3): 9.
- Lucca, G. (2007). *Plonarq: Gerenciamento Eletrônico de documentos arquivísticos baseado em software livre*. Centro de Tecnologia. Santa Maria, RS, Brasil, Universidade Federal de Santa Maria: 92.
- Machado, M. P. and S. F. Souza (n.d.). *Métricas e Qualidade de Software*. Departamento de informática, Universidade Federal do Espírito Santo. Mestrado.
- Machado, R. J. (2008). Unidade(s) Curriculare(s) de Dissertação - ano lectivo 2008/09. Guimarães, Universidade do Minho: 11.
- Marques, A. B. (2006). *Sistemas de Informação Computadorizados (Introdução e Conceitos Básicos)*. Rio de Janeiro, CEFET/RJ.
- Marques, P., R. Rodrigues, *et al.* (2007). *Validação de Sistemas Computorizados na Indústria Farmacêutica – O Caso BIAL*. CAPSI 2007. Aveiro.
- McDowall, R. D. (2005). *Writing the User Requirements Specification. Validation of Chromatography Data Systems: Meeting Business and Regulatory Requirements*, Royal Society of Chemistry, The. 1: 259.
- Mehedintu, A., C. Pîrvu, *et al.* (2007). "Using Electronic Systems for Document Management in Economic Entities." *Informática Económica* 1(41): 7.
- Muegge, E. (2007) *Developing and Documenting Quality Test Scripts*. 105.
- Myers, G. J. (1979). *The art of software testing*. New York
- Neal, C. (2003). "Prerequisites for Successful Validation." *Journal of Validation Technology* 9(3): 6.
- O`Brien, J. A. (2001). *Sistemas de Informação e as decisões gerenciais na era da internet*.
- Odegaard, S. (2006). *10 Steps to SW Validation, Process Pro*: 4.
- Omar, M. B. (2005). *Felda Document Management System*. Computer science and information systems, Universiti Teknologi Malaysia: 237.

- Parreira, J. and M. Teixeira (2004). Porquê Gestão Documental? Dalara Ciberguia: 20.
- Paz, A. Validação de Sistemas Computadorizados. Sociedade Brasileira de Controle de Contaminação: 8.
- Reid, C. and S. Strause (2007). Risk Based Approach to Computerised System Validation. Integrity Solutions: 102.
- Rodrigues, R., P. Marques, *et al.* (2007). Validação de Sistemas Computadorizados na Indústria Farmacêutica: O caso BIAL. CAPSI 2007. Guimarães, Trofa, Bial, Universidade do Minho: 11.
- Sarmiento, A. M. T. (2002). Estudo de Casos de Adopção e Utilização de Sistemas Workflow. Departamento de Sistemas de Informação. Guimarães, Universidade do Minho: 417.
- Schousboe, M. (2005). Computer Validation Master Planning "Validation Strategies". Pharmaceutical Technology: 12.
- Stokes, T., The Survive and Thrive Guide to Computer Validation, Interpharm Press, 1998.
- Tran, E. (1999). Verification/Validation/Certification. Dependable Embedded Systems, Carnegie Mellon University
- Turban, E. (1995). Decision Support System and Expert Systems. Englewood Cliffs. New Jersey.
- Zachary, W. and M. Weiland (1994). Interface Agents for Effective Human-Computer Coordination in Hybrid Automation Systems. Advances in Agile Manufacturing. W. Karwowski. Amsterdão: 313-316.
- Zwirtes, C. L. and D. G. Durante (2007). Gestão documental: atuação do secretário executivo. Secretariado Executivo em Revist@: 80.

Anexos

Em anexo ao presente documento, segue um conjunto de tabelas, descritivas das métricas apresentadas no capítulo 4 da presente publicação, conforme constam na documentação referente à norma ISO/IEC 9126. Às tabelas originais foi acrescentada uma última coluna, indicando sobre quais dos outputs incide a referida métrica. De salientar também que as tabelas abaixo não contém todas as métricas apresentadas na referida norma, mas apenas aquelas que, na sua especificidade, se enquadram no contexto dos sistemas de gestão documental e suas necessidades de validação.

Métricas de validação durante a fase de desenvolvimento de SGD

Funcionalidade					
Adequação					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adequação funcional	Verificar se as funções analisadas são adequadas.	Para o conjunto de funções implementadas com o propósito de executar uma determinada tarefa, medir o rácio entre aquelas nas quais foram detectadas problemas e o número global.	$X = 1 - A/B$ A = Número de funções com problema B = Número de funções verificadas	$0 \leq X \leq 1$ O valor obtido será tanto mais favorável quanto mais próximo de 1, significando este uma adequação perfeita.	URS FRS Código-fonte
Complitude da implementação funcional	Verificar se a implementação funcional está	Medir o rácio entre as funções não implementadas e o conjunto de funções descritas	$X = 1 - A/B$ A = Número de	$0 \leq X \leq 1$ O valor obtido será tanto mais	URS DRS

	completa.	na especificação de requisitos.	funções em falta B = Número de funções descritas na URS	completo, quanto mais próximo de 1.	Código-fonte
Cobertura da implementação funcional	Verificar o quão correcta é a implementação funcional.	Comparar o número de funções em falta, ou implementadas de forma incorrecta, e compará-lo com o número de funções definidas na especificação de requisitos.	$X = 1 - A/B$ A = Número de funções em falta ou incorrectas B = Número de funções descritas na especificação de requisitos	$0 \leq X \leq 1$ O valor obtido será tanto mais correcto, quanto mais próximo de 1.	URS DRS Código-fonte
Estabilidade da especificação funcional	Verificar se a especificação funcional é estável ao longo do ciclo de desenvolvimento do produto.	Comparar o número de funções inicialmente definidas na especificação de requisitos com o número de funções que sofreram qualquer tipo de alteração até à fase de codificação.	$X = 1 - A/B$ A = Número de funções alteradas B = Número de funções descritas na especificação de requisitos	$0 \leq X \leq 1$ Quanto mais próximo de 1 for o valor de X, mais estável é a especificação funcional.	URS

Exactidão

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Exactidão Computacional	Verificar se os requisitos relativos à precisão foram completamente implementados.	Contar o número de funções nas quais foram aplicados os requisitos de precisão e comparar este valor com o número de funções que têm requisitos de precisão associados	$X = A/B$ A = Número de funções nas quais os requisitos foram implementados B = Número de funções com requisitos de precisão específicos	$0 \leq X \leq 1$ O valor obtido será tão mais completo quanto mais próximo de 1.	URS DRS Código-fonte
Precisão	Verificar se os diferentes níveis de precisão para conjuntos de dados foram correctamente implementados.	Contar o número de itens de dados nos quais foram aplicados os níveis de precisão e comparar este valor com o número de itens de dados que têm níveis de precisão associados	$X = A/B$ A = Número de itens de dados aplicados com diferentes níveis de precisão B = Número de itens de dados com níveis de precisão	$0 \leq X \leq 1$ O valor obtido será tão mais completo quanto mais próximo de 1.	URS DRS Código-fonte

			específicos		
Interoperabilidade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Consistência do interface	Verificar se os protocolos do interface foram correctamente implementados.	Comparar o número de protocolos correctamente implementados com o número de protocolos a serem implementados, de acordo com as especificações.	$X = A/B$ A = Número de protocolos correctamente implementados B = Número de protocolos a implementar, conforme especificado	$0 \leq X \leq 1$ O valor obtido será tão mais consistente quanto mais próximo de 1.	URS DRS Código-fonte
Segurança					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Auditabilidade de acesso	Verificar se o login no sistema se encontra de acordo com os padrões definidos.	Contar o número de tipos de acesso que estão a ser conectados correctamente, e comparar este valor com o número de tipos de acesso que são necessários para cumprir os requisitos especificados.	$X = A/B$ A = Número de tipos de acesso a serem conectados correctamente B = Número de tipos de acesso requeridos pela especificação de requisitos	$0 \leq X \leq 1$ Quanto mais próximo estiver de 1, mais correcto estará o valor de X.	URS DRS Código-fonte
Controlo de acesso	Verificar se os acessos ao sistema são controlados.	Comparar o número de requisitos de controlo de acesso efectivamente implementados com o número dos mesmos definido na especificação de requisitos.	$X = A/B$ A = Número de requisitos de controlo de acesso correctamente implementados B = Número de requisitos de controlo de acesso a implementar, conforme especificado	$0 \leq X \leq 1$ Quanto mais próximo de 1 estiver o valor de X, mais fácil será controlar os acessos ao sistema.	URS DRS Código-fonte
Prevenção de corrupção de	Verificar o quão completa é a	Contar o número de instâncias de prevenção de corrupção de	$X = A/B$	$0 \leq X \leq 1$	URS

dados	prevenção de corrupção de dados.	dados correctamente aplicadas, e comparar este valor com o número das mesmas, definido na especificação de requisitos.	<p>A = Número de instâncias de prevenção de corrupção de dados correctamente implementadas</p> <p>B = Número de instâncias de prevenção de corrupção de dados a implementar, conforme especificado</p>	Quanto mais próximo estiver de 1, mais completa será a prevenção de danos.	<p>DRS</p> <p>Código-fonte</p>
-------	----------------------------------	--	--	--	--------------------------------

Funcionalidade

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Conformidade funcional	Verificar se os interfaces estão em conformidade com as regras, normas e convenções legais aplicáveis.	Contar o número de itens que requerem conformidade legal que foram implementados, e comparar este valor com o número de itens que requerem conformidade legal definidos na especificação.	<p>$X = A/B$</p> <p>A = Número de itens que requerem conformidade legal que foram implementados</p> <p>B = Número de itens que requerem conformidade legal definidos na especificação</p>	<p>$0 \leq X \leq 1$</p> <p>Quanto mais próximo X estiver de 1, maior conformidade funcional existirá.</p>	<p>DRS</p> <p>Código-fonte</p>

Confiabilidade

Maturidade

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Detecção de falhas	Contabilizar o número de falhas detectadas durante a revisão do produto.	Contar o número de falhas detectadas durante a revisão do produto, e comparar com o valor estimado.	<p>$X = A/B$</p> <p>A = Número de falhas detectadas aquando da revisão do produto</p> <p>B = Número de falhas estimadas</p>	<p>$0 \leq X$</p> <p>Um valor de X reduzido significa qualidade de produto. No entanto, $X = 0$ não indica necessariamente a ausência de falha.</p>	Plano

Remoção de falhas	Contabilizar quantas falhas foram corrigidas, e qual a proporção de falhas removidas.	Comparar o número de falhas removidas aquando do design com o número de falhas detectadas durante o mesmo.	$X = A$ A = Número de falhas corrigidas durante o design. $Y = A/B$ B = Número de falhas detectadas	$0 \leq X$ Quanto mais elevado for o valor de X, menos falhas se verificarão ainda no sistema. $0 \leq Y \leq 1$ Quanto mais próximo o valor de Y estiver de 1, mais falhas terão sido removidas.	Plano
Adequação dos testes	Verificar quantos casos de teste estavam previstos no plano de testes.	Comparar o número de testes previstos no plano inicial com o número de testes realmente necessários para abranger todos os aspectos necessários.	$X = A/B$ A = Número de testes definidos no plano de teste B = Número de casos de teste necessários	$0 \leq X$ Quanto maior for o valor de X, mais adequado será o plano de teste.	Plano URS

Tolerância a falhas

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Evitação de falhas	Contabilizar quantos padrões de falha foram analisados com vista a evitar erros críticos.	Comparar o número de padrões de falha evitados com o número de padrões de falha considerados.	$X = A/B$ A = Número de padrões de falha evitados durante a codificação B = Número de padrões de falha considerados	$0 \leq X$ Quanto mais elevado for o valor de X, maior será a evitação de falhas.	URS
Evitação de operações incorrectas	Verificar qual o número de funções implementadas que têm capacidade de evitar operações incorrectas.	Comparar o número de funções implementadas para evitar erros críticos com o número de padrões de falha de operações considerados.	$X = A/B$ A = Número de funções implementadas para evitar padrões de falha B = Número de padrões de falha de operações considerados	$0 \leq X$ Quanto mais elevado for o valor de X, maior será a evitação de operações incorrectas.	URS

Recuperabilidade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Capacidade de Restauro	Medir a capacidade do produto se restaurar após um evento inesperado.	Comparar o número de requisitos de restauro efectivamente implementados com o valor previsto na especificação de requisitos.	$X = A/B$ A = Número de requisitos de restauro implementados B = Número de requisitos de restauro definidos na especificação de requisitos	$0 \leq X \leq 1$ Quanto maior for o valor de X, maior será a capacidade de restauro do sistema.	URS
Efectividade de restauro	Verificar até que ponto é que a capacidade de restauro é eficaz.	Após efectuar simulações, comparar o número de requisitos de restauro implementados que cumprem os valores ideais com os previstos na especificação de requisitos.	$X = A/B$ A = Número de requisitos de restauro implementados que cumprem os tempos ideais B = Número de requisitos de restauro definidos na especificação de requisitos, com os tempos ideais	$0 \leq X \leq 1$ Quanto maior for o valor de X, maior será a capacidade efectividade de restauro do sistema.	URS DRS
Confiabilidade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adequação da confiabilidade	Verificar se a fiabilidade do produto está em conformidade com as regras, normas e convenções legais aplicáveis.	Contar o número de itens que requerem conformidade legal que foram implementados, e comparar este valor com o número de itens que requerem conformidade legal definidos na especificação.	$X = A/B$ A = Número de itens que requerem conformidade legal que foram implementados B = Número de itens que requerem conformidade legal definidos na especificação	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior conformidade de fiabilidade existirá.	DRS Código-fonte

Usabilidade

Compreensão

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Complitude da descrição	Verificar qual a proporção de funções descritas na descrição de produto	Comparar o número de funções correctamente descritas com o número total de funções no produto.	$X = A/B$ A = Número de funções correctamente descritas B = Número de funções do produto	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, mais completa será a descrição	URS DRS
Capacidade de demonstração	Verificar qual a proporção de funções que requerem demonstração que são efectivamente demonstráveis.	Comparar o número de funções que são efectivamente demonstráveis com o número de funções que requerem capacidade de demonstração.	$X = A/B$ A = Número de funções cuja capacidade de demonstração foi confirmada B = Número total de funções que requerem demonstração	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior será a capacidade de demonstração	URS DRS
Funções evidentes	Verificar qual a proporção de funções do produto que são evidentes para o utilizador.	Comparar o número de funções evidentes para o utilizador com o número total de funções do produto.	$X = A/B$ A = Número de funções evidentes para o utilizador B = Número de funções totais do produto	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor.	URS DRS
Compreensão das funções	Verificar qual a proporção das funções do produto que o utilizador terá capacidade de entender correctamente.	Comparar o número de funções presentes no interface com o utilizador cujo propósito é claramente compreendido, com o número total de funções do mesmo interface.	$X = A/B$ A = Número de funções de interface compreendidas pelo utilizador B = Número total de funções de interface	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor será a compreensão de funções.	URS DRS

Aprendizagem

Nome da	Objectivo	Método de aplicação	Medida	Interpretação dos	Output ao qual
---------	-----------	---------------------	--------	-------------------	----------------

métrica				valores obtidos	se aplica
Complitude da documentação para o utilizador e/ou meios de ajuda	Verificar qual a proporção de funções abrangidas pela documentação de ajuda ao utilizador.	Contar o número de funções abrangidas pela documentação para o utilizador e/ou meios de ajuda, e comparar este valor com o número total de funções do produto.	$X = A/B$ A = Número de funções descritas na documentação de apoio ao utilizador B = Número total de funções	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, mais completa será a documentação de apoio.	URS DRS

Operabilidade

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Verificação da validade do <i>Input</i>	Verificar qual a proporção de dados de <i>Input</i> que fornecem a possibilidade de verificar a validade dos dados.	Comparar o número de itens de <i>Input</i> que verificam a validade dos dados com o número global de itens de <i>Input</i> que o poderiam fazer	$X = A/B$ A = Número de itens de <i>Input</i> que verificam a validade dos dados B = Número de itens de <i>Input</i> que poderiam verificar a validade dos dados	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor.	URS DRS
Possibilidade de cancelar operações do utilizador	Verificar qual a proporção de funções que podem ser canceladas antes de estarem completas.	Contar o número de funções implementadas que podem ser canceladas pelo utilizador antes de estarem completas, e comparar esse valor com o número de funções que requerem esta capacidade.	$X = A/B$ A = Número de funções implementadas que podem ser canceladas antes de estarem completas B = Número total de funções que requerem a capacidade de cancelamento antes de estarem completas.	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor a capacidade de cancelamento	URS DRS
Anulabilidade das operações do utilizador	Verificar qual a proporção de funções que podem ser anuladas.	Comparar o número de funções que podem ser anuladas após estarem completas com o número total de funções.	$X = A/B$ A = Número de funções que podem ser anuladas depois de completas B = Número total	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior a capacidade de anulação.	URS DRS

			de funções		
Personalização	Verificar qual a proporção de funções que podem ser personalizadas durante a sua operação.	Contar o número de funções que podem ser personalizadas pelo utilizador durante a sua operação, e comparar este valor com o número de funções que requerem esta capacidade.	$X = A/B$ A = Número de funções que podem ser personalizadas pelo utilizador enquanto estão a ser executadas. B = Número total de funções que requerem capacidade de personalização	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior a capacidade de personalização.	URS DRS
Acessibilidade física	Verificar qual a proporção de funções que podem ser personalizadas para utilizadores portadores de deficiências físicas.	Comparar o número de funções que podem ser personalizáveis para utilizadores portadores de deficiências físicas com o número total de funções	$X = A/B$ A = Número de funções personalizáveis para utilizadores com deficiências B = Número total de funções	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor será a acessibilidade física.	URS DRS
Capacidade de monitorizar o estado de operações	Verificar qual a proporção de funções que oferecem a possibilidade de monitorizar o seu estado.	Comparar o número de funções que possibilitam a monitorização do seu estado com o número total de funções que requerem a capacidade de monitorização	$X = A/B$ A = Número de funções que possibilitam a monitorização do seu estado B = Número total de funções que requerem capacidade de monitorização	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor será a capacidade de monitorização	URS DRS
Consistência operacional	Verificar qual a proporção de funções que têm um comportamento semelhante a funções similares presentes noutras partes do sistema.	Contar o número de instâncias de operações com comportamento inconsistente, e comparar este valor com o número total de operações.	$X = A/B$ A = Número de instâncias de operações com comportamento inconsistente B = Número total de operações	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, mais consistente é o sistema	URS DRS
Clareza de	Verificar qual a proporção de	Comparar o número de mensagens implementadas que	$X = A/B$	$0 \leq X \leq 1$	URS

mensagens	mensagens se auto explicam.	são claras para o utilizador com o número total de mensagens implementadas	A = Número de mensagens implementadas com explicações claras B = Número total de mensagens implementadas	Quanto mais próximo X estiver de 1, mais claras serão as mensagens	DRS
Clareza dos elementos do interface	Verificar qual a proporção de elementos do interface que se auto explicam.	Comparar o número de elementos do interface implementados que são claros para o utilizador com o número total de elementos do interface.	$X = A/B$ A = Número de elementos do interface implementados com explicações claras B = Número total de elementos do interface	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, mais clareza	URS DRS
Recuperabilidade de erros operacionais	Verificar que proporção das funções implementadas pode tolerar erros do utilizador.	Comparar o número de funções implementadas com tolerância a erros do utilizador com o número de funções que requerem esta capacidade.	$X = A/B$ A = Número de funções implementadas com tolerância a erros do utilizador B = Número total de funções que requerem a capacidade de tolerar erros do utilizador	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior será a recuperabilidade de erros operacionais.	URS DRS

Atractividade

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Interacção atractiva	Verificar o quão atractivo é o interface para o utilizador	Realização de questionários junto dos utilizadores.	Questionários vocacionados para o aspecto gráfico do produto, realizados junto dos utilizadores.	Classificação da avaliação.	URS DRS
Personalização da aparência do interface de utilizador	Verificar qual a proporção de elementos do interface que podem ser personalizados pelo	Inspeção.	$X = A/B$ A = Número de elementos do interface que podem ser	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, mais personalizável será o	URS DRS

	utilizador.		personalizados B = Número total de elementos do interface	interface.	
Usabilidade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adequação da usabilidade	Verificar se a usabilidade do produto está em conformidade com as regras, normas e convenções legais aplicáveis.	Contar o número de itens que requerem conformidade legal que foram implementados, e comparar este valor com o número de itens que requerem conformidade legal definidos na especificação.	$X = A/B$ A = Número de itens que requerem conformidade legal que foram implementados B = Número de itens que requerem conformidade legal definidos na especificação	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior conformidade de usabilidade existirá.	DRS Código-fonte

Eficiência					
Temporal					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Tempo de resposta	Definir o tempo estimado para completar uma tarefa.	Avaliar a eficiência do sistema operativo e tempos de resposta do sistema. Este valor pode ser calculado ou simulado.	$X = \text{Tempo}$	Quanto menor for o valor de X, melhor será o tempo de resposta	Código-fonte
Rendimento por unidade temporal	Definir o número estimado de tarefas que podem ser executadas por unidade temporal.	Avaliar a eficiência do tratamento dos recursos no sistema. Fazer um factor baseado nas chamadas de aplicação para o sistema de manuseamento dos recursos.	$X = \text{Número de tarefas por unidade de tempo}$	Quanto maior for o valor de X, melhor será o rendimento do sistema.	Código-fonte
Tempo de viragem	Definir o tempo estimado para a completar um conjunto de tarefas relacionadas entre si.	Avaliar a eficiência do sistema operativo e tempos de resposta do sistema. Este valor pode ser calculado ou simulado.	$X = \text{Tempo}$	Quanto menor for o valor de X, melhor o rendimento do sistema.	Código-fonte

Utilização de recursos					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Utilização I/O	Verificar qual a utilização de I/O estimada para completar uma dada tarefa.	Estimar os requisitos de utilização de I/O para a aplicação. Este valor pode ser calculado ou simulado.	X = Número de buffers	Quanto menor for o valor de X, mais eficiente será o sistema.	Código-fonte
Densidade de mensagens de utilização I/O	Verificar qual a densidade de mensagens relacionadas com a utilização de I/O nas linhas de código responsáveis pelas chamadas de sistema.	Contar o número de erros devidos a falhas e avisos de I/O e comparar com o número estimado de linhas de código responsáveis por chamadas de sistema.	X = A/B A = Número de mensagens relacionadas com erros de I/O B = Número de linhas de código directamente relacionadas com chamadas de sistema.	Quanto mais elevado for o valor de X, mais eficiente será o sistema.	Código-fonte
Utilização de memória	Verificar qual o espaço de memória que se estima que o sistema ocupe para a concretização de uma dada tarefa.	Estimar os requisitos de utilização de memória. Este valor pode ser calculado ou simulado.	X = Tamanho em bytes	Quanto menor for o valor de X, mais eficiente será o sistema.	Código-fonte
Densidade de mensagens de utilização de memória	Verificar qual a densidade de mensagens relacionadas com a utilização de memória nas linhas de código responsáveis pelas chamadas de sistema.	Contar o número de erros devidos a falhas e avisos de memória, e comparar com o número estimado de linhas de código responsáveis por chamadas de sistema.	X = A/B A = Número de mensagens relacionadas com erros de I/O B = Número de linhas de código directamente relacionadas com chamadas de sistema.	Quanto mais elevado for o valor de X, mais eficiente será o sistema.	Código-fonte
Eficiência					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adequação da	Verificar se a eficiência do	Contar o número de itens que requerem conformidade legal	X = A/B	0 <= X <= 1	DRS

eficiência	produto está em conformidade com as regras, normas e convenções legais aplicáveis.	que foram implementados, e comparar este valor com o número de itens que requerem conformidade legal definidos na especificação.	A = Número de itens que requerem conformidade legal que foram implementados B = Número de itens que requerem conformidade legal definidos na especificação	Quanto mais próximo X estiver de 1, maior conformidade de eficiência existirá.	Código-fonte
------------	--	--	---	--	--------------

Manutenção

Capacidade de análise

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Registo de actividades	Verificar o quão completo é o registo das actividades do sistema.	Comparar o número de itens registados no log de actividades com o número de itens que requerem esse registo.	$X = A/B$ A = Número de itens efectivamente registados no log de actividades B = Número de itens que deveriam estar registados, segundo as especificações	$0 \leq X \leq 1$ Quanto mais próximo de 1 estiver o valor de X, maior a quantidade de dados disponibilizados para registo.	URS
Capacidade de leitura da função de diagnóstico	Verificar o quão profunda é a prestação de funções de diagnóstico.	Contar o número de funções de diagnóstico implementadas, e comparar esse valor com o número de funções de diagnóstico previstas na especificação de requisitos.	$X = A/B$ A = Número de funções de diagnóstico implementadas B = Número de funções de diagnóstico previstas	$0 \leq X \leq 1$ Quanto mais próximo de 1 estiver o valor de X, melhor será a implementação de funções de diagnóstico	URS

Alterabilidade

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Registo de mudança	Verificar se as alterações são correctamente	Rácio do registo de informações relativas à alteração de módulos do	$X = A/B$ A = Número de	$0 \leq X \leq 1$ Quanto mais	Código-fonte

	armazenadas no código, com linhas de comentário.	produto.	alterações a funções/módulos que possuem comentários relativos à mudança B = Número total de alterações efectuadas	próximo de 1 estiver o valor de X, maior será a capacidade de registo. Um valor nulo indica ou pouco controlo ou poucas alterações.	
--	--	----------	---	---	--

Estabilidade

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Impacto de mudança	Verificar qual a frequência de impactos negativos após alterações ao sistema.	Comparar o número de impactos adversos ocorridos após alterações ao sistema com o número total de alterações.	$X = 1 - A/B$ A = Número de impactos negativos após alterações ao sistema B = Número total de alterações efectuadas	$0 \leq X \leq 1$ Quanto mais próximo de 1 estiver o valor de X, menor será o impacto de mudança.	Código-fonte
Localização do impacto de modificação	Verificar a extensão do impacto de uma modificação no produto de software.	Comparar o número de variáveis afectadas pela alteração com o número total de variáveis no produto.	$X = A/B$ A = Número de variáveis afectadas pela alteração B = Número total de variáveis do produto	$0 \leq X \leq 1$ Quanto mais próximo de 0 estiver o valor de X, menor será o impacto de mudança.	Código-fonte

Capacidade de teste

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Autonomia da testabilidade	Verificar em que medida é que o software pode ser testado de forma independente.	Contar o número de dependências de outros sistemas para testar que foram simulados em esboços e comparar esse valor com o número total de dependências de outros sistemas para teste.	$X = A/B$ A = Número de dependências de outros sistemas para testes, simuladas anteriormente B = Número total de dependência de outros sistemas para teste	$0 \leq X \leq 1$ Quanto mais próximo de 1 estiver o valor de X, mais autónomo será um sistema, no que diz respeito à sua testabilidade.	DRS

Manutenção					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adequação da manutenção	Verificar se a manutenção do produto está em conformidade com as regras, normas e convenções legais aplicáveis.	Contar o número de itens que requerem conformidade legal que foram implementados, e comparar este valor com o número de itens que requerem conformidade legal definidos na especificação.	$X = A/B$ A = Número de itens que requerem conformidade legal que foram implementados B = Número de itens que requerem conformidade legal definidos na especificação	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior conformidade de manutenção existirá.	DRS Código-fonte

Portabilidade					
Adaptabilidade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adaptabilidade de estruturas de dados	Verificar a adaptabilidade do produto a alterações na estrutura de dados.	Contar o número de estruturas de dados que são operáveis e não têm quaisquer limitações após adaptação, e comparar esse valor com o número total de estruturas de dados que requerem capacidade de adaptação.	$X = A/B$ A = Número de estruturas de dados que são operáveis e não tem quaisquer limitações após adaptação B = Número total de estruturas de dados que requerem capacidade de adaptação.	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor a adaptabilidade de estruturas de dados.	URS DRS
Adaptabilidade ao hardware	Verificar a adaptabilidade do produto a alterações na estrutura de hardware.	Comparar o número de funções implementadas que conseguem atingir os resultados esperados em diferentes ambientes de hardware com o número de funções que tem como requisito a adaptação a diferentes ambientes de hardware.	$X = A/B$ A = Número de funções implementadas que conseguem atingir os resultados esperados em diferentes ambientes de	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor a adaptabilidade ao hardware.	URS DRS

			hardware B = Número total de funções que tem como requisito a adaptação a diferentes ambientes de hardware.		
Adaptabilidade ao ambiente organizacional	Verificar a adaptabilidade do produto a alterações no ambiente organizacional.	Comparar o número de funções implementadas que conseguem atingir os resultados esperados em diferentes ambientes organizacionais, com o número de funções que tem como requisito a adaptação a diferentes ambientes organizacionais.	$X = A/B$ A = Número funções implementadas que conseguem atingir os resultados esperados em diferentes ambientes organizacionais B = Número total de funções que tem como requisito a adaptação a diferentes ambientes organizacionais.	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor a adaptabilidade ao ambiente organizacional.	URS DRS
Adaptabilidade ao software	Verificar a adaptabilidade do produto a alterações no ambiente de software.	Comparar o número de funções implementadas que conseguem atingir os resultados esperados em diferentes ambientes de software, com o número de funções que tem como requisito a adaptação a diferentes ambientes de software.	$X = A/B$ A = Número funções implementadas que conseguem atingir os resultados esperados em diferentes ambientes de software B = Número total de funções que tem como requisito a adaptação a diferentes ambientes de software.	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor a adaptabilidade ao ambiente de software.	URS DRS
Capacidade de instalação					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica

Facilidade de reinstalação	Verificar se é fácil repetir a operação de instalação.	Contar o número de operações implementadas para voltar a tentar instalar o sistema, e comparar esse valor com o número de operações necessárias para voltar a instalar o sistema.	$X = A/B$ A = Número de operações implementadas para voltar a tentar instalar o sistema B = Número total de operações necessárias para instalar o sistema.	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, mais fácil será reinstalar o sistema.	Código-fonte
Esforço de instalação	Verificar qual o nível de esforço necessário para instalar o software.	Comparar o número de passos de instalação implementados com o número de passos previstos.	$X = A/B$ A = Número de passos para instalar o sistema efectivamente implementados B = Número total de passos necessários para instalar o sistema.	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, mais fácil será instalar o sistema.	Código-fonte
Flexibilidade de instalação	Verificar o quão flexível e personalizável é a capacidade de instalação do sistema.	Contar o número de operações personalizadas de instalação implementadas, e comparar esse valor com o número total de operações personalizadas de instalação requeridas	$X = A/B$ A = Número de operações personalizadas de instalação implementadas B = Número total de operações personalizadas de instalação requeridas.	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, mais flexível será a instalação do software.	Código-fonte

Coexistência

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Coexistência disponível	Verificar até que ponto é que o produto é capaz de partilhar um conjunto de recursos com outra solução de software, sem prejuízo para qualquer um dos	Comparar o número de entidades com as quais o produto pode coexistir, com o número de entidades no ambiente de produção que requerem essa coexistência.	$X = A/B$ A = Número de entidades com as quais o produto pode coexistir conforme especificado B = Número total	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor será a coexistência.	URS

	dois.		de entidades no ambiente de produção que requerem coexistência.		
Capacidade de substituição					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Uso contínuo de dados	Verificar qual a quantidade de dados que permanece inalterável após substituição do produto.	Contar o número de itens de dados que continuam a ser utilizados após a substituição, e comparar esse valor com o número total de itens de dados antigos que deverão continuar a ser usados depois da substituição do software.	$X = A/B$ A = Número de itens de dados que continuam a ser utilizados após a substituição B = Número total de itens de dados antigos que deverão continuar a ser usados depois da substituição do software.	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor.	DRS Código-fonte
Inalterabilidade de funções	Verificar qual a quantidade de funções que permanece inalterável após substituição do produto.	Contar o número de funções abrangidas pelo novo software que produzem resultados semelhantes, e comparar esse valor com o número total de funções no software antigo.	$X = A/B$ A = Número de funções abrangidas pelo novo software que produzem resultados semelhantes B = Número total de funções no software antigo.	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor.	DRS Código-fonte
Portabilidade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adequação da portabilidade	Verificar se a portabilidade do produto está em conformidade com as regras, normas e convenções legais aplicáveis.	Contar o número de itens que requerem conformidade legal que foram implementados, e comparar este valor com o número de itens que requerem conformidade legal definidos na especificação.	$X = A/B$ A = Número de itens que requerem conformidade legal que foram implementados B = Número de itens que requerem	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior conformidade de portabilidade existirá.	DRS Código-fonte

			conformidade legal definidos na especificação		
--	--	--	---	--	--

Métricas de validação durante a fase de conclusão do desenvolvimento de SGD

Funcionalidade					
Adequação					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adequação funcional	Verificar se as funções analisadas são adequadas.	Comparar o número de funções nas quais foram detectados problemas com o número de funções avaliadas.	$X = 1 - A/B$ A = Número de funções avaliadas nas quais foram detectados problemas B = Número de funções avaliadas	$0 \leq X \leq 1$ O valor obtido será tanto mais favorável quanto mais próximo de 1, significando este uma adequação perfeita.	URS TR
Complitude da implementação funcional	Verificar se a implementação funcional está completa, de acordo com a especificação de requisitos.	Após a execução de testes funcionais, comparar o número de funções em falta com o número de funções descritas na especificação de requisitos.	$X = 1 - A/B$ A = Número de funções em falta B = Número de funções descritas na URS	$0 \leq X \leq 1$ O valor obtido será tanto mais completo, quanto mais próximo de 1.	URS TR
Cobertura da implementação funcional	Verificar o quão correcta é a implementação funcional.	Através da execução de testes funcionais, comparar o número de funções em falta, ou implementadas de forma incorrecta, e compará-lo com o número de funções definidas na especificação de requisitos.	$X = 1 - A/B$ A = Número de funções em falta ou incorrectas, verificado através de teste B = Número de funções descritas na especificação de requisitos	$0 \leq X \leq 1$ O valor obtido será tanto mais correcto, quanto mais próximo de 1.	URS TR
Estabilidade da especificação	Verificar se a especificação funcional é estável	Comparar o número de funções definidas na especificação funcional que tiveram que ser	$X = 1 - A/B$ A = Número de	$0 \leq X \leq 1$ Quanto mais	URS

funcional	ao longo do ciclo de desenvolvimento do produto.	alteradas depois de o sistema ter começado a ser utilizado, com o número total de funções descritas na especificação de requisitos.	funções alteradas depois de o sistema ter começado a funcionar B = Número de funções descritas na especificação de requisitos	próximo de 1 for o valor de X, mais estável é a especificação funcional.	TR
-----------	--	---	--	--	----

Exactidão

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Exactidão de expectativa	Verificar se as diferenças entre o resultado final e os resultados esperados são aceitáveis.	Através de testes de <i>Input</i> e <i>output</i> , comparar o <i>output</i> obtido aos resultados esperados. Contar o número de casos que revelaram diferenças inaceitáveis neste aspecto.	$X = A/T$ A = Número de casos encontrados que revelaram diferenças inaceitáveis entre o <i>output</i> esperado e o efectivamente obtido T = Tempo da operação	$0 \leq X$ Quanto mais próximo de 0 for o valor de X, mais favorável é o resultado.	URS TR
Exactidão Computacional	Verificar a frequência com que os utilizadores do sistema se deparam com resultados inadequados.	Registar o número de resultados computacionais inadequados, com base nas especificações.	$X = A/T$ A = Número de resultados computacionais inadequados T = Tempo da operação	$0 \leq X$ Quanto mais próximo de 0 for o valor de X, mais favorável é o resultado.	URS TR
Precisão	Verificar a frequência com que os utilizadores do sistema se deparam com resultados com níveis de precisão inadequados.	Registar o número de resultados com níveis de precisão inadequados, com base nas especificações.	$X = A/T$ A = Número de resultados com níveis de precisão inadequados T = Tempo da operação	$0 \leq X$ Quanto mais próximo de 0 for o valor de X, mais favorável é o resultado.	URS TR

Interoperabilidade

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
-----------------	-----------	---------------------	--------	-----------------------------------	--------------------------

Troca de dados – baseada no formato dos dados	Verificar se as funções para troca de dados específicos foram implementadas correctamente.	Após testes, contar o número de formatos de dados aprovados para troca com outro software ou sistema, e comparar este valor com o número total de formatos de dados para troca.	$X = A/B$ A = Número total de formatos de dados para troca aprovados B = Número total de formatos de dados para troca	$0 \leq X \leq 1$ Quanto mais próximo de 1 for o valor de X, mais favorável é o resultado.	URS TR
Troca de dados – baseada nas tentativas de sucesso do utilizador	Verificar: - A frequência com que a troca de dados entre o produto em teste e outro produto de software falha; - A frequência com que a troca de dados entre o produto em teste e outro produto de software é bem sucedida; - Se o utilizador, por norma, é bem sucedido na troca de dados.	Contar o número de casos em que as funções de interface foram utilizadas e falharam.	$X = 1 - A/B$ A = Número de casos em que a troca de dados entre o produto em teste e outro produto de software falhou B = Número de casos em que se tentou trocar dados entre o produto em teste e outro produto de software $Y = A/T$ T = Tempo de operação	$0 \leq X \leq 1$ O valor obtido será tão mais favorável, quanto mais próximo de 1. $0 \leq Y$ Quanto mais próximo de 0 for o valor de X, mais favorável é o resultado.	URS TR

Segurança

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Auditabilidade de acesso	Verificar o quão completos são os registos de auditoria relativos à utilização do sistema e dados, por parte do utilizador	Avaliar o número de acessos gravados pelo sistema no seu histórico.	$X = A/B$ A = Número de acessos ao sistema gravados no histórico B = Número de acessos ao sistema efectuados durante o período de teste	$0 \leq X \leq 1$ Quanto mais próximo estiver de 1, mais correcto estará o valor de X.	TS TR
Controlo de acesso	Verificar o quão controláveis são os acessos ao sistema.	Contar o número de operações ilegais detectadas no sistema, e comparar esse valor com o número de operações ilegais previstas nas especificações de	$X = A/B$ A = Número de tipos de especificações	$0 \leq X \leq 1$ Quanto mais próximo de 1 estiver o valor de X, mais	TS TR

		teste.	ilegais detectadas no sistema B = Número de tipos de especificações ilegais previstas na especificação.	fácil será controlar os acessos ao sistema.	
Prevenção de corrupção de dados	Verificar com que frequência ocorre a corrupção de dados.	Contar o número de ocorrências relativas à corrupção de dados.	$X = 1 - A/B$ A = Número de ocorrências de corrupção de dados B = Número de casos de teste efectuados com vista a provocar corrupção de dados. $Y = A/T$ T = Tempo de operação	$0 \leq X \leq 1$ Quanto mais próximo estiver de 1, melhor. $0 \leq Y$ Quanto mais próximo X estiver de 0, melhores serão os resultados.	TS TR

Funcionalidade

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Conformidade funcional	Verificar se as funcionalidades do produto estão em conformidade com as regras, normas e convenções legais aplicáveis.	Contar o número de itens que requerem conformidade legal que foram implementados, e comparar este valor com o número de itens que requerem conformidade legal definidos na especificação.	$X = A/B$ A = Número de itens que requerem conformidade legal que foram implementados B = Número de itens que requerem conformidade legal definidos na especificação	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior conformidade funcional existirá.	TS TR
Conformidade <i>standard</i> do interface	Verificar se os interfaces estão em conformidade com as regras, normas e convenções legais aplicáveis.	Contar o número de interfaces que requerem conformidade legal que foram implementados, e comparar este valor com o número de interfaces que requerem conformidade legal definidos na especificação.	$X = A/B$ A = Número de interfaces que requerem conformidade legal que foram correctamente implementados	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior conformidade existirá.	TS TR

			B = Número de itens que requerem conformidade legal		
--	--	--	---	--	--

Confiabilidade					
Maturidade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Densidade estimada de falhas latentes	Contabilizar o número de problemas ainda existentes, que poderão surgir como falhas futuras.	Contar o número de falhas detectadas durante um determinado período, e, através de um modelo de estimativa de crescimento, estimar o número de falhas futuras.	$X = \frac{\text{ABS}(A1 - A2)}{B}$ <p>ABS() = Valor absoluto</p> <p>A1 = Número de falhas previstas</p> <p>A2 = Número de falhas detectadas</p> <p>B = Tamanho do produto</p>	$0 \leq X$ <p>A interpretação deste valor depende da fase em que se executa o teste, no entanto, quanto mais avançada for esta, melhores serão os valores mais próximos de zero.</p>	TR
Densidade de falha em casos de teste	Verificar o número de falhas ocorridas durante o período de teste.	Contar o número de falhas verificadas e comparar esse valor com o número de casos de teste realizados.	$X = A/B$ <p>A = Número de falhas verificadas</p> <p>B = Número de casos de teste realizados</p>	$0 \leq X$ <p>A interpretação deste valor depende da fase em que se executa o teste, no entanto, quanto mais avançada for esta, melhores serão os valores mais próximos de zero.</p>	TR
Resolução de falhas	Verificar o número de falhas resolvidas.	Contar o número de falhas que, em condições que anteriormente conduziam a erros, deixaram de ocorrer.	$X = A/B$ <p>A = Número de falhas resolvidas</p> <p>B = Número de falhas encontradas</p>	$0 \leq X$ <p>Quanto mais próximo de 1 for o valor de X, mais falhas terão sido resolvidas.</p>	TR
Remoção de falhas	Contabilizar quantas falhas foram corrigidas, e qual a proporção de falhas	Comparar o número de falhas removidas aquando do teste, com o número de falhas detectadas durante o mesmo e com o número de falhas	$X = A/B$ <p>A = Número de falhas corrigidas</p> <p>B = Número de</p>	$0 \leq X \leq 1$ <p>Quanto mais próximo o valor de X estiver de 1, menos falhas permanecerão</p>	TR

	removidas.	previstas.	falhas detectadas $Y = A/C$ C = Número de falhas previstas	no sistema. $0 \leq Y$ Quanto mais próximo o valor de Y estiver de 1, menos falhas permanecerão no sistema.	
Tempo médio entre falhas	Verificar a frequência com que ocorrem falhas no sistema.	Contar o número de falhas que ocorrem durante o tempo de operação, e computar o intervalo médio entre falhas.	$X = A/B$ $Y = C/B$ A = Tempo de operação B = Soma dos intervalos de tempo entre falhas consecutivas C = Número total de falhas no período de observação	$0 < X, Y$ Quanto maior for o valor de X e Y, maior o intervalo de tempo entre falhas.	TR
Adequação dos testes	Verificar quantos dos casos de teste necessários foram efectivamente executados.	Comparar o número de testes executados com o número de testes realmente necessários para abranger todos os aspectos importantes.	$X = A/B$ A = Número de testes efectuados B = Número de casos de teste necessários	$0 \leq X \leq 1$ Quanto mais próximo de 1 for o valor de X, mais adequados serão os testes.	URS TS TR
Maturidade dos testes	Verificar se o produto está bem testado.	Comparar o número de casos de teste com resultados positivos com o número de casos de teste definidos como necessários pelos requisitos.	$X = A/B$ A = Número de casos de teste efectuados com resultados positivos B = Número de casos de teste necessários para cobrir todos os requisitos	$0 \leq X \leq 1$ Quanto mais próximo de 1 for o valor de X, melhor	URS TS TR
Tolerância a falha					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Evitação de falhas	Contabilizar quantos padrões	Comparar o número de padrões de falha evitados com	$X = A/B$	$0 \leq X \leq 1$	TR

	de falha foram analisados com vista a evitar erros críticos.	o número de padrões de falha considerados.	A = Número de padrões de falha evitados durante os casos de teste B = Número de casos de teste de padrões de falha executados.	Quanto mais próximo de 1 for o valor de X, melhor será a evitação de falhas.	
Evitação de operações incorrectas	Verificar qual o número de funções implementadas que têm capacidade de evitar operações incorrectas.	Comparar o número de casos de teste de operações incorrectas que foram evitadas com o número de padrões de falha considerados.	$X = A/B$ A = Número de ocorrências críticas evitadas B = Número de casos de teste de padrões de falha executados.	$0 \leq X \leq 1$ Quanto mais próximo de 1 for o valor de X, melhor será a evitação de operações incorrectas	TR

Recuperabilidade

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Tempo médio de indisponibilidade	Medir o tempo médio que o sistema se encontra indisponível, após um evento inesperado.	Medir o tempo que o sistema se encontra indisponível quando, em períodos de teste, são encontrados erros, e calcular a média.	$X = A/B$ A = Tempo de indisponibilidade total B = Número total de casos de erro	$0 < X$ Quanto menor for o valor de X, menor será o tempo que o sistema se encontra indisponível.	TR
Tempo médio de recuperação	Medir o tempo médio que o sistema demora a recuperar, após um evento inesperado.	Medir o tempo que o sistema demora a recuperar quando, em períodos de teste, são encontrados erros, e calcular a média.	$X = \text{Soma}(A)/B$ A = Tempos de recuperação de cada vez que há problemas no sistema B = Número total de casos de erro	$0 < X$ Quanto menor for o valor de X, menor será o tempo de recuperação do sistema.	TR
Reinício	Medir a frequência com que o sistema poderá reiniciar dentro do tempo necessário.	Comparar o número de vezes que o sistema reinicia dentro d um intervalo de tempo aceitável, com o número total de vezes que o sistema reiniciou durante o período de teste.	$X = A/B$ A = Número de vezes que o sistema reinicia dentro d um intervalo de tempo aceitável B = Número total	$0 \leq X \leq 1$ Quanto maior for o valor de X, e mais próximo estiver de 1, maior a facilidade do sistema em reiniciar	TR

			de vezes que o sistema reiniciou durante o período de teste	rapidamente.	
Restauração	Medir capacidade do sistema se restaurar após um evento inesperado.	Comparar o número de vezes que o sistema foi restaurado com sucesso durante o período de teste com o número de casos de teste de restauração requerido pelas especificações.	$X = A/B$ A = Número de vezes que o sistema foi restaurado com sucesso durante o período de teste B = Número de casos de teste de restauração requerido pelas especificações	$0 \leq X \leq 1$ Quanto maior for o valor de X, maior será a capacidade de restauração do sistema.	URS TS TR
Efectividade de restauração	Verificar até que ponto é que a capacidade de restauração é eficaz.	Comparar o número de testes de restauração efectuados que cumprem os valores ideais com o número de casos efectuados.	$X = A/B$ A = Número de testes de restauração efectuados que cumprem os valores ideais B = Número de casos de teste efectuados	$0 \leq X \leq 1$ Quanto maior for o valor de X, maior será a capacidade efectividade de restauração do sistema.	TR
Fiabilidade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adequação da fiabilidade	Verificar se a fiabilidade do produto está em conformidade com as regras, normas e convenções legais aplicáveis.	Contar o número de itens que requerem conformidade legal que foram implementados, e comparar este valor com o número de itens que requerem conformidade legal definidos na especificação.	$X = 1 - A/B$ A = Número de itens que requerem conformidade legal que não foram implementados durante o teste B = Número de itens que requerem conformidade legal definidos na especificação	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior conformidade de fiabilidade existirá.	TS TR

Usabilidade

Compreensão

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Complitude da descrição	Verificar qual a proporção de funções descritas na descrição de produto	Efectuar questionários a um conjunto de utilizadores. Comparar o número de funções correctamente compreendidas com o número total de funções no produto.	$X = A/B$ A = Número de funções compreendidas B = Número de funções do produto	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, mais completa será a descrição	TR
Acessibilidade de demonstração	Verificar qual a proporção de demonstrações e/ou tutoriais aos quais o utilizador tem acesso.	Comparar o número de funções que são efectivamente demonstráveis com o número de funções que deveriam ter essa capacidade.	$X = A/B$ A = Número de demonstrações/tutoriais aos quais o utilizador tem efectivamente acesso B = Número total de demonstrações/tutoriais	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior será a acessibilidade de demonstração	TR
Efectividade de demonstração	Verificar qual a proporção de funções que o utilizador consegue operar com sucesso após demonstração	Observação do comportamento do utilizador.	$X = A/B$ A = Número de funções executadas com sucesso B = Número demonstrações acedidas	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior será a efectividade de demonstração	TR
Funções evidentes	Verificar qual a proporção de funções do produto que são evidentes para o utilizador.	Efectuar questionários e entrevistas aos utilizadores. Comparar o número de funções evidentes para o utilizador com o número total de funções do produto.	$X = A/B$ A = Número de funções evidentes para o utilizador B = Número de funções totais do produto	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor.	TR
Compreensão das funções	Verificar qual a proporção das funções do produto	Efectuar questionários e entrevistas aos utilizadores. Comparar o número de funções	$X = A/B$ A = Número de funções de interface	$0 \leq X \leq 1$ Quanto mais próximo X estiver de	TR

	que o utilizador terá capacidade de entender correctamente.	presentes no interface com o utilizador cujo propósito é claramente compreendido, com o número total de funções do mesmo interface.	compreendidas pelo utilizador B = Número total de funções de interface	1, melhor será a compreensão de funções.	
Compreensão do <i>Input e output</i>	Verificar se os utilizadores compreendem os <i>Inputs</i> pedidos e os <i>outputs</i> fornecidos pelo sistema.	Efectuar questionários e entrevistas aos utilizadores. Comparar o número de <i>Inputs</i> e <i>outputs</i> compreendidos pelo utilizador, com o número total de <i>Inputs</i> e <i>outputs</i> .	X = A/B A = Número de <i>Inputs</i> e <i>outputs</i> compreendidos pelo utilizador B = Número total de <i>Inputs</i> e <i>outputs</i> disponíveis no interface.	0 <= X <= 1 Quanto mais próximo X estiver de 1, melhor será a compreensão de funções.	TR

Aprendizagem

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Facilidade de aprendizagem das funções	Controlar o tempo que demora ao utilizador para aprender uma determinada função.	Observar e registar o comportamento do utilizador.	T = Tempo médio necessário para aprender a utilizar correctamente uma função.	0 < T Quanto menos for o valor de T, mais satisfatório será o resultado.	TR
Efectividade da documentação para o utilizador e/ou meios de ajuda	Verificar qual a proporção de tarefas que são correctamente utilizadas após consulta dos meios de ajuda.	Contar o número de tarefas completadas com sucesso após recorrer a mecanismos de ajuda, e comparar este valor com o número total de tarefas testadas.	X = A/B A = Número de tarefas completadas com sucesso após recorrer a mecanismos de ajuda B = Número total de tarefas testadas	0 <= X <= 1 Quanto mais próximo X estiver de 1, melhor	TR
Acessibilidade à ajuda	Analisar a proporção de itens de ajuda aos quais o utilizador tem acesso.	Comparar o número de tarefas para as quais foi localizada informação útil para ajuda com o número de tarefas testadas,	X = A/B A = Número de tarefas para as quais foi localizada ajuda online B = Número total de tarefas testadas	0 <= X <= 1 Quanto mais próximo X estiver de 1, mais acessível será a ajuda.	TR
Frequência de	Verificar com que frequência é que	Observar o comportamento do utilizador, e registar o número	X = A	0 <= X	TR

ajuda	um utilizador tem que recorrer à ajuda para completar uma tarefa.	de vezes que este tem necessidade de recorrer a ajuda para completar a sua tarefa.	A = Número de acessos que o utilizador faz à ajuda, até completar a tarefa	Quanto mais próximo X estiver de 0, mais satisfatório será o resultado.	
Operabilidade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Consistência operacional	Verificar a consistência dos componentes do interface com o utilizador.	Observar e questionar o utilizador.	$X = 1 - A/B$ $Y = N/T$ A = Número de mensagens ou funções que o utilizador considerou que defraudaram inaceitavelmente as suas expectativas B = Número de mensagens ou funções N = Número de operações que o utilizador considerou que defraudaram inaceitavelmente as suas expectativas T = Tempo de operação	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor. $0 \leq Y$ Quanto mais próximo X estiver de 0, melhor.	TR
Correcção de erros	Verificar a facilidade com que o utilizador pode corrigir erros em tarefas.	Realizar testes e observar o comportamento do utilizador.	$X = A - B$ A = Tempo final da correcção de erros numa determinada tarefa B = Tempo inicial da correcção de erros numa determinada tarefa	$0 < X$ Quanto menor for o valor de X, menor será o tempo dispendido para a correcção de erros, logo, mais satisfatório será o resultado.	TR
Disponibilidade de valores	Verificar a facilidade com que o utilizador pode	Contar o número de vezes que o utilizador tenta estabelecer ou seleccionar valores para um	$X = 1 - A/B$ A = Número de	$0 \leq X \leq 1$ Quanto mais	TR

	seleccionar valores para os parâmetros necessários para executar uma dada operação.	parâmetro e não é bem sucedido.	vezes que o utilizador tenta estabelecer ou seleccionar valores para um parâmetro e não é bem sucedido. B = Número total de vezes que o utilizador tenta estabelecer ou seleccionar valores.	próximo X estiver de 1, melhor	
Compreensão de mensagens	<p>Analisar a capacidade que o utilizador tem, para:</p> <ul style="list-style-type: none"> - Compreender as mensagens do sistema; - Processar a mensagem e prosseguir com a tarefa; - Memorizar mensagens. 	Testes ao utilizador.	$X = A/B$ A = Número de vezes que o utilizador para a sua tarefa, ou falha no seu desempenho por falta de compreensão de uma mensagem B = Tempo de operação	$0 \leq X$ Quanto mais próximo X estiver de 0, melhor	TR
Recuperabilidade de erros operacionais	Verificar se o utilizador consegue recuperar de situações difíceis.	Observação do comportamento do utilizador.	$X = 1 - A/B$ A = Número de vezes que erros ou alterações que conduziram a uma situação difícil, da qual o utilizador não havia sido avisado. B = Número total de erros ou alterações	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior será a recuperabilidade de erros operacionais.	TR
Tempo entre operações causadas por erro humano	Avaliar se o software funciona durante longos períodos de tempo, sem erros humanos.	Observação do comportamento do utilizador.	$X = A/B$ A = Tempo de operação B = Número de ocorrências de erros humanos	$0 < X$ Quanto maior for o valor de X, mais favorável será o resultado.	TR

Anulabilidade das operações do utilizador	Verificar a frequência com que o utilizador pode corrigir ou anular erros de <i>Input</i> .	Observação do comportamento do utilizador através de teste.	$X = A/B$ A = Número de erros de <i>Input</i> ou condições que o utilizador corrige com sucesso B = Número total de tentativas para corrigir erros/condições	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor.	TR
Personalização	Verificar qual a proporção de funções que podem ser personalizadas; Analisar a facilidade de personalização de procedimentos de operação.	Observação do comportamento do utilizador através de teste.	$X = A/B$ A = Número de funções personalizadas com sucesso B = Número total de tentativas de personalização	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior a capacidade de personalização.	TR
Acessibilidade física	Verificar qual a proporção de funções que podem ser personalizadas para utilizadores portadores de deficiências físicas.	Observação do comportamento do utilizador através de teste.	$X = A/B$ A = Número de funções acedidas com sucesso (nas condições testadas) B = Número total de funções	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor será a acessibilidade física.	TR
Atractividade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Interação atractiva	Verificar o quão atractivo é o interface para o utilizador	Realização de questionários junto dos utilizadores.	Questionários vocacionados para o aspecto gráfico do produto, realizados junto dos utilizadores.	Classificação da avaliação.	TR
Personalização da aparência do interface de utilizador	Verificar qual a proporção de elementos do interface que podem ser personalizados pelo utilizador.	Observação do comportamento do utilizador através de teste.	$X = A/B$ A = Número de elementos do interface que podem ser personalizados B = Número de	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, mais personalizável será o interface.	TR

			elementos do interface que o utilizador gostaria de poder personalizar.		
Usabilidade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adequação da usabilidade	Verificar se a usabilidade do produto está em conformidade com as regras, normas e convenções legais aplicáveis.	Especificar quais os itens que requerem conformidade legal, e elaborar e executar casos de teste de acordo com esses itens.	$X = 1 - A/B$ A = Número de itens que requerem conformidade legal que não foram implementados durante o teste B = Número de itens que requerem conformidade legal definidos na especificação	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior conformidade de usabilidade existirá.	URS TS TR

Eficiência					
Temporal					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Tempo de resposta	Definir o tempo estimado para completar uma tarefa, e quanto tempo demora o sistema a reagir a uma determinada operação.	Dar início a uma determinada tarefa, e verificar o tempo que demora até esta estar completa.	$X = \text{Tempo}$	$0 < X$ Quanto menor for o valor de X, melhor será o tempo de resposta	TR
Taxa de transferência	Avaliar a quantidade de tarefas concorrentes que podem ser executadas com sucesso num dado período de tempo.	Definir cada tarefa de acordo com a sua prioridade. Executar tarefas concorrentes e determinar o tempo que demora a executar a tarefa, nessas circunstâncias.	$X = A/B$ $A = \sum(A_i)/N$ $A_i = \text{Número de tarefas concorrentes observadas durante o período de tempo}$	$0 < X$ Quanto maior for o valor de X, melhor será o tempo de rendimento	TR

			avaliado N = número de avaliações B = taxa de transferência média requerida.		
Tempo de viragem	Definir o tempo estimado para a completar um conjunto de tarefas relacionadas entre si.	Através de testes, iniciar o conjunto de tarefas e registar o tempo que demora até que esteja completo.	$X = \text{Tempo}$	$0 < X$ Quanto menor for o valor de X, melhor.	TR
Tempo de espera	Calcular a percentagem de tempo que os utilizadores perdem à espera que o sistema responda.	Executar um conjunto de testes envolvendo tarefas concorrentes. Medir e registar o tempo necessário para concluir cada uma destas tarefas.	$X = A/B$ A = Tempo total gasto em espera B = Tempo da tarefa	$0 \leq X$ Quanto menor for o valor de X, menor o tempo de espera.	TR

Utilização de recursos

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Utilização de dispositivos I/O	Verificar se a utilização de dispositivos I/O é demasiado elevada, causando ineficiência.	Executar um conjunto de tarefas concorrentes, e registar a utilização de dispositivos I/O. Comparar esse valor com os valores previstos como ideais.	$X = A/B$ A = Tempo de ocupação dos dispositivos I/O B = Tempo especificado para ocupação dos dispositivos I/O	$0 \leq X \leq 1$ Quanto mais perto de 1 for o valor de X, melhores os resultados.	TR
Erros relacionados com I/O	Verificar a frequência com que o sistema encontra erros em operações relacionadas com dispositivos I/O.	Através de testes, verificar uma condição em que o sistema alcance uma situação de carregamento máximo de I/O. Registar o número de erros devido a falhas e avisos I/O.	$X = A/B$ A = Número de avisos ou falhas relacionadas com I/O B = Tempo de operação	$0 \leq X$ Quanto menor for o valor de X, menor o número de erros relacionados com I/O	TR
Tempo de espera do utilizador para utilização de componentes I/O	Analisar o impacto da utilização de dispositivos I/O nos tempos de	Executar um conjunto de tarefas concorrentes, e medir os tempos de espera do utilizador decorrentes de	$X = A$ A = Tempo total gasto à espera do termino de	$0 < X$ Quanto menor for o valor de X, menor o	TR

	espera.	operações I/O.	operações I/O.	tempo de espera.	
Ocorrência média de erros de memória	Verificar o número médio de erros de memória que ocorrem num dado período de tempo e numa dada componente do sistema.	Executar uma condição que conduza a que o sistema se encontre em situação de carga máxima. Registrar o número de erros que ocorrem, relativos a erros ou avisos de memória.	$X = A/B$ $A = \sum(A_i)/N$ $A_i =$ Número de mensagens de erros de memória. $N =$ Número de testes $B =$ Número de mensagens de erro de memória previstas.	$0 \leq X$ Quanto menor for o valor de X, melhor.	TR
Rácio de erros/tempo de memória	Avaliar quantos erros de memória foram encontrados num determinado período de tempo, e para um dado recurso.	Executar uma condição que conduza a que o sistema se encontre em situação de carga máxima. Registrar o número de erros que ocorrem, relativos a erros ou avisos de memória.	$X = A/B$ $A =$ Número de mensagens de aviso ou erros de sistema $B =$ Tempo de operação	$0 \leq X$ Quanto menor for o valor de X, melhor.	TR
Ocorrência média de erros de transmissão	Avaliar a média de erros relacionados com a transmissão foram encontrados num determinado período de tempo, e para um dado recurso.	Executar uma condição que conduza a que o sistema se encontre em situação de carga máxima. Registrar o número de erros que ocorrem, relativos a erros ou avisos de transmissão.	$X = A/B$ $A = \sum(A_i)/N$ $A_i =$ Número de mensagens de erro relativas a transmissões $N =$ Número de testes $B =$ Número de mensagens de erro relativas a transmissões previstas.	$0 \leq X$ Quanto menor for o valor de X, melhor.	TR
Transmissão média de erros por unidade temporal	Avaliar quantos erros relacionados com a transmissão foram encontrados num determinado período de tempo, e para um dado recurso.	Executar uma condição que conduza a que o sistema se encontre em situação de carga máxima. Registrar o número de erros que ocorrem, relativos a erros ou avisos de transmissão.	$X = A/B$ $A =$ Número de mensagens de aviso ou falha do sistema $B =$ Tempo da tarefa	$0 \leq X$ Quanto menor for o valor de X, melhor.	TR

Utilização da capacidade de transmissão	Verificar se o sistema é capaz de executar tarefas de acordo com a sua capacidade de transmissão.	Executar, concorrentemente, diversas tarefas com diversos utilizadores. Observar a capacidade de transmissão e comparar com a especificada.	$X = A/B$ A = Capacidade de transmissão B = Capacidade de transmissão especificada para o software em questão.	$0 \leq X \leq 1$ Quanto mais próximo de 1 estiver valor de X, melhor.	TR
---	---	---	--	---	----

Eficiência

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adequação da eficiência	Verificar se a eficiência do produto está em conformidade com as regras, normas e convenções legais aplicáveis.	Contar o número de itens que requerem conformidade legal que foram implementados, e comparar este valor com o número de itens que requerem conformidade legal definidos na especificação.	$X = 1 - A/B$ A = Número de itens que requerem conformidade legal que não foram implementados durante o teste B = Número de itens que requerem conformidade legal definidos na especificação	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior conformidade de eficiência existirá.	TS TR

Manutenção

Capacidade de análise

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Capacidade de rastrear de auditoria	Verificar possibilidade de saber especificamente qual a operação que causou a falha.	Observar o comportamento do utilizador ou da equipa de manutenção que está a tentar resolver as falhas.	$X = A/B$ A = Número de itens efectivamente registados durante a operação B = Número de itens planeados a registar, suficientes para supervisionar o estado do software durante a	$0 \leq X$ Quanto mais próximo de 1 estiver o valor de X, melhor.	TR

			operação		
Funções de diagnóstico	Verificar capacidade de prestação de funções de diagnóstico na análise causal.	Observar comportamento do utilizador ou da equipa de manutenção que está a tentar resolver as falhas usando funções de diagnóstico.	$X = A/B$ A = Número de falhas que a equipa de manutenção pode diagnosticar (usando as funções de diagnóstico) para entender a relação de causa-efeito B = Número total de falhas registadas	$0 \leq X \leq 1$ Quanto mais próximo de 1 estiver o valor de X, melhor o nível das funções de diagnóstico.	TR
Capacidade de análise de falhas	Verificar possibilidade de saber especificamente qual a operação que causou a falha e saber a causa da falha	Observar o comportamento do utilizador ou da equipa de manutenção que está a tentar resolver as falhas.	$X = 1 - A/B$ A = Número de falhas para as quais as suas causas ainda não foram encontradas B = Número total de falhas registadas	$0 \leq X \leq 1$ Quanto mais próximo de 1 estiver o valor de X, melhor a capacidade de análise de falhas.	TR
Eficiência da análise de falhas	Verificar eficiência da análise da causa da falha e facilidade com que se encontra a causa da falha	Observar o comportamento do utilizador ou da equipa de manutenção que está a tentar resolver as falhas.	$X = \text{Sum}(A)/B$	$0 \leq X$ Quanto mais próximo de 0 estiver o valor de X, melhor.	TR
Capacidade de monitorização do estado	Verificar capacidade de saber especificamente que operação causou a falha monitorizando os dados durante a mesma.	Observar o comportamento do utilizador ou da equipa de manutenção que está a tentar obter dados monitorizados de estados de software a registar durante a operação.	$X = 1 - A/B$ A = Número de casos nos quais a equipa de manutenção falhou na obtenção dos dados monitorizados B = Número de casos nos quais a equipa de manutenção tentou obter dados monitorizados registando o estado do software durante a operação	$0 \leq X \leq 1$ Quanto mais próximo de 1 estiver o valor de X, melhor.	TR

Alterabilidade

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Eficiência do ciclo de mudança	Verificar se o problema pode ser resolvido dentro de uma escala de tempo aceitável.	Monitorizar a interacção entre o utilizador e o fornecedor. Registrar o tempo demorado entre o pedido inicial do utilizador e a resolução do problema.	$X = \text{Sum}(A) / B$ $A = C - D$ D = Hora a que o utilizador acabou de mandar o pedido de manutenção ao fornecedor com o relatório do problema C = Hora a que o utilizador recebeu a versão de lançamento revista (ou estado do relatório) B = Número de versões revistas	$0 < X$ Quanto mais próximo de 0 estiver o valor de X, melhor, excepto quando o número de versões revistas é elevado.	TR
Mudança de implementação do tempo decorrido	Verificar se há a possibilidade de mudar facilmente o software para resolver uma falha	Observar o comportamento do utilizador ou da equipa de manutenção enquanto tentam mudar o software.	$X = \text{Sum}(A) / B$ $A = C - D$ C = Hora a que a causa das falhas são removidas com a mudança no software D = Hora a que a causa das falhas são encontradas B = Número de falhas registadas e removidas	$0 < X$ Quanto mais próximo de 0 estiver o valor de X, melhor, excepto quando o número de falhas é grande.	TR
Complexidade de modificação	Verificar a facilidade de mudança do software para resolver o problema	Observar o comportamento do utilizador ou da equipa de manutenção que está a tentar mudar o software.	$X = \text{Sum}(A / B) / N$ A = Tempo de trabalho gasto a mudar B = Tamanho da mudança de	$0 < X$ Quanto mais próximo de 0 estiver o valor de X, melhor.	TR

			software N = Número de mudanças		
Modificabilidade dos parâmetros	Verificar capacidade de facilmente mudar os parâmetros e o software e resolver problemas.	Observar o comportamento do utilizador ou da equipa de manutenção enquanto tentam mudar o software.	$X = 1 - A/B$ A = Número de casos em que a equipa de manutenção falha a mudança de software usando os parâmetros B = Número de casos em que a equipa tenta mudar o software usando os parâmetros	$0 \leq X \leq 1$ Quanto mais próximo de 1 estiver o valor de X, melhor.	TR
Capacidade de mudança controlada de software	Verificar a capacidade de identificar com facilidade versões revistas e a facilidade de mudar o software de forma a resolver os problemas.	Observar o comportamento do utilizador ou da equipa de manutenção enquanto tentam mudar o software.	$X = A/B$ A = Número de mudanças nos dados realmente registados B = Número de mudanças planeadas nos dados a serem registados de modo a rastrear as mudanças no software.	$0 \leq X \leq 1$ Quanto mais próximo de 1 estiver o valor de X, melhor ou quanto mais próximo de 0 menores são as hipóteses de mudanças terem acontecido.	TR

Estabilidade

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Rácio de sucesso na mudança	Verificar se o utilizador poderá, após alterações ao sistema, continuar a utiliza-lo com sucesso, bem como reduzir as falhas resultantes de efeitos colaterais à mudança.	Através de testes, observar o comportamento do utilizador após a manutenção. Contar as falhas que este encontra durante a execução do software, antes e depois da manutenção.	$X = A/B$ $Y = \{(A/B)/(C-D)\}$ A = Número de casos em que o utilizador encontra falhas antes do software ser alterado. B = Tempo de	$0 \leq X$ $0 \leq Y$ O rácio de sucesso na mudança será tanto mais positivo quando menos e mais próximo de 0 forem os valores de X e Y.	TR

			<p>operação antes do software ser alterado.</p> <p>C = Número de casos em que o utilizador encontra falhas depois do software ser alterado.</p> <p>D = Tempo de operação depois do software ser alterado.</p>		
Impacto da mudança	Verificar se o utilizador poderá, após alterações ao sistema, continuar a utilizá-lo com sucesso, bem como reduzir as falhas resultantes de efeitos colaterais à mudança.	Contar a ocorrência de falhas afectadas pela mudança depois das alterações.	$X = A/B$ A = Número de falhas que foram resolvidas pela mudança, no período especificado. B = Número de falhas resolvidas.	$0 \leq X$ O rácio de sucesso na mudança será tanto mais positivo quando menos e mais próximo de 0 for o valor de X.	TR

Capacidade de teste

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Disponibilidade da função de teste integrado	Verificar se o utilizador pode executar testes ao sistema sem a necessidade de qualquer outro recurso externo ao mesmo.	Observar o comportamento do utilizador que está a testar o sistema, após a manutenção.	$X = A/B$ A = Número de casos nos quais o utilizador pode utilizar funcionalidades de teste integradas. B = Número de casos de oportunidade de teste	$0 \leq X \leq 1$ Quanto maior e mais próximo de 1 for o valor de X, melhor.	TR
Eficiência de re-teste ao sistema	Analisar a facilidade com que o utilizador pode voltar a testar o sistema, como meio de saber se este está ou não pronto a funcionar	Observar o comportamento do utilizador que está a testar o sistema, após a manutenção.	$X = \text{Sum}(A)/B$ A = Tempo gasto em testes para assegurar que a falha em questão tinha sido resolvida	$X < 0$ Quanto menor for o valor de X, mais positivos serão os resultados.	TR

	nas condições desejadas.		B = Número de falhas resolvidas.		
Manutenção					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adequação da manutenção	Verificar se a manutenção do produto está em conformidade com as regras, normas e convenções legais aplicáveis.	Contar o número de itens que requerem conformidade legal que foram implementados, e comparar este valor com o número de itens que requerem conformidade legal definidos na especificação.	$X = 1 - A/B$ A = Número de itens que requerem conformidade legal que não foram implementados durante o teste B = Número de itens que requerem conformidade legal definidos na especificação	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior conformidade de manutenção existirá.	URS TS TR

Portabilidade					
Adaptabilidade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adaptabilidade de estruturas de dados	Verificar a adaptabilidade do produto a alterações na estrutura de dados.	Executar testes ao sistema, e verificar o comportamento do utilizador quando tenta adaptar o software a um novo ambiente.	$X = A/B$ A = Número de estruturas de dados que são operáveis, mas não visíveis devido a operações incompletas causadas por limitações de adaptação B = Número total de dados que se espera que sejam operáveis no novo ambiente ao qual o software foi adaptado.	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, melhor a adaptabilidade de estruturas de dados.	TR
Adaptabilidade ao	Verificar a adaptabilidade do	Executar testes ao sistema, e verificar o comportamento do	$X = 1 - A/B$	$0 \leq X \leq 1$	TR

hardware	produto a alterações na estrutura de hardware.	utilizador quando tenta adaptar o software a um novo ambiente.	A = Número funções operacionais que não conseguiram atingir os resultados esperados em diferentes ambientes de hardware B = Número total de funções testadas	Quanto mais elevado for o valor de X, melhor a adaptabilidade ao hardware.	
Adaptabilidade ao ambiente organizacional	Verificar a adaptabilidade do produto a alterações no ambiente organizacional.	Executar testes ao sistema, e verificar o comportamento do utilizador quando tenta adaptar o software a um novo ambiente organizacional.	$X = 1 - A/B$ A = Número funções operacionais que não conseguiram atingir os resultados esperados devido ao ambiente organizacional B = Número total de funções testadas	$0 \leq X \leq 1$ Quanto mais elevado for o valor de X, melhor a adaptabilidade ao ambiente organizacional.	TR
Adaptabilidade ao ambiente de software	Verificar a adaptabilidade do produto a alterações no software	Executar testes ao sistema, e verificar o comportamento do utilizador quando tenta adaptar o software a um novo ambiente de software	$X = 1 - A/B$ A = Número funções operacionais que não conseguiram atingir os resultados esperados durante testes num ambiente de software específico. B = Número total de funções testadas	$0 \leq X \leq 1$ Quanto mais elevado for o valor de X, melhor a adaptabilidade ao ambiente de software.	TR
Capacidade de instalação					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Facilidade de Instalação	Verificar se o utilizador é capaz de instalar o sistema com facilidade.	Observar o comportamento do utilizador quando tenta instalar o sistema.	$X = A/B$ A = Número de casos de sucesso na instalação do sistema B = Número total	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, mais fácil será reinstalar o sistema.	TR

			de tentativas de instalação do sistema.		
Facilidade de reinstalação	Verificar se o utilizador é capaz de reinstalar o sistema com facilidade.	Observar o comportamento do utilizador quando tenta reinstalar o sistema.	$X = 1 - A/B$ A = Número de casos em que o utilizador falha na reinstalação do sistema B = Número total de tentativas de reinstalação do sistema.	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, mais fácil será reinstalar o sistema.	TR

Coexistência

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Coexistência disponível	Analisar a frequência com que são encontradas restrições ou erros inesperados aquando da interacção com outros sistemas.	Testar o produto concorrentemente com outro software frequentemente utilizado pelo utilizador.	$X = A/B$ A = Número de restrições ou erros encontrados durante o teste B = Duração do teste	$0 \leq X$ Quanto mais próximo X estiver de 0, melhor.	TR

Capacidade de substituição

Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Uso contínuo de dados	Verificar se é possível continuar a usar os mesmos dados após substituição do produto; Avaliar as perspectivas de sucesso na migração de dados.	Observar o comportamento do utilizador aquando da substituição do software.	$X = A/B$ A = Número de itens de dados que continuam a ser utilizados após a substituição B = Número total de itens de dados antigos que deverão, segundo planeado, continuar a ser usados depois da substituição do software.	$0 \leq X \leq 1$ Quanto mais elevado for o valor de X, melhor.	TR

Inclusão de funções	Verificar se é possível continuar a usar as mesmas funções após substituição do produto; Avaliar as perspectivas de sucesso na migração de dados.	Observar o comportamento do utilizador aquando da substituição do software.	$X = A/B$ A = Número de funções que produzem resultados semelhantes às anteriores, e nas quais não foram necessárias alterações B = Número de funções testadas que são semelhantes às funções fornecidas pelo software a substituir.	$0 \leq X \leq 1$ Quanto mais elevado for o valor de X, melhor.	TR
Consistência funcional de apoio ao utilizador	Verificar a consistência dos novos componentes com o interface do utilizador.	Observar e questionar o utilizador.	$X = 1 - A/B$ A = Número de funções que o utilizador considerou inaceitavelmente inconsistentes com as suas expectativas B = Número de funções novas	$0 \leq X$ Quanto mais elevado for o valor de X, melhor.	TR
Portabilidade					
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos	Output ao qual se aplica
Adequação da portabilidade	Verificar se a portabilidade do produto está em conformidade com as regras, normas e convenções legais aplicáveis.	Contar o número de itens que requerem conformidade legal que foram implementados, e comparar este valor com o número de itens que requerem conformidade legal definidos na especificação.	$X = 1 - A/B$ A = Número de itens que requerem conformidade legal que não foram implementados durante o teste B = Número de itens que requerem conformidade legal definidos na especificação	$0 \leq X \leq 1$ Quanto mais próximo X estiver de 1, maior conformidade de portabilidade existirá.	TS TR

Métricas de validação durante a fase de utilização de SGD

Efectividade				
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos
Efectividade de tarefas	Verificar qual a proporção de objectivos para cada tarefa é atingida correctamente.	Testes ao utilizador.	$X = 1 - \Sigma A $ A= valor proporcional de cada componente errado ou em falta no <i>output</i> da tarefa.	$0 \leq X \leq 1$ Quanto mais perto de 1 estiver o valor de X, mais efectivas são as tarefas.
Complitude de tarefas	Verificar a proporção de tarefas completas.	Testes ao utilizador.	$X = A/B$ A = Número de tarefas completas B = Número de tarefas que o utilizador tentou executar	$0 \leq X \leq 1$ Quanto mais perto de 1 estiver o valor de X, melhor.
Frequência de erro	Concluir acerca da frequência dos erros.	Testes ao utilizador.	$X = A/B$ A = Número de erros feitos pelo utilizador B = Tempo ou número de tarefas	$0 \leq X$ Quanto mais perto de 0 estiver o valor de X, mais favorável é o resultado.

Produtividade				
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos
Tempo de tarefas	Verificar o tempo que demora a executar uma tarefa.	Testes ao utilizador.	$X = A$ A = Tempo da tarefa	$0 \leq X$ Quanto menor for o valor de X, mais favorável será o resultado.
Eficiência de tarefas	Avaliar o quão eficientes são os utilizadores.	Testes ao utilizador.	$X = A/B$ A = Efectividade da tarefa B = Tempo da tarefa	$0 \leq X$ Quanto mais elevado for o valor de X, mais favorável será o resultado.
Produtividade económica	Averiguar o quão rentável é o utilizador.	Testes ao utilizador.	$X = A/B$ A = Efectividade da tarefa B = Custo total da tarefa	$0 \leq X$ Quanto mais elevado for o valor de X, mais favorável será o resultado.
Proporção produtiva	Calcular qual a percentagem de tempo que o utilizador gasta a efectuar acções produtivas.	Testes ao utilizador.	$X = A/B$ A = Tempo produtivo = Tempo da tarefa - tempo de ajuda - tempo de erro - tempo de pesquisa B = Tempo da tarefa	$0 \leq X \leq 1$ Quanto mais próximo de 1 for o valor de X, mais favorável será o resultado.
Danos de software	Comparar a eficiência do utilizador com a de um <i>expert</i> no assunto.	Testes ao utilizador.	$X = A/B$ A = Eficiência do utilizador para concretizar uma tarefa B = Eficiência do <i>expert</i> para concretizar uma tarefa	$0 \leq X \leq 1$ Quanto mais próximo de 1 for o valor de X, mais favorável será o resultado.

Segurança				
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos
Segurança das pessoas afectadas pela utilização do sistema	Verificar a incidência de danos sobre as pessoas que utilizam o sistema.	Estatísticas de uso.	$X = 1 - A/B$ A = Número de pessoas em risco B = Número total de pessoas potencialmente afectadas pelo sistema	$0 \leq X \leq 1$ Quanto mais próximo de 1 for o valor de X, mais favorável será o resultado.
Danos económicos	Verificar a incidência de danos económicos.	Estatísticas de uso.	$X = 1 - A/B$ A = Número de ocorrências de danos económicos B = Número total de situações de uso	$0 \leq X \leq 1$ Quanto mais próximo de 1 for o valor de X, mais favorável será o resultado.
Danos de software	Verificar a incidência de danos no software.	Estatísticas de uso.	$X = 1 - A/B$ A = Número de ocorrências de corrupção de software B = Número total de situações de uso	$0 \leq X \leq 1$ Quanto mais próximo de 1 for o valor de X, mais favorável será o resultado.

Satisfação				
Nome da métrica	Objectivo	Método de aplicação	Medida	Interpretação dos valores obtidos
Escala de satisfação	Verificar o nível de satisfação do utilizador.	Testes ao utilizador.	$X = 1 - A/B$ A = Escalas psicométricas B = Média de população	$0 < X$ Quanto maior for o valor de X, mais favorável será o resultado.
Questionário de satisfação	Verificar a satisfação do utilizador, relativamente a funcionalidades	Testes ao utilizador.	$X = \Sigma(A)/B$ A = Resposta a uma questão	Comparar com valores obtidos em avaliações anteriores.

	específicas.		B = número de respostas	
Uso discricionário	Avaliar que proporção de utilizadores escolhem utilizar o sistema.	Observação de uso.	$X = A/B$ A = Número de vezes que funções ou aplicações do software são utilizadas. B = Número de vezes que é suposto que elas sejam utilizadas.	$0 \leq X \leq 1$ Quanto mais próximo de 1 for o valor de X, mais favorável será o resultado.