



Trabalho Final de Curso

Licenciatura Informática de Gestão

Software de Gestão CMMI e FPA

07-07-2012

∴

Fernando Martinez Amante – a20090028
Nelson Trindade Silva – a20090261

Turma 3N1

Orientador: Prof. Doutor Pedro Malta

Orientador Externo: Prof. Pedro Morais Ferreira

Índice

Resumo.....	4
Abstract.....	4
1 Introdução.....	5
1.1 Visão Geral do Projeto.....	5
2 Enquadramento teórico.....	6
2.1 Enquadramento Capability Maturity Model® Integration	6
2.2 Enquadramento Functon Point Analysis.....	7
2.3 A escolha de FPA para processo de medição do CMMI.....	7
3 Método.....	10
3.1 Visão Geral do Produto.....	10
3.2 Análise Técnica dos Requisitos.....	11
3.2.1 Componente CMMI.....	11
3.2.2 Componente FPA.....	11
3.2.3 Componente de Documentação.....	12
3.2.4 Integração dos vários requisitos.....	13
3.3 Plataformas de desenvolvimento.....	13
3.3.1 PHP e HTML.....	13
3.3.2 MySQL.....	14
3.4 Metodologias, Modelos e Ferramentas Utilizadas.....	14
3.4.1 Metodologia Unified Modeling Language.....	14
3.4.2 Metodologia Business Process Modeling Notation	15
3.4.3 Modelo Relacional de Base de Dados.....	16

3.4.4	MySQL Workbench.....	16
3.4.5	Eclipse Java EE IDE for Web Developers – UML2 Tool Diagramming.....	16
3.4.6	LibreOffice.....	17
3.4.7	Microsoft Visio com Addin Process Modeler 5.....	17
4	Resultados.....	17
4.1	Componente CMMI.....	17
4.2	Componente FPA.....	18
4.3	Componente de Documentação.....	19
4.4	Componente de Utilizadores, Grupos de utilizadores e permissões.....	19
4.5	Componente de Fluxo de Processos.....	20
4.6	Gestão de Requests.....	21
5	Trabalho Futuro e Conclusões.....	22
5.1	Possíveis desenvolvimentos.....	22
5.1.1	Controlo de tempos.....	22
5.1.2	Reportes de produtividade.....	22
5.1.3	Consultas dos Logs.....	22
5.2	Conclusões.....	23
	Bibliografia.....	24
	Anexos.....	25
	Anexo 1 – Levantamento de Requisitos.....	25
	Anexo 2 – Use Cases.....	25
	Anexo 3 – Desenho da Base de Dados.....	25
	Anexo 4 – Diagrama da Base de Dados.....	25
	Anexo 5 – Diagramas de Funcionamento da Aplicação.....	25
	Glossário.....	26

Resumo

O principal objetivo deste projeto é o desenho de uma aplicação que possa ser instalada em diferentes organização, com diferentes tipos de processos produtivos e que desejem ter uma ferramenta, que consiga gerir a difícil tarefa de implementar Processos CMMI nos seus processos. Isto significa que a ferramenta tem de ser totalmente parametrizável e ainda assim ser suficientemente robusta para conseguir manter os dados sem problemas de integridade, sendo uma mais valia para a organização, ao invés de um problema na implementação de metodologias como CMMI e FPA.

Abstract

The main goal of this project is to design an application to be installed in different kinds of organization, with different working process and desire to have a tool, that manages the difficult task of implementing CMMI Processes in their processes. This means that the tool has to be totally parametrizable and still be sturdy enough to keep the data clean and without problems, being a help to the organization, instead of a problem in the implementation of methodologies like CMMI and FPA.

1 Introdução

1.1 Visão Geral do Projeto

Este projeto visa o Levantamento de Requisitos e Análise de uma ferramenta que permita a uma empresa de serviços informáticos, a gestão de todo o seu processo produtivo e ou projetos recorrendo ao modelo Capability Maturity Model Integrationⁱ (CMMI), com recurso a uma metodologia de contagem de tamanho de software, como os Function Point Analysisⁱⁱ (FPA) totalmente integrado.

Este sistema deverá ser parametrizável de forma a que a organização consiga ter o controlo do seu processo de maturidade no modelo CMMI, permitindo por exemplo a perceção do nível CMMI em que o seu processo produtivo se encontra, a escolha dos vários documentos e informação necessária para cada nível consoante a sua própria maturidade, a fácil obtenção de certificados CMMI, etc. Deverá ser um sistema que vise a eliminação de documentos físicos, permitindo uma gestão efetiva e simples dos mesmos, bem como um repositório efetivo de informação, para que seja fácil tornar essa mesma informação em conhecimento pelos intervenientes no processo.

Visa também a implementação de um método de contagem de tamanho de software baseado em FPA, com o intuito de medir o esforço necessário para cada projeto/processo da empresa, de forma a permitir um controle e visão sobre o esforço necessário para o desenvolvimento de projetos futuros, bem como uma visão estratégica com base nos dados colecionados na ferramenta, que normalmente só surge mais tarde em organizações com uma maior maturidade neste ponto.

2 Enquadramento teórico

2.1 Enquadramento Capability Maturity Model® Integration

O CMMI é um enquadramento de referência que as organizações podem empregar para melhorar os seus processos de desenvolvimento, aquisição, e manutenção de produtos e serviços. Nascido no *Software Engineering Institute* pertencente à Carnegie Mellon University, CMMI é a nova geração de uma linha de modelos de maturidade que se iniciou no princípio dos anos noventa com o famoso CMM-SW (*Capability Maturity Model for Software Engineering*).

O CMMI é um modelo de referência que contém práticas (*Genéricas ou Específicas*) necessárias à maturidade em disciplinas específicas (*Systems Engineering (SE)*, *Software Engineering (SW)*, *Integrated Product and Process Development (IPPD)*, *Supplier Sourcing (SS)*). Desenvolvido pelo SEI (*Software Engineering Institute* ⁱⁱⁱ) da Carnegie Mellon University ^{iv}, o CMMI é uma evolução do CMM ^v e procura estabelecer um modelo único para o processo de melhoria corporativo, integrando diferentes modelos e disciplinas ^{vi}.

Baseados nos princípios da qualidade total (TQM) popularizados por autores como Crosby, Deming e Juran, estes modelos propõem um conjunto de práticas que as organizações podem adotar para implantar processos produtivos mais efetivos. São chamados modelos de maturidade porque propõem adotar as ditas práticas de uma forma gradual: primeiro devem pôr-se em prática as áreas de processo pertencentes a um determinado nível, para depois, sobre esta base, introduzir as correspondentes ao nível seguinte.

2.2 Enquadramento Function Point Analysis

A FPA é uma técnica de medição das funcionalidades fornecidas por um software do ponto de vista dos utilizadores, desenvolvida pela International Function Point Users' Group (IFPUG) ^{vii}. Function Point (FP) é a sua unidade de medida, que tem por objetivo tornar a medição independente da tecnologia utilizada para a construção do software. Ou seja, a FPA procura medir o que o software faz, e não como ele foi construído.

Portanto, o processo de medição (também chamado contagem de function points) é baseado em uma avaliação padronizada dos requisitos funcionais do utilizador. Este procedimento padrão está descrito pelo IFPUG^{viii} no seu Manual de Práticas de Contagem ou em inglês Counting Practices Manual – CPM.

As principais técnicas de estimativa de projetos de desenvolvimento de software assumem que o tamanho de um software é um vetor importante para a determinação do esforço para sua construção. Logo, saber o seu tamanho é um dos primeiros passos do processo de estimativa de esforço, prazo e custo.

Daí ser importante destacar que os function points não medem diretamente esforço, produtividade ou custo. É exclusivamente uma medida de tamanho funcional do software. Este tamanho, em conjunto com outras variáveis, é que poderá ser usado para derivar produtividade, estimar esforço e custo do projeto de software.

2.3 A escolha de FPA para processo de medição do CMMI

“As estimativas de tempo de desenvolvimento de projetos é um dos grandes problemas do desenvolvimento de *software*, e a situação não parece melhorar mesmo com a implementação de *frameworks* como o CMMI. Não parece importar o quanto avançada ou madura é a organização das TI. Quanto um projeto parece cumprir a data de entrega, é frequentemente uma coerção – isto é, o gestor de projeto consegue manipular ou de alguma forma alterar a estimativa do projeto para bater certo com uma data final muitas vezes arbitrária.” (DEKKERS, 2010).

Carol Dekkers é a presidente da *Quality Plus Technologies* ^{ix} – empresa dedicada exclusivamente a trabalhar com outras empresas para que as mesmas melhorem o seu

Return of Investment(ROI) na tecnologia – e uma das *experts* mundiais no que toca á FPA. É também *Project Editor* na *International Organization for Standardization* (ISO). Para além destes cargos, Carol tem inúmeras *Professional Affiliations*, como aliás é referido no sítio da *Quality Plus Technologies* ^x.

A SEI CMMI apresenta um modelo conjunto padrão de processos de software a ser em satisfeitos antes que uma organização possa ser avaliada na maturidade da capacidade de um particular nível. Medição, se utilizado adequadamente, pode ser um facilitador, para que as empresas possam avançar e completa muitas estas áreas de processo. «Medir» deve encorajar repetibilidade do processo: é exatamente o lugar onde a maturidade do processo e FP se conjugam para trazer resultados mais positivos. A contagem FPA espelha o processo estruturado de avaliação pelos pares para solidificar e quantificar as necessidades funcionais dos utilizadores. Como tal, há processos no CMMI, que podiam ser suportados pelo processo de documentação de funções lógicas contadas com os FP's, enquanto outras áreas podem ser suportadas no processo de contagem dos FP's. Em último caso, os resultados baseadas na medição com FP's podem ser um fator suficientemente significativo para ajudar o modelo CMMI a alcançar a maturidade desejada das áreas de processo.

Uma imagem vale mil palavras e um exemplo pode providenciar um entendimento mais claro de determinada afirmação teórica.

1. Gestão de Requisitos:

- Medições podem ser efetuadas para perceber o estado dos requisitos – estarão eles aceites, revistos ou rejeitados?
- O tamanho das alterações pode ser levantado – comparando o “size” (tamanho) em FP levantado originalmente com o que se obteve aquando das alterações.

2. Planeamento do Projeto:

- Estimativas do “size” do desenvolvimento de produtos de *software* ou alterações dos mesmos, podem ser derivadas de acordo com o processo

previamente documentado, i.e. tem-se à disposição um repositório de informação para uma estimativa rápida, tão necessária hoje em dia.

Nos últimos 5/6 anos, autores de livros e *experts* em CMMI, reconheceram a importância da medição de *software* em modelos CMMI. Desde 2001 que o nível 2 do modelo de CMMI, tem incluído uma área de processo específico para a medição e análise de *software*, ao contrário da versão anterior em que se previa uma forma de medição, a partir do nível 3. Pois bem, os FP's quantificam o “size” do software, objetivamente de um ponto de vista do utilizador, independentemente da implementação, podendo ser considerado válido para ser usado em conjunto com a *framework* CMMI.

3 Método

3.1 Visão Geral do Produto

A finalidade deste software é a de agilizar o processo de trabalho de uma organização, permitindo a implementação de metodologias complexas como CMMI Development e FPA, além de ter um maior controlo sobre as tarefas que fazem parte do processo produtivo de uma organização, nomeadamente de uma organização cujo o modelo de negocio seja o desenvolvimento de software. Apesar deste ser o principal objetivo, ao desenharmos uma aplicação cuja componente CMMI seja parametrizável, iremos também estar a responder a outros modelos CMMI como o Acquisition e Services.

O carácter parametrizável desta ferramenta leva a que de uma forma mais simples as organizações consigam adaptar o seu processo produtivo ao CMMI. Como acontece na esmagadora maioria das empresas que não partem do zero e que adotam o CMMI, numa primeira fase o CMMI “adapta-se” ao processo produtivo da organização e não o contrário. Daí existir o Capability Level, que numa primeira fase, o primeiro grande objetivo de uma organização é de atingir o Capability Level 3. Ao longo deste caminho de adaptação de um processo produtivo de uma organização, os vários Processos CMMI implementados, podem pertencer a vários Maturity Levels, e pode-se dar o caso de essa mesma organização estar a cumprir apenas alguns dos Processos do Maturity Level 2, não podendo ser certificada com o Maturity Level 3.

Como cada organização é um caso diferente de outro e a implementação de modelos como o CMMI difere de organização para organização, uma ferramenta que seja pouco maleável e force o uso de determinados Processos, independente de serem usados inicialmente pela organização, está condenada a simplesmente não ser usada.

Assim e aproveitando o facto de que uma implementação CMMI depende muito do processo produtivo inicial, isto leva-nos a pensar numa ferramenta que consiga servir qualquer um destes casos. E a melhor forma de o fazer, é a de a desenhar com o carácter marcadamente virado para uma parametrização praticamente total. De facto, desenhar um software altamente parametrizável e que consiga ser ao mesmo tempo, manter a coerência dos dados e dos processos e seja de simples implementação, é uma tarefa complexa e que

requer um alto grau de abstração na abordagem ao mesmo.

Para além do referido, o software deverá ser obviamente, de carácter colaborativo, permitindo que os utilizadores usem o mesmo como base de trabalho, não necessitando de outras ferramentas que suportem a gestão do seu trabalho diário.

3.2 Análise Técnica dos Requisitos

De forma a responder aos vários requisitos, e dado existirem vários grupos de requisitos distintos, iremos abordar o tema por grupos, fazendo uma abordagem global e integrada da solução.

3.2.1 Componente CMMI

Alguns dos requisitos, tal como o 5-CMMI e o 6-CMMI¹ apontam claramente para que tivéssemos uma ferramenta parametrizável, e dado que, de organização para organização os processos produtivos mudam, não poderíamos abordar esta questão de outra forma. Conforme o requisito 2-CMMI, surge a necessidade de diferenciar Projetos e Aplicações, ou seja, cada Projeto poderá ter várias Aplicações, cada Aplicação poderá ter várias Áreas de Processo e por sua vez vários Processos. Obviamente, se falamos de vários Processos por Projeto e Aplicação, estamos a falar de vários objetivos por Processo e várias premissas para cada Objetivo. No desenho da B.D. é possível validar justamente o que referimos. Toda a componente CMMI pode ser parametrizada da acordo com o que o cliente pretender, e de acordo com o seu processo produtivo.

Ainda dentro dos requisitos definidos na componente CMMI, os requisitos 1-CMMI, 4-CMMI e 7-CMMI, apontam para que na aplicação seja possível ter vários tipos de utilizadores bem como o estabelecimento de vários perfis de utilizadores por forma a conseguirem intervir nos processos CMMI.

3.2.2 Componente FPA

¹ Consultar Anexo 1 – Levantamento de Requisitos

Apesar de serem apenas 5 requisitos, a necessidade do cliente em ter as funcionalidades referidas, leva a que seja pensada uma estrutura que consiga albergar todas as funcionalidades de FPA. Sendo os requisitos 2-FPA e 5-FPA puramente ligados à componente FPA, os outros três requisitos apontam para a ligação com a componente CMMI que falámos anteriormente. Assim, os requisitos 1-FPA, 3-FPA e 4-FPA complementam-se indicando que a contagem de FP deve estar diretamente relacionada com um ou mais Processos CMMI. Para além disso o requisito 4-FPA volta a referir a necessidade do sistema conseguir controlar o acesso dos utilizadores às várias funcionalidades do software, recorrendo à criação de uma sistema com permissões. Daqui também se pode concluir que as permissões para CMMI devem ser as mesmas para FPA, sendo o tratamento das permissões dos vários grupos de utilizadores igual em ambas as componentes.

Quanto aos requisitos 2-FPA e 5-FPA, estamos perante a vertente básica de FPA e sem as quais não é possível efetuar as contagens. O requisito 2-FPA estabelece que Tipo de Contagem se vai efetuar, ou seja, projeto de desenvolvimento ou de melhoria, que obviamente tem um impacto diferente consoante a escolha. E o requisito 5-FPA que se refere um elemento extremamente importante para a contagem de FP, que é o Fator de Ajustamento o qual avalia a complexidade de uma determinada aplicação, e que servirá para ajustar os *function points* desajustados, calculados para uma determinada funcionalidade de uma aplicação.

3.2.3 Componente de Documentação

Este ponto, é claramente um apoio para o processo CMMI, de onde advém a necessidade de estabelecer *templates* e o tratamento da documentação associada aos vários Processos CMMI. Mais uma vez, dada a variação existente de organização para organização, no que toca à forma de adaptar o seu próprio processo produtivo à metodologia CMMI, levando a que existam documentos diferentes para os Processos, de cliente para cliente, leva-nos a uma abordagem muito pouco rígida. Se o software guardasse todos os dados necessários na sua própria B.D. para depois produzir os

documentos quando necessário, iria tornar a aplicação muito pouco flexível e iria torna a tarefa de levantamento de todas as necessidades de todos os clientes, em algo sobre natural e sem grande interesse, dado que se pode simplesmente associar um documento a um Processo e com isso ganhar noutras vertentes, tais como a facilidade de adaptação do software a várias organizações, a não limitação nas necessidades de guardar dados na aplicação, que da outra forma ficariam limitadas.

3.2.4 Integração dos vários requisitos

Dos vários requisitos das várias componentes, surgem várias necessidades de implementação no software. Uma das principais trata-se da integração das três componentes de forma a que para o utilizador, as mesmas sejam transparentes. O outro grande desafio é o de criar um conceito de macro-tarefa que englobe o trabalho a ser desenvolvido pelos colaboradores das organizações e que passam por vários Processos CMMI, que podem ou não ter contagem de FP e também podem ou não ter documentação associada com os mesmos Processos. Para além deste ponto, surge ainda outro, que passa pelo controlo das autorizações ou permissões para os utilizadores do software. De fato e se avaliarmos bem um Processo Produtivo de uma organização verificaremos que as responsabilidades das pessoas variam consoante o cargo desempenhado, bem como com as tarefas que lhes são atribuídas. Assim, as permissões devem envolver as três componentes e ainda conseguir abordar o requisito das macro-tarefas.

No ponto Resultados (ponto 4) iremos abordar a forma como desenhamos a aplicação e como procuramos responder aos requisitos do utilizador, mantendo a identidade parametrizável da aplicação que apresentamos.

3.3 Plataformas de desenvolvimento

3.3.1 PHP e HTML

Sendo uma linguagem de programação cada vez mais usada para ferramentas colaborativas, e dado o seu carácter “invisível” aos browsers já que corre sempre do lado do servidor, acreditamos que em conjunto com HTML, seria a plataforma ideal para o

desenvolvimento desta aplicação. A forma fácil de fazer ligação a base de dados MySQL e a versatilidade da mesma, torna esta linguagem a nossa preferida para um eventual desenvolvimento. Durante toda a nossa análise tivemos essa ideia em mente, e isso mesmo está espelhado no desenho da solução que elaboramos.

3.3.2 MySQL

Motor de B.D. candidato ideal e normalmente associado a PHP, dado a sua fácil ligação com essa linguagem de programação, e com todas as funcionalidades de motores normalmente conhecidos por serem mais robustos. MySQL tem hoje em dia um forte expressão no mundo da informática, e compensa a sua menor capacidade de lidar com base de dados muito grandes, com uma grande facilidade de implementação e de interfaces disponíveis para o seu uso. Como poderá ser visto mais abaixo neste relatório, usamos uma ferramenta para desenhar a base de dados que exporta ativamente para SQL e que consegue gerir uma B.D. em MySQL.

3.4 Metodologias, Modelos e Ferramentas Utilizadas

Neste ponto demonstraremos a razão do uso das várias metodologias e modelos bem como as várias ferramentas utilizadas, para que fique claro a nossa abordagem no desenho deste software, e para que as premissas pelas quais nos guiámos sejam também mais transparentes.

3.4.1 Metodologia Unified Modeling Language

UML ou em português Linguagem de Modelagem Unificada foi criada por Rumbaugh, Booch e Jacobson, profissionais da área de sistemas e processos, que se uniram com o objetivo de se criar um padrão para desenvolvimento de software que reunisse as melhores práticas de metodologia de sistemas.

Esta linguagem pode ser utilizada para criar um modelo que tenha a capacidade de se abstrair as fases de um projeto de criação de software. Neste modelo, diversos diagramas

auxiliam na visualização do problema e a conceção da solução, permitindo uma visão macro dos objetos e seus relacionamentos.

Sendo o único objetivo do Modelo de Análise a tradução para linguagem formal dos elementos levantados pela Engenharia de Requisitos, ou seja a modelação dos Requisitos necessários para a Engenharia de Software, o uso do UML como metodologia descritiva torna-se perfeitamente natural na especificação, construção, visualização e documentação de SI. Naturalmente os diagramas Use Case dão um importante contributo para tornar os Requisitos de Utilizador mais claros e com um objetividade nem sempre bem conseguida pela documentação tradicional.

3.4.2 Metodologia Business Process Modeling Notation

Cada vez mais as organizações precisam de processos de negócios bem definidos e, ao mesmo tempo, totalmente flexíveis, e os sistemas de informação precisam acompanhar o seu progresso.

Através do BPMN podemos alcançar esses objetivos, através de uma notação gráfica padronizada para modelação de processos de negócio que funciona como meio-termo entre as áreas de desenvolvimento e os utilizadores. A interligação dos Use Cases (UML) com os diagramas em BPMN é, apesar de não serem normalmente associados, mais próxima do que se pode imaginar. Se os Use Case espelham bem a realidade do negócio, os diagramas BPMN espelham de uma forma integral o passo seguinte no desenho de uma solução. Ao mesmo tempo que demonstra a interatividade do utilizador com o sistema, consegue também espelhar os pormenores técnicos, próprios do desenho de um software, de uma forma clara e tão importante quando isto, num único diagrama. Esta abordagem pode não ser a mais usada em termos académicos, mas cada vez mais empresas e pessoas usam Use Cases e diagramas em BPMN de uma forma complementar.

Neste trabalho que apresentamos, abordamos justamente a questão dessa forma. Dos requisitos passamos para os Use Cases, depois para os diagramas em BPMN, dos quais conseguimos, em conjunto com os Use Case e Requisitos, desenhar a Base de Dados da aplicação.

3.4.3 Modelo Relacional de Base de Dados

O modelo relacional, pode estar ultrapassado por alguns, mas continua a ser o mais completo para uma aplicação que não vise, uma base de dados LOB e não seja puramente de *reporting*. Para uma aplicação que pretende ser altamente parametrizável, sem perder a ligação e a garantia de coerência de dados, este modelo continua a ser rei. Neste trabalho, como poderá ser visto pelo Anexo 3 e 4, modelamos a base de dados recorrendo ao Modelo Relacional e ao contrário do que é defendido por muitos informáticos hoje em dia, implementamos chaves estrangeiras, o que nos permite ter uma base de dados, que simultaneamente nos garante a coerência e ligação correta dos dados, e também altamente parametrizável. Sendo a parametrização um dos principais objetivos do nosso trabalho, estamos fortemente convictos que, conseguimos obter uma base de dados estável, coerente, e que nos permite uma abordagem no desenho do software totalmente virada para a parametrização total da mesma.

3.4.4 MySQL Workbench

Ferramenta da Oracle que contém todas as funcionalidades disponíveis, com a vantagem de ser *open source* correr também em Linux, para que de uma forma controlada e visual, desenhar uma base de dados relacional, com a vantagem de a qualquer momento do desenho da mesmo, poder ser gerado o *script* em SQL (*forward-engineering*) para podermos testar o desenho da B.D.. No Anexo 4, apresentamos o diagrama da B.D. desenhada precisamente com o auxílio da ferramenta MySQL Workbench, a qual consegue exportar nativamente para formato PDF.

3.4.5 Eclipse Java EE IDE for Web Developers – UML2 Tool Diagramming

Outra ferramenta *open source*, a qual é mais conhecida pela sua vertente de desenvolvimento de software na linguagem de programação Java, contém um SDK para UML, o qual usamos para desenhar os Use Case que apresentamos no Anexo 2. Neste

ponto foi bastante útil e conseguimos de uma forma rápida e expedita desenhar os Use Case, mesmo não tendo muito conhecimento no uso desta SDK.

3.4.6 LibreOffice

Plataforma na qual efetuamos todos os documentos deste projeto. Com características muito semelhantes ao OpenOffice, não fosse o LibreOffice derivar desta última, tem como principal estandarte ser totalmente livre de usar e com todas as funcionalidades que outros Office Suits disponíveis e supostamente superiores.

3.4.7 Microsoft Visio com Addin Process Modeler 5

Para a modelação em BPMN, usamos o Addin Process Modeler 5 em versão de “Evaluation”. Apesar de não termos todas as funcionalidades disponíveis, temos aquelas que necessitávamos para efetuarmos a análise que pretendíamos. Apesar de ser uma ferramenta paga, a hipótese *open source*, addin para o Eclipse, está ainda numa fase de estabilização, o que não nos permitiu o seu uso, levando-nos a usar versões de software pago, mas com licença “Evaluation”.

4 Resultados

Neste ponto, iremos abordar a nossa solução e a forma como desenhamos a aplicação, passando pelas várias componentes levantadas a partir dos requisitos, a procurando mostra a nossa resposta a cada uma dos requisitos do cliente.

4.1 Componente CMMI

Para o desenho da vertente CMMI, tínhamos que deixar a mesma o mais parametrizável possível. Como se poderá constatar pela Base de Dados que apresentamos², ao criarmos a tabela Process dependente da ProcessArea, estamos automaticamente a dizer que podem existir vários processos que podem pertencer à mesma Área de Processo. Da mesma forma, como sabemos que cada Processo pode ter um ou mais objetivos, a tabela

2 Consultar Anexo 4 – Diagrama da Base de Dados

Goal permite isso mesmo e em conjunto com a tabela Permiss permite-nos abranger a totalidade da solução CMMI que queremos implementar. A tabela MaturityLevel permite-nos saber a que nível de maturidade CMMI pertencem os Processos CMMI.

As tabelas Project e Application, respondem diretamente ao requisito 2-CMMI e para além disso, vai permitir uma parametrização de toda a aplicação por Projeto e Aplicação, já que as mesmas são a base de toda a aplicação. De fato, e como se pode ver pelo desenho que efetuamos para o software³ o Projeto e a Aplicação são centrais para a mesma e são chave única para toda a aplicação.

4.2 Componente FPA

De acordo com a informação levantada na análise dos requisitos, os parâmetros de Ajustamento a partir dos quais se calcula o VAF, irão ficar na tabela AdjustmentParameter, que em conjunto com a tabela Adjustment, iremos ter o VAF por Projeto/Aplicação. Como se poderá constatar pelo diagrama, desenhamos a aplicação por forma a que os fatores de ajustamento sejam devidamente balizados no tempo, com a inclusão de duas datas nas tabelas de parametrização (as mesmas datas poderão ser encontradas noutras componentes) que indicam a data de inicia e data fim em que vigoram determinados fatores de ajustamento. Esta abordagem tem a vantagem de mantermos o histórico de fatores de ajustamento na própria tabela onde se encontram os fatores atuais, mas também de parametrizar esses fatores de uma forma pró-ativa e com antecedência relativamente à data em que o novo VAF deverá entrar em produção.

Ainda nesta componente, temos a tabela CountingType, que indicará o tipo de contagem que pode ser efetuado, bem como os tipos de Função de FPA e as tabelas de complexidade dessas mesmas funções. Tal como as tabelas de ajustamento, também estas contêm como chave, o projeto e aplicação, permitindo também diferentes parametrizações para cada projeto/aplicação, mesmo utilizando a mesma instalação deste software numa determinada organização.

3 Consultar o Anexo 5 – Diagramas de Funcionamento da Aplicação

4.3 Componente de Documentação

Pela análise dos requisitos de documentação, foi estabelecido como necessidade, a possibilidade de anexação de documentação, bem como *templates* para processos, e desta forma no nosso desenho da mesma, criamos uma tabela `DocumentType`, que conterá os vários tipos de documentos para o projeto/aplicação, que entrarão como parte dos fluxos de Processos CMMI (que iremos abordar no ponto 4.5). Assim, tendo em conta, que existem soluções de gestão documental bastantes avançadas e que disponibilizam web-services, para que ferramentas como esta que apresentamos consigam comunicar e usar as potencialidades das mesmas, decidimos simplificar o processo da nossa aplicação, e guardar todo e qualquer documento, num outro software, ficando do lado da nossa aplicação, a indicação de como chegar a esse mesmo documento, bem como, quais os *templates* para cada Processo CMMI.

O software de gestão documental Alfresco (a nossa escolha para esta solução), permite-nos de uma forma totalmente integrada com a nossa ferramenta, usar dos seus web-services e deixar essa complexidade para um software que já provou saber fazê-lo muito bem. Sendo um software *open-source*, permite-nos um acesso mais simples à documentação e com isso, apresentarmos uma solução como a que estamos a fazer. O Alfresco, permite cumprir com os requisitos da documentação, já que para além de efetuar a gestão de documentos, tem também um módulo que permite também a gestão de assinaturas eletrónicas para os documentos CMMI que delas necessitem. Na aplicação que agora desenhamos, apenas temos de saber como obter os respetivos documentos associados às várias tarefas de cada Processo CMMI.

4.4 Componente de Utilizadores, Grupos de utilizadores e permissões

Grupos de utilizadores, com permissões diferentes entre eles e a própria gestão de utilizadores, foram os pontos que quisemos responder do desenho da aplicação. Assim, temos na nossa base de dados, a tabela de utilizadores, com um estado que indica a situação de cada “user”, utilizador esse que pode pertencer a vários projetos (ver tabela `UserProject`) e que têm uma determinada função dentro do mesmo. Essa função permite

estabelecer grupos de utilizadores (tabela Role), que terão assignadas permissões para cada Processo CMMI (tabela Permission).

4.5 Componente de Fluxo de Processos

Esta componente aparece-nos referida de forma indireta nos requisitos levantados, principalmente nos requisitos 3-CMMI e 5-CMMI, que levantaram a necessidade de ter várias macro-tarefas que passam por vários Processos e que têm vários estados. Para além disso, surge também a necessidade de estabelecer níveis de acesso aos Processos envolvidos. Assim, a criação da figura de Fluxo de Processos, surge com naturalidade. Este fluxo não é nada mais, nada menos que a definição de uma série de Processos pelo qual passa uma macro-tarefa, no seu processo produtivo. Estas macro-tarefas, foram batizadas de *Requests* e pertencerão apenas a um único Fluxo de Processos, durante a sua vida útil, desde o momento em que são criados.

Pela B.D. é possível verificar que um *Request* pertence a um único Projeto/Aplicação e um Request pertence a um único Fluxo de Processos. Pode passar por vários Processos (e Áreas de Processo) e que um fluxo poderá ter n Processos envolvidos e dentro de cada um deles, terem vários Estados, parametrizáveis de acordo com o Processo. Da mesma forma, poderão existir zero a n documentos por cada Processo (limitado obviamente por Projeto e Aplicação).

Como as permissões dos grupos de utilizadores estão estabelecidas por Processo CMMI, os requisitos que pretendiam o controlo de responsabilidades e acessos ficam respondidos.

Tal como a componente FPA, também os Fluxos de Processos e Estados dos Processos, podem ser balizados no tempo, permitindo, tal como referimos, uma abordagem pró-ativa sobre a ferramenta, fazendo deste software uma parte da solução e não parte do problema, aquando da definição de implementação de novos Processos CMMI no processo produtivo.

4.6 Gestão de *Requests*

Chegados a este ponto, podemos dizer que é na Gestão de *Requests* que poderemos efetuar uma prova de conceito do modelo que desenhamos. É por esta funcionalidade que temos todas as outras tabelas de parametrização que nos dão a possibilidade de gerir as “macro-tarefas” referidas nos requisitos.

Ainda que esta componente pareça de menor importância devido ao pequeno número de tabelas, é aqui que tudo se passa na aplicação. Ou seja, e tentando fazer um exercício concreto, quando é aberto um *Request*, temos de estabelecer um Fluxo, que contém todos os Processos CMMI pelo qual deverá passar esse *Request*, Processos esses que por sua vez terão Permissões para vários Grupos de utilizadores. Esses *Requests*, no seu Fluxo, passarão por Processos que podem ou não ter documentação associada, podem ter n Estados definidos (com um limite mínimo de 2 Estados) e podem ou não ter contagem de *function points*, parametrizável na tabela Process através do indicador booleano *FunctionPointCounting*.

Assim, como se pode entender, conseguimos apresentar uma solução que permite um elevado nível de parametrização e que ao mesmo tempo, permite um elevado nível de liberdade na forma de como implementar a metodologia CMMI numa organização.

5 Trabalho Futuro e Conclusões

5.1 Possíveis desenvolvimentos

No sentido evolutivo desta aplicação, apresentamos várias possibilidades de desenvolvimento que no futuro poderiam ser acrescentadas ao produto que apresentamos, tornando o mesmo mais completo e que englobaria mais vertentes corporativas necessárias para uma organização.

5.1.1 Controlo de tempos

Controlo de tempos de entrada e saída dos utilizadores e também registo de horas gastas por Estado de Processos em *Requests*. Desta forma e em conjunto com os *function points*, poderíamos observar a produtividade dos colaboradores e entender em quais os Processos onde são consumidas mais horas.

5.1.2 Reportes de produtividade

Esta funcionalidade só seria possível, evidentemente, em conjunto com a anterior, já que alguns dos dados necessários, não estão a ser atualmente armazenados na B.D. da aplicação. Assim, como referimos no ponto anterior, poderiam ser desenvolvidos vários tipos de Reportes, que poderiam ajudar as equipas de gestão a melhorar o seu processo Produtivo.

5.1.3 Consultas dos Logs

Atualmente a aplicação que apresentamos já regista dados de alterações á B.D. bem como dados de acessos de utilizadores à aplicação, ainda que não haja a possibilidade de os consultar sem acessos diretos á B.D.. Consideramos que normalmente é uma funcionalidade que não é necessária desenvolver, já que é muito útil para a equipa de desenvolvimento da aplicação, mas pouco útil ao utilizador final. De qualquer forma, é uma funcionalidade que uma vez desenvolvida teria sempre alguma utilização.

5.2 Conclusões

Numa última abordagem a este trabalho, consideramos que conseguimos atingir os objetivos a que nos propusemos, para além de termos respondido a todos os requisitos e critérios de aceitação⁴ da solução. Analisando os critérios de aceitação levantados, poderemos ver que, relativamente ao critério 1-CA conseguimos responder totalmente com a criação dos conceitos de *Request* e Fluxo, que como mostrámos, trata-se da gestão das macro-tarefas enunciadas nos requisitos. Disponibilizamos a possibilidade de parametrizar os vários Estados dos Processos e o acesso dos utilizadores aos Processos e Contagem de FP está devidamente acautelado através das Permissões para Grupos de utilizadores.

O segundo critério de aceitação 2-CA, refere a capacidade de contagem de FP devidamente integrada com os Processos CMMI. Ora bem, se relembrar-mos a solução apresentada iremos verificar que a indicação de necessidade de Contagem de FP, está na componente CMMI, nomeadamente na tabela de Processos, sendo que desta forma e dado que os Processos se encontram devidamente parametrizados nos Fluxos de Processos para serem usados pelos utilizadores nos *Requests*, estamos a responder a todo o critério de aceitação definido.

Finalmente o terceiro ponto, 3-CA que refere a necessidade de gestão de documentos, a implementação da ferramenta Alfresco permite-nos responder a um terço do critério, sendo os outros dois terços do critério respondidos para pela forma como associamos os documentos para cada Fluxo/Processo, que como se sabe, será usado nos *Requests*.

Quanto aos nossos próprios objetivos, conseguimos facilmente verificar que a aplicação que apresentamos, está preparada para ser totalmente parametrizada, respondendo assim a um dos pontos que consideramos mais importante, que passa pelo posicionamento desta aplicação como parte da solução e não como parte do problema, tal com referimos na Visão Geral do Produto. A vertente colaborativa é garantida pela tecnologia escolhida, já que o software é acedido por *browser* e sem nenhuma instalação de software, e a possibilidade de usar outros modelos de CMMI é garantida pela possibilidade de parametrização que disponibilizamos na ferramenta.

4 Consultar Anexo 1 - Levantamento de Requisitos

Bibliografia

DEKKERS, 2010: Carol Dekkers, Jeopardy for IT projects... the answer is July 15, 2010, <http://musingsaboutsoftwaredevelopment.wordpress.com/2010/12/08/jeopardy-for-it-projects-the-answer-is-july-15/>

OMG, 2010: OMG, BPMN 2.0 by Example, 2010, <http://www.omg.org/spec/BPMN/20100601/10-06-02.pdf>

BPM OFFENSIVE BERLIN, 2012: Berliner BPM-Offensive, BPMN 2.0 - Notação e Modelo de Processo de Negócio - Poster, 2012, http://www.bpmb.de/images/BPMN2_0_Poster_PT.pdf

CMMI-DEV, V1.3, 2010: CMMI Product Team, CMMI® for Development, Version 1.3, 2010, www.sei.cmu.edu/reports/10tr033.pdf

CMMI-SVC, V1.3, 2010: CMMI Product Team, CMMI® for Services, Version 1.3, 2010, www.sei.cmu.edu/reports/10tr034.pdf

CMMI-ACQ, V1.3, 2010: CMMI Product Team, CMMI® for Acquisition, Version 1.3, 2010, www.sei.cmu.edu/reports/10tr032.pdf

CMMI-DEV 1.3 Poster, 2010: methodpark, CMMI® – CAPABILITY MATURITY MODEL® INTEGRATION - Poster, 2010, http://www.methodpark.de/fileadmin/downloads/resources/CMMI_for_Development_high.pdf

IFPUG - 4.2, 2004: International Function PointUsers Group (IFPUG), Function Point CountingPractices Manual - Release 4.2, 2004,

SMS, 2004: Software Management Solutions, Inc., Counting Function Points Reference Guide - IFPUG CPM Version 4.1, 2004,

IFPUG - 4.3.1, 2010: International Function PointUsers Group (IFPUG), Function Point CountingPractices Manual - Release 4.3.1, 2010, <http://www.scribd.com/doc/77134872/0004-3-1-Part-0-2010-01-17>

ALEXANDRE, HABRA, 2010: Simon Alexandre, Naji Habra, UML Modeling of Five Process Maturity Models, 2003, <http://www.cetic.be/IMG/pdf/UMLModelingOfFiveProcessModels-V1-3.pdf>

KARONA, 2010: karona, The «include» and «extend» Relationships in Use Case Models, 2010, [UseCases_IncludesAndExtends.pdf](http://www.karona.com/UseCases_IncludesAndExtends.pdf)

MYSQL WORKBENCH, 2012: Oracle, MySQL Workbench - Manual, 2012, <http://downloads.mysql.com/docs/workbench-en.pdf>

Anexos

Anexo 1 – Levantamento de Requisitos

Este documento visa o levantamento de todos os requisitos envolvidos no projeto, funcionais e não-funcionais, bem como os potenciais *stakeholders* envolvidos e critérios de aceitação do projeto.

Anexo 2 – Use Cases

Poder-se-á encontrar neste documento todos os Use Cases levantados para o projeto, desenhados com a metodologia UML, por forma a responder aos requisitos levantados.

Anexo 3 – Desenho da Base de Dados

Este documento contém informação sobre a Base de Dados que foi desenhada para o projeto, com uma explicação sobre as opções efetuadas sobre a mesma, bem como os *scripts* em SQL (para B.D. em *MySQL*), para criação da Base de Dados.

Anexo 4 – Diagrama da Base de Dados

Este documento contém o Diagrama da Base de Dados, em formato A2. Mais informação pode se obtida através da consulta do Anexo 3.

Anexo 5 – Diagramas de Funcionamento da Aplicação

Este documento contém os diagramas de funcionamento da aplicação, desenvolvidos em BPMN.

Glossário

Aplicação – Conjunto de software que responde funcionalmente aos requisitos dos utilizadores finais numa determinada área, por exemplo, aplicação de Clientes, Financeira ou Contabilidade

Fluxo – Caracterização dos vários processos CMMI pelo qual passa um *request*

Projeto – Conjunto de aplicações e/ou conjunto de software. Normalmente está associado a um contrato com um cliente, podendo se de manutenção ou novo desenvolvimento

Request – Entende-se por *request* uma macro-tarefa, composto por várias tarefas (processos), que se usará para o fluxo do processo CMMI e que passará por vários estados e pelos vários processos definidos na ferramenta.

- i <http://www.sei.cmu.edu/cmmi/>
- ii <http://www.ifpug.org/>
- iii <http://www.sei.cmu.edu/solutions/softwaredev/>
- iv <http://www.cmu.edu/index.shtml>
- v http://pt.wikipedia.org/wiki/Capability_Maturity_Model
- vi <http://pt.wikipedia.org/wiki/CMMI>
- vii <http://www.ifpug.org/about/>
- viii <https://www.ifpug.org/>
- ix <http://www.qualityplustech.com/>
- x Mais informação sobre Carol Dekkers no sítio: <http://www.qualityplustech.com/carol.html>