

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Ferramenta de Gestão de Projetos: Integração de CMMI, TSP e Scrum

Sónia Cristina Palma Liquito

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Doutor João Carlos Pascoal de Faria

Julho de 2012

Ferramenta de Gestão de Projetos: Integração de CMMI, TSP e Scrum

Sónia Cristina Palma Liquito

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Doutor Raul Fernando de Almeida Moreira Vidal

Vogal/Arguente: Doutor Ricardo Jorge Silvério Magalhães Machado

Vogal/Orientador: Doutor João Carlos Pascoal de Faria

Julho de 2012

Resumo

Este relatório foi desenvolvido no âmbito da unidade curricular da Dissertação e apresenta um trabalho de investigação, conceção e implementação desenvolvido ao longo de um semestre.

O objetivo deste trabalho é a definição e conceção de um dos componentes pertencentes a uma ferramenta eficaz na gestão do ciclo de vida dos projetos de desenvolvimento de software, a desenvolver no âmbito do projeto AIMS, financiado pelo QREN, e envolvendo a FEUP, a Strongstep e a Multicert.

Esta ferramenta será disponibilizada segundo o modelo SaaS (*Software as a Service*) e baseada na metodologia AIM (*Accelerated Improvement Method*), desenvolvida pelo *Software Engineering Institute* da *Carnegie Mellon University*, com vista à rápida implementação de práticas CMMI (*Capability Maturity Model Integration*) recorrendo ao TSP (*Team Software Process*), às avaliações SCAMPI e aos elementos do *Six-Sigma* como tecnologias nucleares. O AIM facilita uma rápida implementação de práticas de CMMI de nível 2 e 3, e proporciona um suporte significativo para a adoção de níveis de maturidade mais elevados. O CMMI funciona como um guia do ponto de vista estrutural e organizacional, enquanto o TSP e o *Six-Sigma* concretizam as práticas definidas inicialmente. No projeto AIMS, e em particular no componente de gestão de projetos, dada a apetência do mercado pela metodologia Scrum e dada a elevada complementaridade e compatibilidade entre as metodologias Scrum e TSP, pretende-se oferecer um suporte integrado para estes dois processos, tirando partido das melhores características de cada um deles.

Foi concebido e parcialmente implementado em Ruby on Rails um componente de gestão de projetos sobre o Redmine¹ com suporte integrado para as metodologias TSP e Scrum, integração da gestão de projetos com a gestão de processos, disponibilização de assistentes de planeamento (*wizards*) para a geração automática de uma lista de tarefas e cálculo de estimativas de esforço e prazos e, no futuro, com suporte para gestão de desempenho organizacional. Na prática, a usabilidade também é um dos fatores em causa, e para tal foram implementados mecanismos de *drag and drop* e de edição de campos *in place*. Espera-se que esta ferramenta ajude a organizar a eficiência na calendarização das tarefas inerentes.

Este relatório faz uma breve análise das definições da metodologia AIM, das práticas do CMMI e da implementação destas através de TSP e Scrum. Também apresenta uma comparação entre ferramentas de gestão de projetos já existentes no mercado e uma solução de implementação para o componente de gestão de projetos em causa.

¹Redmine é uma ferramenta de gestão de projetos selecionada como base para o projeto AIMS.

Abstract

This report was developed in the course of Master's Thesis and it presents a research, design and implementation work which was developed during one semester.

The goal of this work is the definition and conception of a component belonging to a management tool of the software development projects life cycle, to develop under the AIMS project, funded by QREN, and involving the FEUP, the Strongstep and Multicert.

This tool is based on methodology AIM (Accelerated Improvement Method), developed by Carnegie Mellon Software Engineering Institute, in order to do a quick implementation of CMMI practices with high level and also covering the TSP (Team Software Process), SCAMPI assessment and Six-Sigma elements as nuclear technology. AIM provides a quick implementation of practices of CMMI (Capability Maturity Model Integration) level 2 and 3, and provides significant support for the adoption of higher levels of maturity. CMMI serves as a guide in terms of structural and organizational, as the TSP and Six-Sigma materialize the initially defined practices.

This tool will be available under the SaaS (Software as a Service) model and based on the methodology AIM (Accelerated Improvement Method), developed by Software Engineering Institute of Carnegie Mellon University, with a view to early implementation of CMMI (Capability Maturity Model Integration) practices using the TSP (Team Software Process), SCAMPI assessments and the elements of Six-Sigma as nuclear technologies. The AIM provides a fast implementation of CMMI practices level 2 and 3, and it provides significant support for the adoption of maturity higher levels. The CMMI serves as a guide for the structural and organizational point of view, as the TSP and the Six Sigma get the practices initially defined. In the AIMS project, and in particular the component of project management, given the attractiveness of the market by the Scrum methodology and given the high complementarity and compatibility between the TSP and Scrum methodologies, it is intended to provide an integrated support for these two processes, taking advantage of each best features.

It was designed and partially implemented in Ruby on Rails a component of project management on the Redmine² with the concepts of TSP and Scrum, project management integration with process management and (in the future) with organizational performance management, planning wizards availability for automatic generation of tasks and efforts and dates estimated. In practice, the usability is also one of the factors concerned, and such mechanisms have been implemented with drag and drop functionalities and editing fields in place. It is hoped that this tool will help to organize the efficient scheduling of involved tasks.

This report reviews the definitions of the AIM methodology, the CMMI practices and implementation of these practices using TSP and Agile. It also presents a comparison of some project management tools on the market and a deployment solution for the projects management component in question.

²Redmine is a basic tool for project management selected for the AIMS project

Conteúdo

1	Introdução	1
1.1	Motivação e enquadramento	1
1.2	Objetivos e aspetos inovadores	3
1.3	Metodologia e planeamento	4
1.4	Estrutura do relatório	6
2	Análise de Metodologias	7
2.1	AIM	7
2.2	CMMI	8
2.3	TSP	11
2.3.1	Breve introdução	11
2.3.2	Mapeamento com CMMI	11
2.4	Scrum	13
2.4.1	Breve introdução	13
2.4.2	Mapeamento com CMMI	15
2.4.3	Mapeamento com TSP	15
3	Análise e Seleção de Ferramentas	19
3.1	Requisitos	19
3.2	Ferramentas candidatas	19
3.2.1	Jira	21
3.2.2	Visual Studio Team Foundation Server 2010	22
3.2.3	Process Dashboard	22
3.2.4	Redmine	23
3.2.5	Comparação entre as ferramentas e estratégias candidatas	23
3.3	Solução adotada	27
4	Conceção e Implementação	29
4.1	Mapeamento entre conceitos de TSP, Scrum, Redmine e AIMS	29
4.2	Modelo de domínio	30
4.3	<i>User Stories</i>	30
4.4	Tecnologias e arquitetura	32
4.5	Modelo de dados	33
4.6	Módulos da solução	36
4.6.1	Processos	36
4.6.2	Visão Geral	36
4.6.3	Equipa	37
4.6.4	<i>Roadmap</i>	37

CONTEÚDO

4.6.5	Tarefas	37
4.6.6	<i>Backlog</i>	38
4.6.7	Produtos	38
4.6.8	Atividade	38
4.6.9	Gráficos	39
4.6.10	Definições	41
5	Apresentação e Avaliação da Solução	43
5.1	Cenário implementado	43
5.2	Validação da solução	49
6	Conclusões	51
6.1	Resultados alcançados	51
6.2	Trabalho futuro	53
A	Elementos do TSP	55
B	Relação das práticas do CMMI com elementos do TSP	61
C	Relação das práticas do CMMI com práticas do Scrum	71
	Referências	75

Lista de Figuras

1.1	Enquadramento do componente de gestão de projetos na ferramenta AIMS	3
1.2	Processo Scrum de Scrum	5
1.3	Mapa de Gantt para a fase de desenvolvimento	5
1.4	Etapas de cada iteração semanal correspondente à iteração 3	6
2.1	Componentes da metodologia AIM [CM12]	8
2.2	Impacto das práticas do TSP nas áreas de processo do CMMI com nível de maturidade 2 [MW05]	12
2.3	Mapeamento dos conceitos de TSP com os conceitos de Scrum [Roy08] [DavPAa]	17
2.4	Mapeamento dos ciclos do TSP com as iterações do Scrum [DavPAa]	17
2.5	Mapeamento do planeamento de TSP com o plano a ser seguido no Scrum [DavPAa]	18
3.1	Representação das ferramentas selecionadas para análise de acordo com a importância relativa para o projeto no formato de uma <i>Word Cloud</i> , usando o programa <i>Wordle</i>	21
3.2	Tabela de comparação entre as ferramentas: JIRA [Jir12a], Visual Studio Team Foundation Server 2010 [Tfs12], Clarizen [Cla12] e Redmine [Red12a]	24
3.3	Tabela informativa sobre Scarab, Process Dashboard, Redmine, Collabtive, Embeta Forge, Plandora Project	25
3.4	Tabela de comparação entre o JIRA, a opção de desenvolver uma ferramenta de raiz, o Visual Studio Team Foundation Server 2010 e o Redmine [Jir12a] [Tfs12] [Red12b]	26
4.1	Comparação de conceitos entre TSP, Scrum e Redmine e apresentação dos conceitos do <i>plugin</i> AIMS	29
4.2	Modelo de domínio dos componentes de gestão de projetos e processos	30
4.3	Lista de <i>user stories</i> criada para o Projeto AIMS	31
4.4	Arquitetura em alto-nível do Redmine e do <i>plugin</i> AIMS	33
4.5	Modelo de dados do Redmine e <i>plugin</i> AIMS	35
4.6	Exemplo de um gráfico com o cálculo do <i>earned value</i> [EVi12]	39
4.7	Exemplo de um gráfico <i>burndown</i> ao longo de várias iterações [Coh08]	40
4.8	Exemplo de um gráfico simples de <i>resource burnup</i> para uma iteração	41
5.1	Formulário para criação de um projeto (módulo Visão Geral)	44
5.2	Vista de todos os projetos públicos	44
5.3	Vista e formulário para criação de processos (módulo Processos)	44
5.4	Formulário para edição de projetos (módulo Definições)	45
5.5	Vista das informações de um processo (módulo Processos)	45
5.6	Vista do conjunto de passos de um subprocesso (módulo Processos)	46

LISTA DE FIGURAS

5.7	Vista do módulo <i>Backlog</i> e do respetivo conjunto de iterações	46
5.8	Formulário para criação de uma <i>issue</i> de qualquer tipo	47
5.9	Vista do módulo Produtos e do respetivo conjunto de iterações	48
5.10	Vista do módulo Visão Geral e respetivas informações do projeto	48
5.11	Vista do módulo Equipa	49
A.1	Tabela 1 de <i>Scripts</i> [MW05]	55
A.2	Tabela 2 de <i>Scripts</i> [MW05]	56
A.3	Tabela 1 de Formulários e outros artefatos [MW05]	56
A.4	Tabela 2 de Formulários e outros artefatos [MW05]	57
A.5	Tabela de Papéis [MW05]	57
A.6	Tabela com outros elementos [MW05]	58
A.7	Tabela de ações de formação [MW05]	59
B.1	Tabela 1 sobre a área de processo: Planeamento de Projetos (PP) [MW05]	63
B.2	Tabela 2 sobre a área de processo: Planeamento de Projetos (PP) [MW05]	63
B.3	Tabela 3 sobre a área de processo: Planeamento de Projetos (PP) [MW05]	64
B.4	Tabela 4 sobre a área de processo: Planeamento de Projetos (PP) [MW05]	64
B.5	Tabela 5 sobre a área de processo: Planeamento de Projetos (PP) [MW05]	65
B.6	Tabela 6 sobre a área de processo: Planeamento de Projetos (PP) [MW05]	65
B.7	Tabela 1 sobre a área de processo: Monitorização e Controlo de Projetos (PMC) [MW05]	66
B.8	Tabela 2 sobre a área de processo: Monitorização e Controlo de Projetos (PMC) [MW05]	66
B.9	Tabela 3 sobre a área de processo: Monitorização e Controlo de Projetos (PMC) [MW05]	67
B.10	Tabela 4 sobre a área de processo: Monitorização e Controlo de Projetos (PMC) [MW05]	67
B.11	Tabela 5 sobre a área de processo: Monitorização e Controlo de Projetos (PMC) [MW05]	68
B.12	Tabela 1 sobre a área de processo: Gestão de Requisitos (REQM) [MW05]	68
B.13	Tabela 2 sobre a área de processo: Gestão de Requisitos (REQM) [MW05]	68
B.14	Tabela 3 sobre a área de processo: Gestão de Requisitos (REQM) [MW05]	69
B.15	Tabela 1 sobre a área de processo: Gestão de Acordo com o Fornecedor (SAM) [MW05]	69
B.16	Tabela 2 sobre a área de processo: Gestão de Acordo com o Fornecedor (SAM) [MW05]	70
B.17	Tabela 3 sobre a área de processo: Gestão de Acordo com o Fornecedor (SAM) [MW05]	70
C.1	Tabela de métodos para a Gestão de Requisitos do CMMI através do Scrum [PS09]	71
C.2	Tabela 1 de métodos para o Planeamento de Projetos do CMMI através do Scrum [PS09]	72
C.3	Tabela 2 de métodos para o Planeamento de Projetos do CMMI através do Scrum [PS09]	72
C.4	Tabela de métodos para a Monitorização e Controlo de Projetos do CMMI através do Scrum [PS09]	73

Lista de Tabelas

3.1	Lista de requisitos de alto-nível	20
4.1	Listagem e descrição de cada uma das tabelas apresentadas no modelo de dados da Figura 4.5	34
B.1	Terminologia da classificação atribuída nos mapeamentos das figuras seguintes [MW05]	62

LISTA DE TABELAS

Abreviaturas e Símbolos

AIM	<i>Accelerated Improvement Method</i>
CM	<i>Configuration Management</i>
CMM	<i>Capability Maturity Model</i>
CMMI	<i>Capability Maturity Model Integration</i>
CMMI-ACQ	<i>CMMI for Acquisition</i>
CMMI-DEV	<i>CMMI for Development</i>
CMMI-SVC	<i>CMMI for Services</i>
CMU	<i>Carnegie Mellon University</i>
EV	<i>Earned Value</i>
FEUP	<i>Faculdade de Engenharia da Universidade do Porto</i>
IPPD	<i>Integrated Product and Process Development</i>
MA	<i>Measurement and Analysis</i>
PMC	<i>Project Monitoring and Control</i>
PP	<i>Project Planning</i>
PPQA	<i>Process and Product Quality Assurance</i>
PROBE	<i>PROxy Based Estimation</i>
PSP	<i>Personal Software Process</i>
QREN	<i>Quadro de Referência Estratégico Nacional</i>
REQM	<i>Requirements Management</i>
SaaS	<i>Software as a Service</i>
SAM	<i>Supplier Agreement Management</i>
SEI	<i>Software Engineering Institute</i>
TSP	<i>Team Software Process</i>
WWW	<i>World Wide Web</i>

Capítulo 1

Introdução

Este capítulo apresenta o enquadramento e a motivação do trabalho, assim como identifica e define os problemas que o trabalho desta dissertação aborda. Também refere a estrutura deste relatório com um breve resumo de cada um dos capítulos posteriores e o planeamento e metodologia de trabalho adotada.

1.1 Motivação e enquadramento

Esta secção enquadra o trabalho proposto para dissertação de mestrado no âmbito do projeto AIMS, promovido pelo Consórcio AIMS composto pela Strongstep¹, FEUP e Multicert² e com apoio do QREN³. O projeto AIMS tem por objetivo o desenvolvimento de uma plataforma *Software as a Service* de gestão do ciclo de vida do software, suportando a metodologia AIM desenvolvida pelo CMU-SEI. O componente em que se foca o presente trabalho de dissertação é o de gestão de projetos suportando o TSP (*Team Software Process*). Este componente deve funcionar de forma integrada com os restantes componentes da ferramenta, nomeadamente com os componentes de gestão de processos e gestão de desempenho. O trabalho envolve a seleção de ferramentas de base e o desenvolvimento de uma parte desse componente com base nessas mesmas ferramentas.

Atualmente existe uma percentagem elevada de projetos de software cancelados e ainda mais são aqueles que não cumprem com os objetivos estipulados inicialmente. Como principais causas apontadas para o insucesso dos projetos de desenvolvimento de software é possível referir as metas de projeto irreais, estimativa de recursos mal definida, má definição dos requisitos do sistema, acompanhamento desajustado do processo de desenvolvimento, riscos não considerados,

¹Strongstep é uma empresa especializada em Engenharia de Software que contribui para a melhoria de qualidade de software em Portugal e no Mundo [Str12].

²Multicert é a primeira entidade de certificação portuguesa credenciada pela autoridade credenciadora a disponibilizar o pedido de emissão de certificados qualificados (certificados com o mais alto grau de segurança e de confiança), de acordo com a legislação nacional e europeia [Mul12].

³QREN é um quadro de referência estratégico e nacional que valoriza o conhecimento, a ciência, a tecnologia e a inovação, bem como a promoção de níveis elevados e sustentados de desenvolvimento económico e sócio-cultural e de qualificação territorial [QRE12].

Introdução

má comunicação com o cliente, utilização de tecnologias pouco maduras ou uma gestão pouco eficiente dos projetos de desenvolvimento [MCM10]. Para além disso, a qualidade, apesar de ser um aspeto cada vez mais crítico no desenvolvimento de software, é muitas vezes ignorada até à fase de testes dos produtos de desenvolvimento, na qual a maioria dos defeitos é identificada quando os custos para correção são elevados ou as metodologias aplicadas não são eficientes. Tais práticas resultam frequentemente em produtos defeituosos, trabalho adicional desnecessário, custos de desenvolvimento inflacionados, atrasos inesperados na execução dos projetos, falhas nos sistemas e vulnerabilidade ao nível da segurança [MCM10] [Sch10]. Desta forma, para uma gestão eficaz de um projeto de desenvolvimento de software é importante controlar questões como o âmbito, o custo e o prazo, e tal só é possível através da aplicação de práticas profissionais de gestão. Por outro lado, um projeto de desenvolvimento de software requer a utilização de práticas de engenharia eficientes, que garantam soluções de qualidade sustentáveis e adaptadas à utilização social e humana. A utilização de práticas eficientes de engenharia e gestão exige assim a utilização de ferramentas adequadas nos processos de desenvolvimento de software, com o máximo de automatização possível.

As ferramentas de gestão de projetos existentes hoje em dia não estão orientadas para as pequenas e médias empresas, nem têm o nível adequado de flexibilidade ou facilitam a implementação dos processos de forma automatizada [Sch10]. Visto isto, o projeto AIMS visa a criação de uma ferramenta de apoio à gestão do ciclo de vida de projetos de desenvolvimento de software suportada pela metodologia AIM que aborda os princípios do TSP integrados com as práticas do CMMI (*Capability Maturity Model Integration*) na gestão de projetos e processos e também recorre ao *Six-Sigma* para tratamento estatístico de dados fornecidos. Aliado ao TSP, também recorre às metodologias ágeis, nomeadamente Scrum.

A Figura 1.1 indica a relação entre os três principais componentes da ferramenta que se encontra a ser desenvolvida. O componente elaborado ao longo do trabalho de dissertação é o que se encontra encaixilhado a negrito do lado esquerdo. Estes três componentes estão organizados da seguinte forma:

- **Gestão de Projetos**, que permite a gestão dos projetos e contém assistentes de planeamento, mantendo o ponto de vista individual, de equipa e da organização. Para além disso, este componente também se encontra a ser integrado com um outro responsável pelos processos e mais tarde será integrado com um componente de gestão de *plugins* e *add-ons* na gestão de requisitos, configuração e documentação. Também irá suportar o uso de formulários, gráficos e relatórios de acordo com as definições do TSP e Scrum, bem como a monitorização de tempo, defeitos e prazos. Numa segunda fase da elaboração desta ferramenta, também irá integrar com um *plugin* para a revisão de documentos do *Microsoft Office*, bem como suportar a gestão de testes, a integração com o *Microsoft Outlook* e com um ambiente gráfico.
- **Gestão de Processos**, que permite a definição de processos através da definição dos seus *scripts*, papéis e passos, juntamente com a apresentação de propostas de melhoria de cada

Introdução

um dos processos. Também irá suportar o controlo de versões e o mapeamento com as práticas do CMMI, através da integração com as definições *out-of-the-box* para TSP e Scrum e da partilha colaborativa de dados.

- **Gestão de Desempenho**, que irá recorrer a técnicas de *Six-Sigma* para monitorizar e analisar os gráficos gerados e todos os dados estatísticos fornecidos, também através da integração com as definições *out-of-the-box* para TSP.

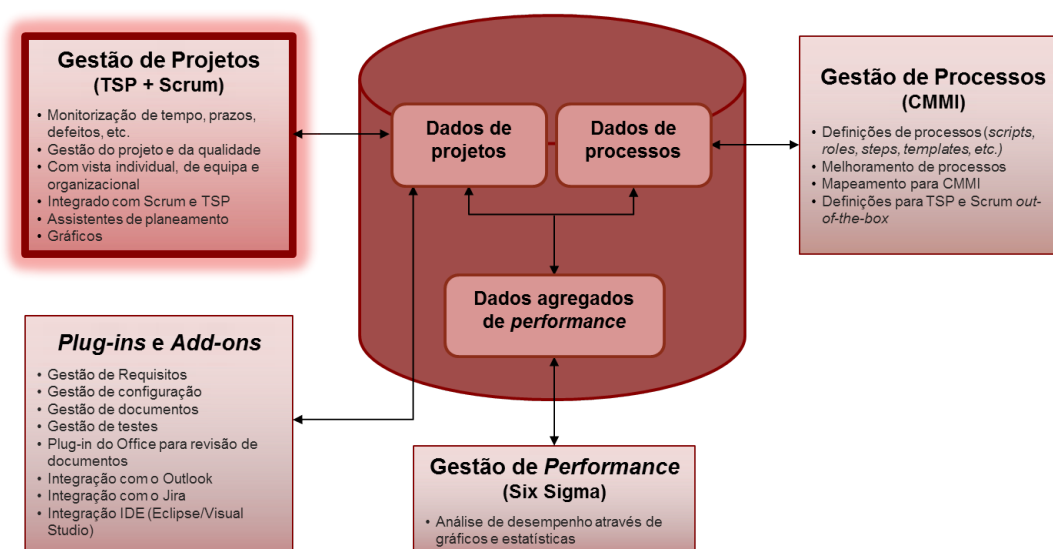


Figura 1.1: Enquadramento do componente de gestão de projetos na ferramenta AIMS

1.2 Objetivos e aspetos inovadores

Os objetivos deste trabalho focam-se no desenvolvimento de um componente que irá ser integrado com os restantes componentes de modo a formar uma ferramenta de gestão de projetos eficaz e flexível. O foco da investigação e do aspeto inovador para este trabalho está na descoberta da melhor forma de utilizar o TSP para implementar as práticas do CMMI em projetos de nível de maturidade 2, juntamente com o uso de Scrum, que será explicado no capítulo seguinte.

De seguida é apresentada a lista de objetivos gerais definidos para a implementação deste componente. É importante referir que esta lista foi previamente validada pelo Consórcio AIMS e serviu como base para a definição e validação dos requisitos e dos critérios de comparação de ferramentas concorrentes.

- Permitir a monitorização de tempo, prazos, defeitos, etc.;
- Permitir a gestão do projeto e da sua qualidade;
- Conter uma vista individual, de equipa e organizacional sobre cada projeto;
- Suportar metodologias ágeis, como o Scrum;

- Integrar com as práticas definidas no CMMI;
- Suportar os formulários, gráficos e relatórios baseados nos parâmetros do TSP;
- Integrar com outras ferramentas de gestão de projetos;
- Suportar integração entre Scrum e TSP;
- Integrar gestão de projetos com gestão de processos.

De notar que o prazo limite previsto no projeto AIMS para o desenvolvimento deste componente excede a duração da dissertação, pelo que esta se concentrou na conceção do componente e na implementação de algumas das características mais inovadoras como o suporte para Scrum, a integração deste com o TSP, e a integração da gestão de projetos com a gestão de processos.

1.3 Metodologia e planeamento

Para o desenvolvimento deste componente de gestão de projetos foi adotada uma metodologia ágil de desenvolvimento de software com base no método Scrum.

O Scrum é um processo ágil de desenvolvimento de software e gestão de projetos que permite focar na entrega do maior valor de negócio no mais curto espaço de tempo. Permite avaliar rapidamente e repetidamente o trabalho real de software desenvolvido a cada iteração, que pode ser de duas semanas a um mês. Neste processo é o negócio quem estabelece as prioridades. A cada duas semanas a um mês qualquer pessoa pode ver parte do produto final desenvolvido e optar por validá-lo ou continuar a melhorar por mais duas semanas a um mês.

Esta metodologia foi escolhida, uma vez que foi necessário que o desenvolvimento fosse feito de forma iterativa para possibilitar a avaliação no final de cada iteração do trabalho realizado e planear o trabalho seguinte, de forma mais realista do que se fosse definido logo todo de início, como sugerem alguns métodos tradicionais (por exemplo, modelo em cascata). Uma vez que se trata de um trabalho ambicioso e com bastantes funcionalidades, foi importante reajustar no final de cada iteração as *user stories* definidas e abandonar algumas mais irrelevantes.

No caso deste componente, foi utilizado uma espécie de Scrum de Scrum onde existiram três iterações, sendo que cada iteração correspondeu a um mês e terminou com uma reunião mensal com o consórcio AIMS para validação do produto desenvolvido até ao momento. Semanalmente, tal como indica a Figura 1.2, houve uma reunião com o orientador e os restantes membros da equipa para monitorização do trabalho desenvolvido durante cada semana, acompanhamento da documentação criada, esclarecimento de dúvidas e planeamento da semana seguinte. E por fim, houve algumas reuniões diárias entre os elementos responsáveis pela implementação dos componentes do projeto (gestão de projetos e gestão de processos).

Ao longo do primeiro semestre, no âmbito da unidade curricular de Preparação da Dissertação, foi analisado o estado da arte e definidos os requisitos. Com este trabalho previamente desenvolvido, foram necessárias três iterações para desenvolver o restante trabalho de dissertação, tal como

Introdução



Figura 1.2: Processo Scrum de Scrum

indica o seguinte mapa de Gantt (Figura 1.3), que teve início na terceira semana de Fevereiro de 2012, mais precisamente dia 13 de Fevereiro, e irá terminar em Julho de 2012.

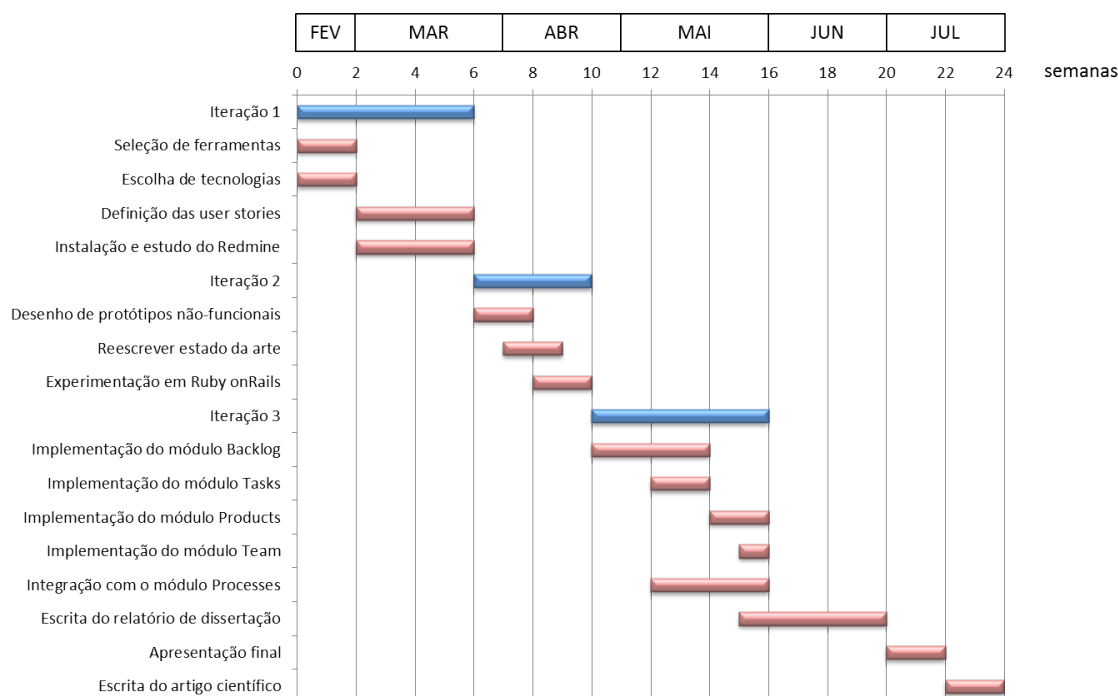


Figura 1.3: Mapa de Gantt para a fase de desenvolvimento

Numa primeira fase do projeto foi definido um conjunto de *user stories* para o desenvolvimento do componente de gestão de projetos, validado pelo Consórcio AIMS.

Enquanto que a primeira iteração foi para escolha de tecnologias, a segunda iteração destinou-se ao desenho da solução e a terceira iteração teve como objetivo o desenho e implementação de alguns protótipos funcionais sob a forma de um *plugin* para a ferramenta base. O mês de Junho foi dedicado exclusivamente à escrita do relatório da dissertação, que já tinha vindo a ser escrito ao longo do semestre.

Cada iteração teve uma duração diferente e seguiu um conjunto de etapas adaptado ao trabalho a ser desenvolvido em cada. O mais relevante de referir, é o caso da iteração 3, implementação do *plugin* AIMS, que seguiu o seguinte conjunto de etapas, tal como indica a Figura 1.4.

Introdução

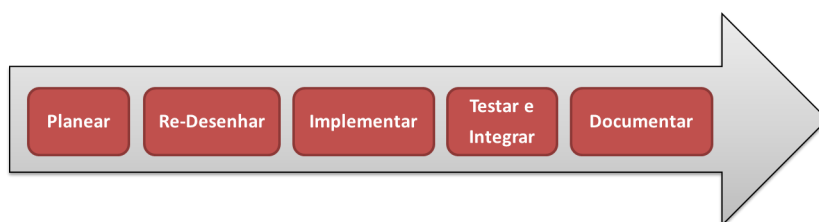


Figura 1.4: Etapas de cada iteração semanal correspondente à iteração 3

Ao longo desta iteração, foi efetuado um planeamento de cada semana no início de todas as semanas, de forma a selecionar as *user stories* a implementar segundo as suas prioridades. Depois deste planeamento, foi quase sempre necessário refazer o desenho da solução pensado na iteração 2. A etapa seguinte foi a mais longa tratando-se da implementação dos módulos e respetivas funcionalidades em cada semana. Depois da implementação seguiram-se os respetivos testes de usabilidade feitos pela própria equipa de desenvolvimento, que, devido à falta de tempo, ocorreram em simultâneo com a integração do componente de gestão de projetos e processos. Por fim, existiu a fase de integração com a versão estável no servidor de produção e em Setembro de 2012 será integrado na empresa Multicert, isto quando as funcionalidades definidas já estiverem completamente implementadas e testadas.

Antes de se dar a iteração por encerrada, foi sendo feita uma breve documentação de todo o processo para que tudo ficasse registado antes de algum pormenor ficar esquecido.

1.4 Estrutura do relatório

Para além da introdução, este relatório de dissertação contém mais cinco capítulos.

No capítulo 2 são apresentados alguns conceitos básicos essenciais para a compreensão da área em que o projeto se insere e a forma como se integram com o projeto.

O capítulo 3 apresenta uma lista dos requisitos definidos em parceria com a Strongstep e a Multicert, bem como uma análise mais detalhada de ferramentas existentes nesta área e candidatas para este projeto.

No capítulo 4 é apresentado uma lista de *user stories* e um modelo de domínio e de dados que ajudaram à implementação do projeto.

O capítulo 5 mostra de que forma a solução final está a ser implementada e em que estado se encontra.

No capítulo 6 são apresentadas as conclusões retiradas ao longo da conceção da solução e é feita uma análise do trabalho futuro.

Capítulo 2

Análise de Metodologias

Neste capítulo são descritos todos os conceitos necessários e a forma como se integram para uma melhor compreensão do estudo do projeto.

2.1 AIM

A metodologia AIM (*Accelerated Improvement Method*) foi criada pelo SEI-CMU com o objetivo de ajudar as organizações a implementar rapidamente, com segurança e com ganhos efetivos de desempenho as práticas de CMMI, recorrendo ao TSP e ao Six-Sigma.

Esta metodologia é composta por cinco elementos chave, tal como indica a Figura 2.1, que trabalham em conjunto para alcançar determinados objetivos, tais como usar uma estratégia para um melhor e mais rápido desenvolvimento, adaptada para cada equipa, e utilizando como modelo de referência para as melhores práticas a versão atual do CMMI-DEV, mais especificamente os objetivos e as práticas definidas nos níveis de maturidade 2 e 3. Também recorre ao TSP, começando com formação em PSP (*Personal Software Process*), que fornece aos indivíduos e equipas de projeto, incluindo a equipa de processos, uma estrutura transparente e operacional do processo, medição e gestão de práticas coerentes com o CMMI. E usa métodos da disciplina *Six Sigma* para analisar os dados operacionais e, em seguida, identificar e agir sobre oportunidades de melhoria do processo usado [MCM10].

Apesar da ideia para a ferramenta final do projeto AIMS ser aplicar todos os elementos da metodologia AIM juntamente com Scrum, para o componente em que o trabalho desta dissertação se foca, apenas interessa mencionar a implementação das práticas de CMMI através do TSP.

Enquanto que os cinco componentes principais da metodologia AIM devem trabalhar juntos e de forma equilibrada para qualquer organização particular, é justo dizer que o TSP é a chave para uma implementação eficaz da AIM. Por si só, práticas TSP mapeiam uma grande parte das práticas específicas de CMMI. Além disso, existem cinco características essenciais do TSP que, mantendo a consistência com as práticas CMMI, vão significativamente além dos requisitos do modelo. O efeito final destas práticas é uma entrega de funcionalidades livres de defeitos e dentro do prazo e orçamento estipulados. As características do TSP resumem-se em [MCM10] [CKS11]:

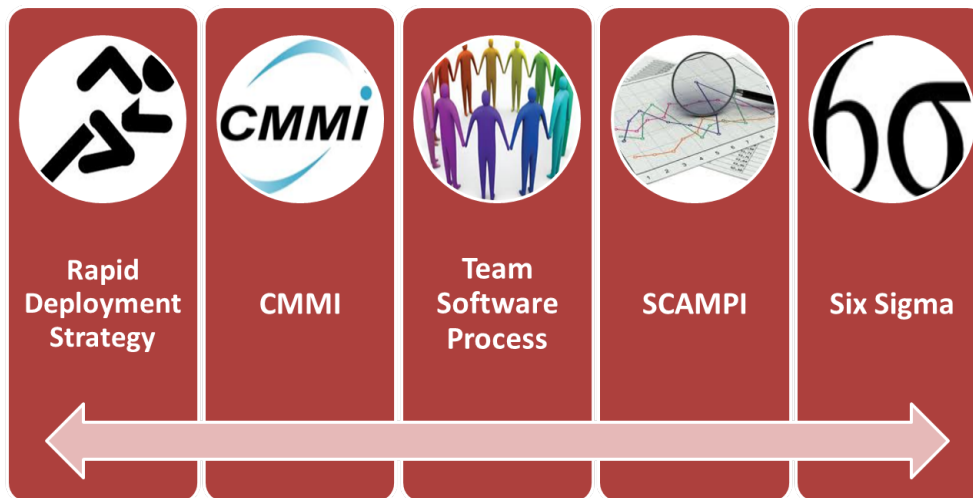


Figura 2.1: Componentes da metodologia AIM [CM12]

1. formação individual (PSP);
2. definição de métricas do TSP;
3. formação para equipas (TSP);
4. práticas de qualidade num ciclo de vida compreensivo;
5. estratégia de melhoria para uma equipa de projeto focada.

Em termos de produtividade, AIM combina TSP e adapta avaliações SCAMPI com elementos *Six Sigma* e outras técnicas para alcançar 30% de ganhos na produtividade de um projeto enquanto que reduz a quantidade de defeitos em 80% e mantém práticas com um nível de maturidade 2 e 3, de acordo com o CMMI [MCM10].

2.2 CMMI

O CMMI (*Capability Maturity Model Integration*) é provavelmente o conjunto de modelos de melhoria para o desenvolvimento de software mais amplamente utilizado. Como um modelo de melhores práticas, o CMMI descreve as características de tais práticas e, apesar de numerosos exemplos de técnicas de implementação, pelo próprio *design* do modelo, evita quaisquer recomendações de como se deve projetar e implementar tais práticas. A intenção do AIM é oferecer apenas um conjunto de tais recomendações, apesar de não definir um conjunto específico.

A versão 1.3 do CMMI foi publicada em 27 de outubro de 2010 e apresenta três modelos:

- CMMI-DEV (*CMMI for Development*), voltado ao processo de desenvolvimento de produtos e serviços.
- CMMI-ACQ (*CMMI for Acquisition*), voltado aos processos de aquisição e terceirização de bens e serviços.

Análise de Metodologias

- CMMI-SVC (*CMMI for Services*), voltado aos processos de empresas prestadoras de serviços.

O CMMI foi construído considerando três dimensões principais: pessoas, ferramentas e procedimentos, e foi baseado nas melhores práticas para desenvolvimento e manutenção de produtos. O processo serve para unir essas dimensões e inclui quatro áreas principais da Engenharia, sendo elas [MCNM07]:

1. Engenharia de sistemas
2. Engenharia de software
3. Desenvolvimento integrado de produtos e processos (IPPD)
4. Fornecimento de produtos auxiliares

A engenharia de software é similar à engenharia de sistemas em relação às áreas de processo, apenas com ênfase diferente nos processos e no seu nível de maturidade.

O CMMI possui duas representações: contínua ou por estágios. Estas representações permitem à organização utilizar diferentes caminhos para a melhoria de acordo com o seu interesse. Numa representação contínua, a organização pode utilizar a ordem de melhoria que melhor pretender para os objetivos de negócio da empresa. Numa representação por estágios são mostradas todas as áreas de processo na forma de *roadmap*, permitindo às organizações focarem-se em melhorias básicas antes de se focarem em objetivos avançados. Esta última representação é caracterizada pelos seguintes níveis de maturidade [MCNM07] [MCM10]:

- Nível 1: Inicial
- Nível 2: Gerido
- Nível 3: Definido
- Nível 4: Gerido Quantitativamente
- Nível 5: Otimizado

Depois de tudo isto, o que se pretende para esta fase de desenvolvimento do componente de gestão de projetos, é o CMMI-DEV nível 2 (gerido). Este modelo foca-se nas seguintes áreas de processo:

- Gestão de Requisitos (REQM), na categoria de Gestão de Projetos;
- Planeamento de Projeto (PP), na categoria de Gestão de Projetos;
- Monitorização e Controlo do Projeto (PMC), na categoria de Gestão de Projetos;
- Gestão de Acordo com o Fornecedor (SAM), na categoria de Gestão de Projetos;

Análise de Metodologias

- Medição e Análise (MA), na categoria de Suporte;
- Garantia da Qualidade do Processo e do Produto (PPQA), na categoria de Suporte;
- Gestão de Configurações (CM), na categoria de Suporte.

Cada área de processo possui um conjunto de práticas associadas, as quais se encontram referidas no Anexo B e no Anexo C juntamente com a integração dos métodos que são apresentados nas secções seguintes.

Nos objetivos desta dissertação, apenas interessa focar na categoria de gestão de projetos, logo apenas serão referidas as áreas de processo de gestão de requisitos, planeamento de projeto, monitorização e controlo de projeto e gestão de acordo com o fornecedor.

Mais tarde poderá envolver mais áreas de processo para que assuma o nível de maturidade 3, nomeadamente desenvolvimento de requisitos, solução técnica, integração do produto, verificação, validação, enfoque no processo organizacional, definição do processo organizacional, formação organizacional, gestão integrada do projeto, gestão de risco, integração de equipas, gestão integrada de fornecedores, ambiente organizacional para integração, análise das decisões e resolução.

As melhores práticas do CMMI-DEV são flexíveis o suficiente para serem aplicadas a uma variedade de indústrias, mas estáveis e consistentes para fornecerem uma referência com a qual uma organização se pode medir e comparar a si própria. A adoção do CMMI-DEV é um investimento sólido e de alto retorno que uma organização pode fazer para garantir resultados a longo prazo e duradouros. Os benefícios de negócio em organizações que usam o CMMI-DEV nas suas estratégias de melhoria de processos incluem satisfação do cliente, aumento da qualidade, planeamentos mais precisos, menores custos de desenvolvimento, retorno substancial do investimento, aumento da moral dos funcionários e aumento do volume de negócios [CM12] [CKS11].

Resumidamente, as práticas do CMMI permitem [MCNM07]:

- estabelecer estimativas: definem o âmbito do projeto, estabelecem estimativas para o desenvolvimento do produto e as tarefas atribuídas, definem o ciclo de vida de um projeto e determinam estimativas de esforço e custo.
- desenvolver o plano do projeto: estabelecem o orçamento e os prazos, identificam riscos do projeto, elaboram o plano para a gestão de dados, elaboram o plano para os recursos do projeto, elaboram o plano para os conhecimentos e práticas necessárias, elaboram o plano para o envolvimento dos *stakeholders* e estabelecem o plano do projeto.
- validação do plano: revêm os planos que afetam o projeto, adaptam o nível de trabalho aos recursos existentes e estabelecem um acordo com o plano.

2.3 TSP

2.3.1 Breve introdução

O TSP (*Team Software Process*) é uma tecnologia complementar do SEI que permite às equipas de desenvolvimento de software uma maior produtividade e eficácia. Este processo mostra às equipas como produzir produtos de qualidade dentro de um custo estipulado e com um plano apertado [MCM10]. O TSP está implementado em diversas organizações, nas equipas de projeto e respetivos elementos [DM03]. Uma vez que o objetivo do TSP é melhorar o desempenho das equipas de projeto mudando o comportamento dos seus respetivos elementos, ele inicia-se com uma formação em PSP (*Personal Software Process*), que depois é levada para projetos de desenvolvimento, onde os elementos da equipa implementam uma *framework* orientada a tarefas de processos, gestão e medição que fornecem *feedback* de desempenho tanto para o indivíduo como para a equipa. Esta abordagem permite mudanças rápidas e profundas no comportamento, que se refletem numa melhoria de desempenho [Smi04].

PSP e TSP não foram pensados apenas para o desenvolvimento de software, mas para qualquer atividade intelectual. O TSP tem vindo a ser utilizado com sucesso há mais de 10 anos e é usado constantemente para verificar a implementação e identificar oportunidades para mais melhorias [MCM10].

Resumidamente, o processo do TSP é constituído pelo seguinte conjunto de elementos [MW05] [Mel09a] [Mel09b], que se encontram detalhados no Anexo A:

- **Scripts** como guias de processos específicos de trabalho;
- **Formulários, relatórios e gráficos** para recolha e apresentação de métricas e dados estatísticos gerados a partir do histórico do projeto e por algum dos *scripts*;
- **Papéis** especificados para guiar os elementos de um projeto ao longo das suas atividades, mas não de forma mecanizada, pois cada projeto é abordado de formas diferentes. O TSP possui um conjunto de papéis principais já pré-definidos, e descritos no Anexo A, nomeadamente: responsável pela interação com o cliente, responsável pelo *design*, responsável pela implementação, gestor de testes, gestor de planeamento, gestor de processos, gestor de qualidade, responsável pelo suporte e líder de equipa.

Estes são os elementos principais, para além destes também existem outras ferramentas, como *checklists*, diretrizes e especificações que não estão relacionadas com os papéis definidos; e também cursos e atividades autorizadas para formação das tecnologias do TSP e PSP.

2.3.2 Mapeamento com CMMI

Enquanto que o TSP se foca no projeto e na equipa, o CMMI foca-se na organização. O TSP fornece desempenho comprovado, eficiência na implementação operacional de processos e no suporte ao funcionamento das equipas. Enquanto que o CMMI fornece institucionalização organizacional, escalabilidade e reconhecimento no mercado [MCNM07].

O objetivo do TSP baseado em CMMI da metodologia AIM é fornecer às organizações um meio de compatibilidade com o CMMI utilizando os pontos fortes do TSP [MW05]. Ou seja, fornece às organizações os produtos e suporte necessários para alcançar os benefícios de desempenho da compatibilidade do TSP com o CMMI.

A Figura 2.2 permite identificar a percentagem de práticas específicas do CMMI relacionadas com o TSP, por cada área de processo no nível de maturidade 2. Para além disso, nas tabelas do Anexo B é possível visualizar que elementos do TSP estão relacionados com as práticas do CMMI.

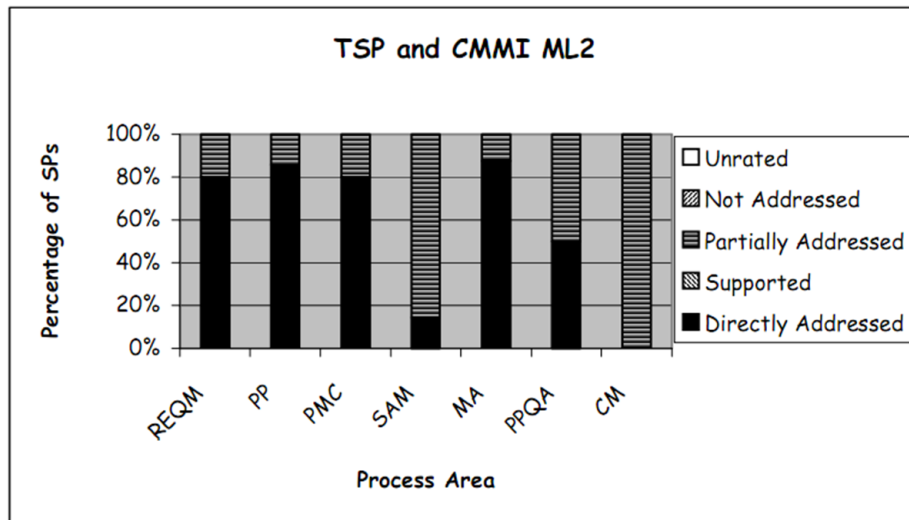


Figura 2.2: Impacto das práticas do TSP nas áreas de processo do CMMI com nível de maturidade 2 [MW05]

Quanto mais cedo o TSP for introduzido nas organizações melhor, não interessa em que nível de maturidade de CMMI se encontrem. Segundo McHale [MCNM07], é aconselhável a introdução do TSP antes do alcance do nível 3.

O SEI defende que são precisos, em média, 22 meses para evoluir um *projeto* do nível 2 para o nível 3 de maturidade, e 28 meses para evoluir um projeto do nível 3 para o nível 4 de maturidade [MW05]. Portanto, são precisos 50 meses para evoluir do nível 2 de maturidade para o nível 4. No entanto, a NAVAIR¹ veio mostrar que com a introdução do TSP isto é possível em apenas 16 meses [MW05]. Daí a vantagem de se construir uma ferramenta de gestão de projetos com integração com TSP.

No entanto o TSP não é mágico e também tem falhas. Normalmente falha pelas mesmas razões que os esforços para melhorar as práticas CMMI também falham: ou a equipa de gestão não percebe ou concorda com a necessidade de mudança, ou há falta de suporte de gestão, mudanças na gestão sénior, falhas de negócio ou cortes de orçamento [Ove10].

É muito mais fácil implementar o TSP numa equipa em que os elementos já estejam ambientados com o processo, porque poupa tempo e recursos na formação de PSP. No entanto, esta

¹A NAVAIR é uma empresa capaz de inovar e desenvolver novas tecnologias, sobretudo na área militar, naval e aeroespacial.

dificuldade existe sempre que se introduz um processo novo numa equipa, qualquer que seja o processo.

Outro aspeto crítico, é que qualquer esforço de mudança numa organização pode envolver bastante burocracia e exigir demais da organização, em vez de ajudá-la. Uma estratégia para facilitar isto é tratar as equipas TSP como clientes, juntamente com a integração de metodologias ágeis, como o Scrum, que irá ser referido mais à frente.

Para que o TSP ajude com sucesso na implementação das práticas CMMI, existe um conjunto de requisitos que têm de ser cumpridos: cada equipa TSP deve estar focada nos processos existentes e deve trabalhar perto das equipas de gestão de qualidade, processos, gestão da configuração, sistemas, requisitos e teste; todos os membros da equipa de desenvolvimento e os envolvidos na gestão do projeto devem estar bem formados por alguém autorizado pelo SEI, que depois deve acompanhar as equipas também no arranque das práticas do CMMI [MW05] [DM03].

No caso deste projeto, existem elementos da equipa de desenvolvimento autorizados pelo SEI para a certificação do TSP e níveis de maturidade do CMMI na empresa piloto Multicert.

2.4 Scrum

2.4.1 Breve introdução

Como já foi referido anteriormente, para além do TSP, também importa abordar as metodologias ágeis, mais precisamente o Scrum, como técnica para ajudar na melhoria de implementação do CMMI e do TSP.

O Scrum é um processo que se baseia num ciclo de desenvolvimento pré-definido, através da utilização de princípios ágeis. As metodologias ágeis promovem um processo de gestão de projetos que incentiva a inspeção e adaptação frequentes, bem como uma filosofia de liderança utilizando o trabalho em equipa e auto-organização [Sch07].

O Scrum é um processo que as equipas podem adotar rapidamente para planear e gerir o seu trabalho. Cada iteração do Scrum define-se por um conjunto de etapas que exige planeamento, conceção, implementação e teste do código, enquanto é analisado o progresso da equipa [Sch04].

A intenção do Scrum é orientar equipas para desenvolver projetos que se encontram prontos para ir para o mercado ao fim de pequenas iterações, em que cada iteração tem uma duração entre duas a quatro semanas. Normalmente, uma iteração de um projeto baseado em Scrum, também conhecido por *sprint*, contém os seguintes passos [Sch04] [Sch07]:

- Escrever os requisitos em formato de *user stories* e armazená-las no *Product Backlog*;
- Planear a entrega final, com base nas suas *user stories* e objetivos;
- Planear a iteração, que pode atingir entre 2 a 4 semanas cada;
- Controlar a iteração;
- Análise;

- *Design*;
- Codificação;
- Teste;
- Retrospectiva de cada iteração.

O Scrum, tal como o TSP, também contém um conjunto de elementos pré-definidos [Sch07] [SS11] [Sch07], tais como:

- **Eventos** também conhecidos por reuniões. O Scrum defende um conjunto de reuniões obrigatórias ao longo do projeto.
 - Reunião para planeamento da *sprint*: antes de qualquer iteração, o *Product Owner*, o *Scrum Master* e a equipa decidem no que se irá trabalhar durante a próxima iteração. O *Product Owner* mantém uma lista priorizada dos itens do *Product Backlog* e do que pode ser reavaliado durante o planeamento da iteração. A equipa seleciona apenas a quantidade de trabalho que sabe que pode terminar nessa iteração. A equipa então planeia a arquitetura e o *design* de como o produto pode ser implementado. Os itens do *Product Backlog* são então divididos em tarefas que se tornam o *Sprint Backlog*.
 - Reuniões *stand-up* diárias para acompanhar o progresso da equipa e identificar os obstáculos encontrados e possíveis soluções. Esta reunião tem um conjunto de regras específicas, nomeadamente, começar no horário marcado, ter uma duração máxima de 15 minutos e ocorrer no mesmo local e hora todos os dias. Para além disto, durante a reunião cada elemento da equipa responde a três perguntas (o que fiz desde ontem? o que planeio fazer hoje? estou com algum obstáculo que me impede de avançar?).
 - E no final de cada iteração, existem duas reuniões, uma para revisão do trabalho feito ao longo da *sprint* e outra para retrospectiva do processo e métodos de trabalho.
- **Papéis** em que normalmente são três os principais (*Product Owner*, *Scrum Master* e elemento da equipa) e são basicamente os que produzem o produto, mas também podem existir papéis auxiliares [Gy11].
 - O *Product Owner* comunica a visão do produto para a equipa de desenvolvimento, o que inclui representar os interesses do cliente, através de requisitos e prioridades. Basicamente funciona como *stakeholder*.
 - O *Scrum Master* atua como uma ligação entre o *Product Owner* e a equipa. Ele não só gere a equipa, como também trabalha para ajudar a equipa a alcançar as suas metas em cada iteração, removendo os obstáculos. O *Scrum Master* verifica se o processo Scrum está a ser bem usado.
 - Os elementos da equipa fazem o trabalho do projeto. Normalmente consiste numa equipa de engenheiros de software, arquitetos, analistas e *testers*.

- **Artefatos** são um conjunto de materiais produzidos e gerados ao longo do projeto. Tais como:
 - *Product Backlog*: é uma lista de itens priorizados, *user stories* que tipicamente vêm do cliente, a serem desenvolvidos para um software.
 - *Sprint Backlog*: é uma lista de itens selecionados do *Product Backlog* e contém tarefas concretas que serão realizadas durante a próxima iteração para implementar os itens selecionados. O *Sprint Backlog* é uma representação em tempo real do trabalho que a equipa de desenvolvimento planeia concluir na iteração corrente.

2.4.2 Mapeamento com CMMI

Scrum é um exemplo de implementação de algumas práticas referentes ao nível de maturidade 2. No Anexo C são listadas as principais práticas do CMMI que mapeiam corretamente para as etapas do processo Scrum. O que não significa que uma organização não possa, eventualmente, acrescentar as práticas do CMMI adicionais para os seus projetos.

Embora as práticas de Scrum forneçam exemplos de implementação de muitas boas práticas do CMMI Nível 2, um problema recorrente é o nível de artefatos e evidências necessárias para avaliar um projeto no nível 2. Se uma equipa perde os seus artefatos, não é possível avaliar o nível de maturidade de um projeto. Idealmente, os elementos da equipa Scrum devem armazenar o seu trabalho para que se possam referir a iterações passadas durante reuniões de retrospectiva [PS09].

Nas tabelas no Anexo C são mostradas várias práticas do CMMI e de que forma o processo Scrum pode implementar cada prática. Tal como o TSP, o Scrum é um processo que não influencia da mesma forma todas as práticas do CMMI no nível de maturidade 2. As áreas de processo mais influenciadas na categoria de Gestão de Projetos, são [Roy08] [Tur11]:

- **Gestão de Requisitos (REQM)**: em que o objetivo é gerir os requisitos referentes ao projeto e seus componentes e identificar inconsistências entre esses requisitos e o planeamento do projeto;
- **Planeamento do Projeto (PP)**: em que o objetivo é estabelecer e manter planos que definam as atividades do projeto;
- **Monitorização e Controlo do Projeto (PMC)**: em que o objetivo é fornecer uma compreensão da progressão do projeto para que as ações necessárias possam ser tomadas corretamente quando o desempenho do projeto se desvia significativamente do plano.

Ainda na categoria de Gestão de Projetos, não há práticas de Scrum que lidam com a seleção e gestão de fornecedores [PS09].

2.4.3 Mapeamento com TSP

Ao contrário do CMMI e do TSP, pelo menos no que se refere à versão oficial de TSP que é um produto licenciado, uma equipa de desenvolvimento não precisa de ser certificada para poder

usufruir das vantagens do Scrum. No entanto, isto não invalida que seja necessário uma formação para o uso deste processo.

A lista a seguir representada mostra quais as principais diferenças entre o Scrum e o TSP [Pad08] [DavPAa] [DavPAb]:

- Nos objetivos, o Scrum defende o produto e o TSP defende o negócio e o produto;
- Nos papéis definidos, enquanto que o Scrum possui três, o TSP possui nove, como já foi mostrado anteriormente;
- Na definição de processos, o TSP define explicitamente processos de engenharia que continuam a ser melhorados pelas equipas, enquanto que o Scrum não possui nenhum processo definido de base;
- No planeamento detalhado, o Scrum detalha tarefas ao nível da equipa, enquanto que o TSP detalha tarefas por cada elemento da equipa;
- Na gestão de riscos, o TSP coloca a equipa a definir e gerir os seus próprios riscos, enquanto que o Scrum não aborda este tema;
- No Scrum as iterações têm um tamanho fixo, enquanto que no TSP pode ser variável; mas ambos os processos defendem iterações curtas;
- As métricas valorizadas no Scrum são a velocidade e o tempo estimado que resta para completar o projeto, já no TSP olha-se ao trabalho já desenvolvido, ao número de horas em tarefas, à existência de defeitos e ao tamanho do produto;
- Em termos práticos, o Scrum recorre a reuniões diárias, enquanto que o TSP aborda reuniões semanais e *checkpoints*. Ambos recorrem a reuniões no início e fim das iterações para planeamento e retrospectiva do projeto.

Apesar destas diferenças, é possível integrar os conceitos de ambos os processos, como indica a Figura 2.3, de forma a conseguir um único processo mais eficaz e produtivo com conceitos únicos, tal como se irá provar mais à frente.

Também é possível coincidir o ciclo do TSP com as iterações do Scrum de uma forma simples, tal como indica a Figura 2.4.

O planeamento de ambos os métodos, também é algo que, embora distinto, é possível associar. Na Figura 2.5 é possível concluir que, em termos de planeamento, o TSP é mais detalhado que o Scrum, o que nem sempre é útil para as empresas. Logo é importante definir e manter um meio termo minimamente flexível para qualquer empresa.

Neste momento, está provado teoricamente que é possível integrar TSP, uma metodologia que utiliza um *workbook* para gestão de projetos com bastante detalhe e carregado de informação para análises futuras, juntamente com o Scrum, uma metodologia ágil que defende o mínimo de documentação possível e um processo mais simples possível. Nos próximos capítulos irá ser

Análise de Metodologias

Scrum	TSP
User Stories	Work Items, Tasks
Release Planning	Launch
Release	Assembly
Sprint Planning	Re-launch
Sprint	Cycle
Project	Project
Sprint Demo	
Daily Standup	Weekly Status Meeting
	Checkpoints
	Earned Value Chart
	Resource Burnup Chart
Burndown Chart	

Figura 2.3: Mapeamento dos conceitos de TSP com os conceitos de Scrum [Roy08] [DavPAa]

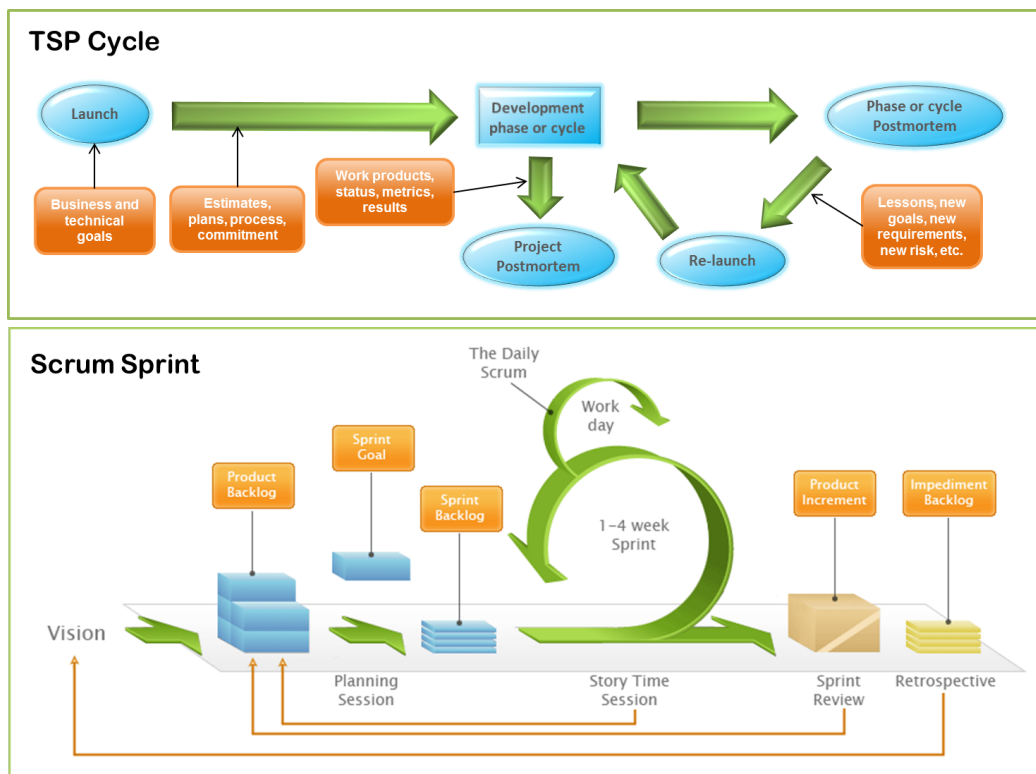


Figura 2.4: Mapeamento dos ciclos do TSP com as iterações do Scrum [DavPAa]

mostrada a forma como é possível criar na prática um *plugin* que implemente estes dois métodos e até que ponto é possível a sua integração.

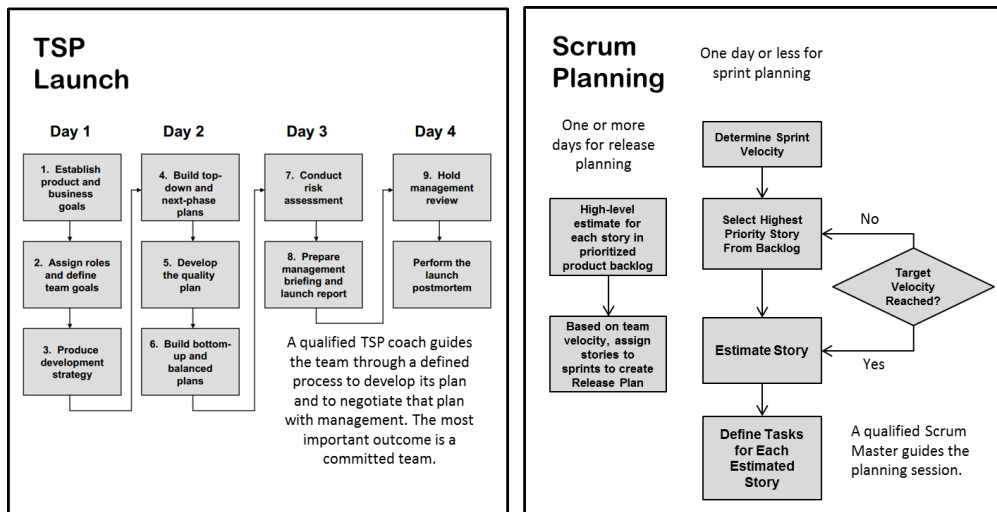


Figura 2.5: Mapeamento do planejamento de TSP com o plano a ser seguido no Scrum [DavPAa]

Capítulo 3

Análise e Seleção de Ferramentas

Este capítulo começa por apresentar um conjunto de ferramentas candidatas ao projeto AIMS e depois é feita uma breve análise das mesmas para o componente de gestão de projetos a ser desenvolvido.

3.1 Requisitos

A Tabela 3.1 representa a lista de requisitos selecionados e considerados relevantes para a elaboração do trabalho desta dissertação. Estes requisitos foram definidos para o projeto AIMS pelo seu Consórcio e serviram como base para a definição de uma lista prévia de *user stories*, bem como para o desenvolvimento de um conjunto de critérios base para avaliação e comparação das ferramentas de gestão de projetos selecionadas na secção seguinte.

Na Tabela 3.1, a numeração dos RID's não aparece com uma ordem sequencial pois, como já foi referido anteriormente, faltam alguns requisitos da ferramenta que não são relevantes para o componente de gestão de projetos e como tal não foram apresentados.

Esta é uma lista de requisitos de alto-nível elaborada numa fase inicial do projeto que mais tarde deu origem a um conjunto de *user stories*, apresentadas no Capítulo 4, que providenciam mais detalhe do desenvolvimento do projeto AIMS.

3.2 Ferramentas candidatas

Durante o estudo do estado da arte foram vistas cerca de 120 ferramentas disponíveis no mercado. Deste conjunto de ferramentas apenas quatro foram selecionadas para análise, juntamente com a hipótese da ferramenta ser criada de raiz. A Figura 3.1 representa o conjunto de ferramentas que sobressaiu devido à sua popularidade, funcionalidades existentes e por recomendação do Consórcio AIMS.

Para além da utilização de uma ferramenta como base, também foi pensado outro cenário, o desenvolvimento de uma ferramenta de raiz. Esta solução seria uma das melhores opções devido à falta de dependências e restrições em termos de tecnologias, base de dados e arquitetura usadas.

Análise e Seleção de Ferramentas

RID	Descrição	Prioridade
R01	Suporte para as práticas de TSP em gestão de projetos	Alta
R02	Suporte para as práticas de TSP em gestão de qualidade	Alta
R03	Suporte para as práticas de TSP em organização de equipas	Alta
R04	Formulários, gráficos e relatórios de TSP	Média
R05	Controlo do tempo, defeitos, tamanho e calendarização	Média
R06	Gestão de riscos e tarefas	Média
R07	Providenciar assistentes de planeamento (gerar lista de tarefas e calendarização)	Média
R08	Providenciar vistas individuais, da equipa e da organização	Média
R09	Otimizar a vista da equipa para suportar as reuniões do projeto (lançamento, post-mortem, reunião semanal, ...)	Média
R10	Otimizar vistas individuais para o trabalho diário e <i>roles</i>	Média
R11	Otimizar vista da organização para relatórios sobre o estado do projeto	Média
R12	Suporte para as práticas de Scrum em gestão de projetos (consistente com TSP)	Média
R13	Suporte para as práticas de Scrum na gestão de equipas (consistente com TSP)	Média
R14	Suporte para as práticas de Scrum na gestão de requisitos	Média
R15	Suporte para gráficos “TSP Earned Value”	Alta
R16	Suporte para gestão de custos	Baixa
R17	Suporte para gestão de reuniões	Baixa
R18	Fornecer dados históricos para planeamento	Média
R19	Medição automatizada do tamanho	Média
R20	Definição de processos (<i>scripts, roles, checklists, templates, KPIs, etc</i>)	Alta
R21	Controlo de versões dos processos	Média
R26	Adaptação do processo (<i>tailoring</i>)	Média
R27	Gráficos de controlo	Baixa
R28	Gráficos de análise	Média

Tabela 3.1: Lista de requisitos de alto-nível

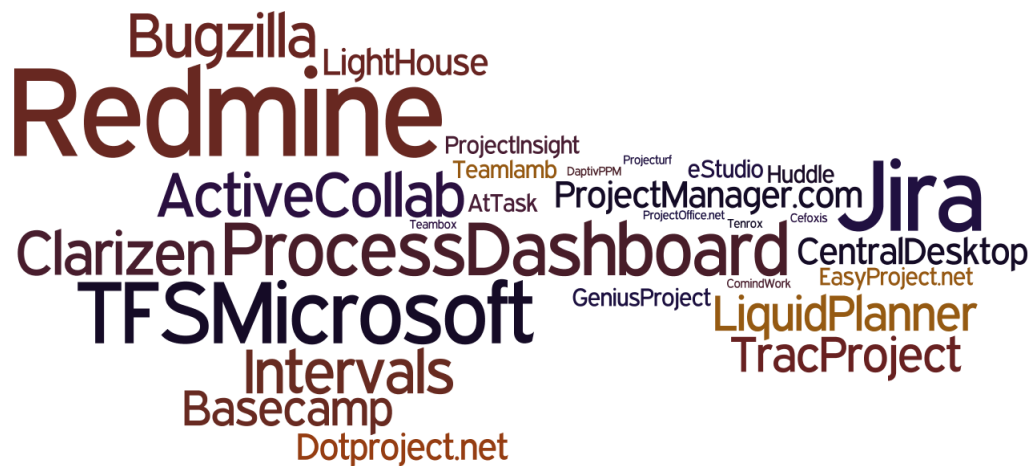


Figura 3.1: Representação das ferramentas selecionadas para análise de acordo com a importância relativa para o projeto no formato de uma *Word Cloud*, usando o programa *Wordle*

No entanto, por falta de tempo e recursos não foi possível nesta primeira fase. Para além disso, o uso de uma ferramenta de base permite reutilizar componentes já desenvolvidos, melhorá-los e adaptá-los de acordo com as necessidades do projeto. Como tal, optou-se por criar um *plugin* que integrasse TSP com Scrum para uma qualquer ferramenta de gestão de projetos, e que fosse o mais independente possível. Assim, numa segunda fase deste projeto, é possível substituir essa ferramenta de gestão de projetos por uma desenvolvida à medida deste projeto, se na altura se verificar necessário.

Segue-se uma análise das principais ferramentas candidatas e o porquê de terem sido selecionadas.

3.2.1 Jira

O Jira é uma ferramenta de gestão de projetos para equipas que desenvolvem projetos de software. É um produto da Atlassian¹ que funciona como o núcleo de uma equipa de desenvolvimento e mantém a ligação da equipa ao trabalho que está a ser desenvolvido. Também permite gerir erros e defeitos, relacionar os problemas encontrados ao respetivo código fonte, planear um desenvolvimento ágil, monitorizar as atividades, reportar o estado do projeto, entre outras funcionalidades [Jir12b].

É uma ferramenta com um enorme conjunto de funcionalidades no que diz respeito à gestão dos acessos e notificações, *gadgets*, gestão de tarefas e projetos, criação de relatórios e gráficos e integra com mais de cem *plugins* [Jir12a]. A linguagem usada para desenvolvimento desta ferramenta é Java.

¹Atlassian é uma empresa de desenvolvimento de *Software as a Service* de renome. Atualmente tem mais de 20,000 empresas clientes e um dos seus produtos é o Jira.

Esta ferramenta não integra com processos do TSP, mas integra *plugins* que permitem adotar metodologias ágeis e outras funcionalidades extra, nomeadamente o produto GreenHopper, no entanto trazem um custo acrescido à licença já paga pelo Jira [Gre12].

Foi escolhida para uma análise mais detalhada por ser uma ferramenta muito popular usada por grandes empresas como Facebook, IBM, ebay, LinkedIn, Cisco, entre outras, e sobretudo porque é a ferramenta usada atualmente pela empresa piloto do Consórcio AIMS, a Multicert [Jir12b].

3.2.2 Visual Studio Team Foundation Server 2010

O Microsoft Visual Studio Team Foundation Server 2010 (TFS) é uma plataforma de colaboração da Microsoft como solução para a gestão do ciclo de vida de um projeto. O TFS automatiza o processo da entrega de software e fornece as ferramentas necessárias para gerir projetos de desenvolvimento de software, de forma eficaz, em todo o seu ciclo de vida [Tfs12].

Esta ferramenta possui um conjunto vasto de funcionalidades no que diz respeito ao controlo de versões, gestão de *work items*, criação de relatórios e gráficos com base no histórico dos dados que grava numa *data warehouse*. Na área de gestão de projetos, também possui uma integração com as práticas de CMMI e Scrum [Tfs12].

Em comparação com o Jira, esta ferramenta está mais adaptada para a integração com TSP, sobretudo na parte dos relatórios e gráficos, e já suporta Scrum. No entanto, a integração desta ferramenta com um novo *plugin* e mesmo na empresa piloto seria mais complexa do que com o Jira, pois iria exigir mais recursos na formação para a ferramenta e mais atraso na produtividade da empresa durante a fase de adaptação.

Foi escolhida para análise por ser uma ferramenta da Microsoft, uma empresa que se encontra bem posicionada no mercado, e por ser a ferramenta que melhor se aproxima dos objetivos do AIMS. Também surgiu como ferramenta candidata por sugestão do Consórcio AIMS e por aparentar ser bastante robusta e com muitas funcionalidades pretendidas, o que se veio a comprovar na análise feita posteriormente.

3.2.3 Process Dashboard

O Software Process Dashboard Project é um projeto *opensource* para criar uma ferramenta que suporta PSP/TSP. Tal como o Consórcio AIMS, este projeto também defende que é importante criar uma ferramenta com suporte para PSP/TSP, de forma a automatizar um processo que requer um levantamento e análise de métricas num nível muito detalhado. E uma vez recolhidas estas métricas, é necessário proceder a análises estatísticas dos dados gerados no planeamento, ao longo do projeto e nas estimativas sobre o desenvolvimento dos produtos de software [Pro12a] [Pro12b].

É importante referir que apesar de se identificar como uma ferramenta *opensource*, apenas o é na integração com PSP. Quando se trata da implementação do processo TSP, na versão oficial, já é necessário pagar um custo de licença para poder usufruir deste processo na ferramenta.

O Process Dashboard não foi selecionado para que pudesse servir como ferramenta base para o projeto AIMS por ser uma ferramenta mais fraca em termos de usabilidade e interação com o

utilizador. No entanto, é importante referi-la como concorrente no mercado por ser das poucas ferramentas que integra o processo TSP.

3.2.4 Redmine

O Redmine é uma ferramenta *opensource* de gestão de projetos bastante flexível, desenvolvida na tecnologia de Ruby on Rails. Esta ferramenta possui um conjunto de funcionalidades, nomeadamente, suporte para múltiplos projetos, acesso aos dados por papéis, gestão de tarefas flexível com suporte para calendarização, mapa de Gantt, documentos e ficheiros, com análise do controlo de versões dos ficheiros, com suporte para rastreamento do tempo. Também suporta múltiplas bases de dados, múltiplos repositórios e criação de *custom fields* [Red12a].

Esta ferramenta foi selecionada para uma análise mais detalhada, precisamente por ser uma ferramenta bastante flexível em comparação com o Jira e o TFS, e com uma melhor interação com o utilizador em comparação com o Process Dashboard. Isto torna o Redmine a ferramenta ideal na seleção para uma ferramenta base, pois permite à equipa de desenvolvimento adaptar a ferramenta de qualquer forma. Ao mesmo tempo que poupa tempo e recursos nesta fase inicial do projeto, pois já possui todas as funcionalidades para gestão de projetos implementadas.

3.2.5 Comparação entre as ferramentas e estratégias candidatas

Inicialmente foi feita uma avaliação, Figura 3.2, em termos de suporte de funcionalidades específicas sobre as seguintes ferramentas: Jira, TFS, Clarizen e Redmine. O Clarizen foi uma ferramenta incluída para um estudo prévio por ser novidade e ter tido algumas avaliações positivas em blogs e revistas online, por exemplo a "TopTenREVIEWS"².

Em simultâneo, foi feito um pequeno estudo apenas sobre algumas ferramentas *opensource*, Figura 3.3, nomeadamente: Scarab, Process Dashboard, Redmine, Collabtive, Embeta Forge e Plandora Project. Este estudo foi importante uma vez que se tratam de ferramentas *opensource* e a licença é um fator relevante para que o Consórcio AIMS possa avançar com o projeto.

Após uma breve descrição de cada ferramenta e uma análise aprofundada das funcionalidades de algumas delas, é possível visualizar as suas principais diferenças através da seguinte tabela comparativa (Figura 3.4), cujos critérios de comparação foram definidos segundo os requisitos apresentados na secção 3.2.

Os critérios identificados como os mais relevantes para o Consórcio AIMS foram os seguintes, apresentados segundo a respetiva ordem de prioridade (refletida em pesos relativos na tabela da Figura 3.4):

1. Fácil desenvolvimento da solução, o que inclui o esforço da equipa a desenvolver a aplicação, o risco da ferramenta ter de ser abandonada a meio e não haver aproveitamento do trabalho feito e flexibilidade da ferramenta para incluir *plugins* e ser adaptada às necessidades do projeto;

²TopTenREVIEWS é uma revista online que todos os anos lança uma comparação das melhores ferramentas online de gestão de projetos (online-project-management-review.toptenreviews.com)

Análise e Seleção de Ferramentas

Name	Jira	Visual Studio Team Foundation Server 2010	Clarizen	Redmine
Company	Atlassian	Microsoft	Clarizen	-
Cost/Licence	\$10 p/month for 10 users or \$1200 full version for 11-25 users	\$499.99 full version for 5 users	\$24.95/user/month or \$44.95 /user/month (unlimited version)	Opensource GPL
Support TSP	✗	✗	✗	✗
Reporting tools (diagrams, charts)	+	+	+	+
Mapping to CMMI	+	+	+	✗
Collaborative document management	✗	+	+	+
Test management	✗	+	✗	✗
Risk and issue management	+	+	+	+
Central repository	+	+	+	+
Version control	✗	+	?	+
Business intelligence (OLAP)	?	+	?	+
Support Agile (Scrum)	✗	+	✗	✗
Workflow management	+	+	+	+
Multiple projects and resources management	+	+	+	+
Integration with other management tools	+	✗	✗	+
IDE integration	✗	+	?	✗
Outlook integration	✗	+	+	✗
FAQ's and knowledge base	+	+	+	+
Export data	+	+	+	+
Data access by roles and users authorized	+	+	+	+
Keep record	+	+	+	+
Audit logs for each data access	+	+	?	+
Filter reports	+	+	+	+
Automatic notification for deadlines	+	+	+	+
Interface customization	+	✗	✗	✗
Manage dashboard	+	+	+	+
Be searchable	+	+	+	+

Figura 3.2: Tabela de comparação entre as ferramentas: JIRA [Jir12a], Visual Studio Team Foundation Server 2010 [Tfs12], Clarizen [Cla12] e Redmine [Red12a]

2. Aceitação pelo piloto, o que inclui o custo que terá para a empresa piloto no caso de haver licenças e a formação para a adaptação da ferramenta no caso desta ser completamente desconhecida para a empresa;
3. Independência, o que significa a quantidade de entidades, licenças e tecnologias sobre as quais o projeto irá depender. Quantas menos, melhor;

Análise e Seleção de Ferramentas

	Scarab	Process Dashboard	Redmine	Collabtive	Embeta Forge	Plandora Project
Link	http://www.solitone.org/s-carab-trunk-doc/databases/index.html	http://www.processdash.com/functionality	http://www.redmine.org/	http://www.collabtive.com/	http://www.emforge.org/home	http://www.plandora.org/project.htm
Technologies	Java servlet, Eclipse, NetBeans, Subclipse, XML, Maven, TomCat, PostgreSQL, Hypersonic, Oracle, MSSQL, Turbine, Torque, Intake, Velocity, Lucene	Java	Ruby onRails, Javasript, CSS, XML, JSON	LAMP servers (Linux, Apache, MySQL, PHP)	Java, Maven, Tomcat, Eclipse, Subversion Client	Java, Tomcat, Eclipse, MySQL, JBDC, JSP, JUDE
Documentation	http://www.solitone.org/s-carab-trunk-doc/guides.html	For TSP: http://www.processdash.com/static/teamhelp/frame.html? Only with PSP: http://www.processdash.com/static/help/frame.html?	http://www.redmine.org/projects/redmine/wiki/Guide	http://code.google.com/p/collabtiv/source/browse/	http://www.emforge.org/wiki/-/wiki/Main/EmForgeSetupGuide	http://www.plandora.org/docs/PlandoraVisao.pdf
Observations	Integration with Jira and Bugzilla using templates.	Has TSP, but this part of the tool is not opensource.	Redmine plugin hooks: allow external code to extend the core Redmine functionality in a clean way		Development workflows	Use cases diagram
Source Code	http://scarab.tigris.org/source/browse/scarab/	http://www.processdash.com/download_mec/pdashSrc	http://www.redmine.org/projects/redmine/wiki/Download	http://code.google.com/p/collabtiv/downloads/list	http://www.emforge.org/wiki/-/wiki/Main/Download	http://sourceforge.net/projects/plandora/files/Version/plandora-v.1.11.0/
Database Diagrams	http://www.solitone.org/s-carab-trunk-doc/databases/schema.html					http://www.plandora.org/docs/mer.png
API	http://www.solitone.org/s-carab-trunk-doc/apidocs/index.html		API with Java API Redmine for Developers	http://code.google.com/p/collabtiv/source/browse/		

Figura 3.3: Tabela informativa sobre Scarab, Process Dashboard, Redmine, Collabtive, Embeta Forge, Plandora Project

4. Nível de integração entre os vários componentes que devem constituir a solução AIMS e a qualidade da solução em termos de usabilidade e quantidade de informação;
5. Custo e licenças que o utilizador final terá de comprar para poder usufruir da ferramenta. Tal como as dependências, quanto menos melhor;
6. Aceitação no mercado, o que sendo uma ferramenta conhecida e utilizada no mercado será sempre uma mais valia;
7. Incorporação já feita das práticas do CMMI, TSP e Scrum. Embora, neste caso, o Scrum seja menos relevante por ser mais fácil de implementar e incluir no projeto do que o TSP;
8. Gestão colaborativa de documentos, o que para a empresa piloto é muito relevante na gestão dos seus projetos;
9. E por último, é avaliada a integração das ferramentas com a gestão de testes, requisitos, configurações e com OLAP ou *Business Intelligence*;

Como já foi referido na secção anterior, o Process Dashboard não foi considerado para a comparação. No entanto, foi considerada uma ferramenta de apoio à implementação do TSP.

Análise e Seleção de Ferramentas

Categoria	Critério de Seleção	P (%)	Jira	Core	TFS	Redmine
Suporte de base	Práticas CMMI	5	*** ¹ 3,75	** 2,5	*** ² 3,75	** 2,5
	TSP	5	** 2,5	* 1,25	** 2,5	** ¹⁴ 2,5
	Gestão colaborativa de documentos e conteúdos	4	*** ³ 3	*** 3	*** ⁴ 3	*** ¹⁵ 3
	Scrum	2	*** ⁵ 1,5	* 0,5	*** ⁶ 1,5	* 0,5
	Gestão de testes, configurações, requisitos, ...	2	** 1	** 1	*** 1,5	** ¹⁶ 1
	OLAP/BI	2	** 1	** 1	*** 1,5	** 1
Desenvolvimento	Fácil desenvolvimento da solução (esforço, risco e flexibilidade)	18	*** 15	** 10	*** 15	*** ¹⁷ 15
Resultado	Nível de integração e qualidade da solução	12	** ⁷ 6	*** 9	**** ⁸ 12	** ¹⁸ 7
	Aceitação pelo piloto (custo, licenças, formação, ...)	16	*** 12	** 8	* ⁹ 4	** 8
	Independência	12	* ¹⁰ 3	**** 12	* ¹¹ 3	*** 9
Comercialização	Aceitação no mercado	10	*** 7,5	* 2,5	*** 7,5	** 5
	Custo e Licença para os utilizadores finais	12	* ¹² 3	**** 12	* ¹³ 3	**** ¹⁹ 12
TOTAL (%)		100	59,25	62,75	58,25	66,5

Legenda	
*	Inexistente ou com faltas de funcionalidades
**	Pouco desenvolvida
***	Bem desenvolvida, mas precisa de melhoramentos
****	Implementada completamente

^[1] Jira+Confluence são boa base para implementar solução de suporte CMMI níveis 2 e 3 de maturidade (<http://www.valiantys.com/>)

^[2] Usa templates para processos (<http://templex.codeplex.com/>), por exemplo para Kanban, Scrum, CMMI process improvement, etc.

^[3] É possível através do Confluence (plugin para Jira).

^[4] É possível através do SharePoint.

^[5] É possível através do GreenHopper (plugin para Jira).

^[6] Usa templates para processos (<http://templex.codeplex.com/>),

por exemplo para Kanban, Scrum, CMMI process improvement, etc.

^[7] O Jira possui mais de 100 plugins (Crucible, Confluence, CVS, Subversion, FishEye, GreenHopper, Bamboo),

no entanto não integra totalmente com o ambiente do Jira.

^[8] O TFS integra sem sair do seu ambiente.

^[9] Foi feita uma suposição em relação à empresa piloto.

^[10] Está depende de alguns produtos da empresa (Atlassian ou Microsoft) para um melhor desempenho.

^[11] Está depende de alguns produtos da empresa (Atlassian ou Microsoft) para um melhor desempenho.

^[12] \$10 para 10 utilizadores (versão completa) por cada plugin

^[13] \$499.99 para 5 utilizadores (versão completa)

^[14] É flexível o suficiente para isso.

^[15] Tem um módulo para criar templates de documentos e também tem uma wiki que permite editar de forma colaborativa.

^[16] Permite gerir requisitos como trackers de issues e tem uma boa gestão de configuração.

^[17] O nível de esforço é relativo pela tecnologia que usa e os conhecimentos da equipa de desenvolvimento.

^[18] Integra com bastantes plugins cujo nível de qualidade varia bastante.

^[19] A ferramenta base não tem custo e não pode ser comercializada,

mas o plugin desenvolvido terá o custo que o Consórcio AIMS entender.

Figura 3.4: Tabela de comparação entre o JIRA, a opção de desenvolver uma ferramenta de raiz, o Visual Studio Team Foundation Server 2010 e o Redmine [[Jir12a](#)] [[Tfs12](#)] [[Red12b](#)]

A coluna “P(%)” da tabela na Figura 3.4, representa o peso distribuído de cada critério de seleção cuja soma dá 100%. À frente das estrelas, em cada linha, é apresentado o peso da ferramenta em cada critério, que é calculado a partir do peso do critério a dividir por quatro (número máximo de estrelas) vezes o número de estrelas atribuídas. Ou seja, o cálculo de cada critério em cada ferramenta é feito da seguinte forma: $P(\%) / 4 * \text{número_de_estrelas_atribuídas}$.

3.3 Solução adotada

Inicialmente, a ferramenta escolhida foi o Jira por ser muito mais simples de integrar na empresa Multicert e em muitas outras grandes empresas que já utilizam esta ferramenta. No entanto, devido ao custo de licenças e falta de flexibilidade, esta decisão foi repensada.

Após um estudo da arte aprofundado sobre os processos e práticas a implementar, juntamente com uma análise comparativa entre três ferramentas de gestão de projetos, e após várias reuniões com o Consórcio AIMS foi possível concluir que existiam dois cenários principais possíveis para a implementação de uma nova ferramenta de gestão de projetos. Ou desenvolver a ferramenta sobre outra já existente que fosse *opensource*, e para tal a ferramenta mais indicada como base seria o Redmine, ou desenvolver a ferramenta de raiz.

O ideal seria desenvolver a ferramenta de raiz para que não houvesse demasiadas dependências de integração com outras ferramentas logo desde o início de vida do projeto, até porque em cada integração existe um custo de licença, isto no caso das ferramentas usadas não serem *opensource*. Mas, visto que os recursos e o tempo disponíveis para o projeto AIMS são reduzidos, a opção mais viável foi desenvolver a ferramenta utilizando o Redmine como ferramenta base, uma vez que se trata de uma ferramenta *opensource* bastante flexível e que se destaca nos critérios escolhidos em conjunto com o Consórcio AIMS.

Concluindo, a ferramenta base adotada foi o Redmine e a implementação da integração do TSP e Scrum será feita através da implementação de um *plugin, plugin* AIMS, que já se encontra a ser desenvolvido para uma primeira fase do projeto.

Análise e Seleção de Ferramentas

Capítulo 4

Conceção e Implementação

Este capítulo apresenta uma lista de *user stories*, juntamente com o modelo de domínio criado para o desenvolvimento do projeto, o cenário de arquitetura e respetivas tecnologias que se encontram a ser usadas no projeto. Depois da definição da solução, é explicado mais em pormenor de que forma está pensada a implementação prática desta solução idealizada para a integração de TSP com Scrum.

4.1 Mapeamento entre conceitos de TSP, Scrum, Redmine e AIMS

Antes de abordar o modelo de domínio, é importante ver de que forma se pode associar os conceitos de TSP e Scrum, Figura 2.3, aos do Redmine e obter um conjunto de conceitos a ser utilizado pelo *plugin* AIMS que se encontra a ser desenvolvido.

A Figura 4.1 mostra o conjunto de conceitos adotados para este *plugin*.

Scrum	TSP	Redmine	Plugin AIMS
User Stories Tasks	Tasks Defects	Issues	User Stories Tasks Bugs
Release	-	Release	Release
-	Assembly	-	Work Product
Project	Project	Project	Project
Sprint	Cycle	-	Iteration
Burndown Chart	-	-	Burndown Chart
-	Earned Value Chart	-	Earned Value Chart
-	Resource Burnup	-	Resource Burnup

Figura 4.1: Comparação de conceitos entre TSP, Scrum e Redmine e apresentação dos conceitos do *plugin* AIMS

4.2 Modelo de domínio

A Figura 4.2 representa um modelo de domínio que contém a relação entre os vários conceitos essenciais para o desenvolvimento do componente de gestão de projetos com integração com a gestão de processos.

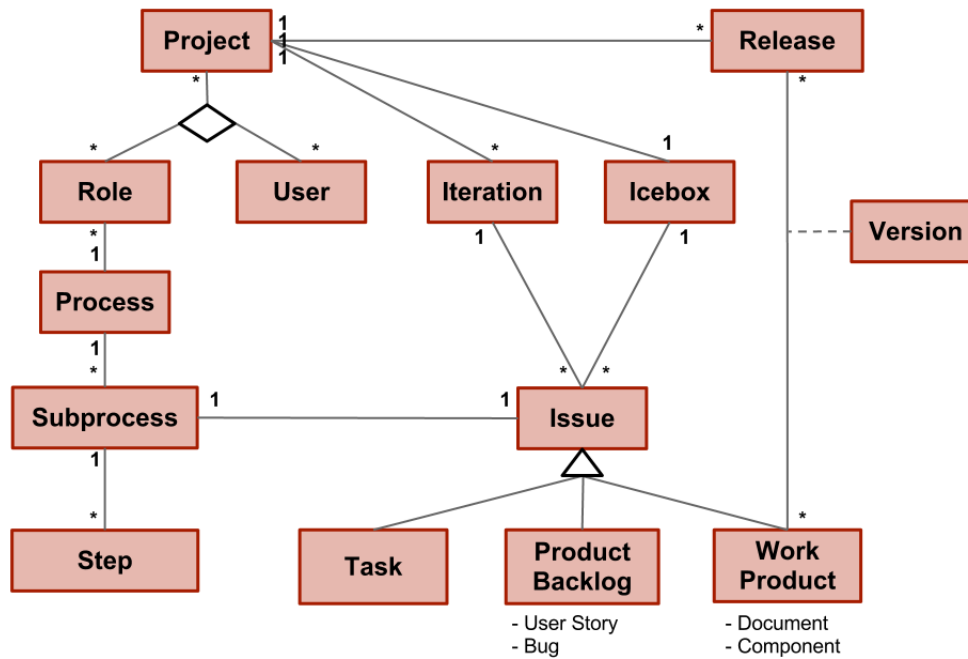


Figura 4.2: Modelo de domínio dos componentes de gestão de projetos e processos

4.3 User Stories

Após uma noção dos conceitos gerais e da forma como se relacionam, foi criada uma lista de *user stories*, Figura 4.3, essenciais para o projeto e com mapeamento para a lista de requisitos. Nesta listagem é possível verificar a prioridade de cada *user story*, a que requisitos é que corresponde e se já se encontra ou não implementada de raiz no Redmine. As *user stories* encontram-se organizadas por módulos.

Conceção e Implementação

USID	Description	Priority	Redmine	RID
OVERVIEW + TEAM				
US01	As an admin, I can create a company or an organization that will not be public for members not assigned.	High	Yes	R06, R03
US02	As an user, I can only view the projects that are public inside the company.	Medium	Yes	R08
US03	As a team member, I can only view my projects or the projects that are public inside the company.	Medium	Yes	R08
US04	As a project manager, I can create a new project (id, name, description, start_date, is_public, planned_end_date).	High	Yes	R01, R12
US05	As a project manager, I can edit a project.	Medium	Yes	R08
US06	As a team member, I can view a schedule performance of a project.	High	No	R11
US07	As a team member, I can view a cost performance of a project.	Low	No	R16
US08	As a team member, I can view the roles of each member of my project.	High	Yes	R01, R12
US09	As a team member, I can create new subprojects inside one project.	Medium	Yes	R08
MY ACCOUNT + LOGIN				
US10	As a team member, I can be notified via email about events in my projects.	Medium	Yes	R10
US11	As an user, I can edit my personal information.	Medium	Yes	R08
US12	As an user, I can view the status of my work in my dashboard.	Medium	Yes	R06, R10, R53
ROADMAP				
US13	As a team member, I can view a release in the roadmap of the project.	Medium	No	R06, R11
US14	As a project manager, I can edit a release in the roadmap of the project.	Medium	No	R08
US15	As a team member, I can view the automatic logs created in each activity of the project for analysing the time and the defects.	Low	No	R06, R11, R18
US16	As a team member, I can view notifications for the project about delayed work.	Medium	No	R02
US17	As a team member, I can view the velocity of a team and project (size unit / time unit).	Medium	No	R06, R10
US18	As a team member, I can view the status of the project using its velocity and the availability of resources (members) and comparing with the planned schedule.	High	No	R02
US19	As a team member, I can view for each release a list of product backlog items, goals and the schedule with its iterations.	Medium	No	R02
TASKS				
US20	As a team member, I can create a task.	Medium	Yes	R06, R14
US21	As a team member, I can generate a list of tasks using the phases and steps of the process.	High	No	R07, R18
US22	As a team member, I can view and edit a list of tasks associated to an user story, defect or other work product, prioritized and filtered.	Medium	No	R01
US23	As a team member, I can view a planned conclusion date and week for each task.	High	No	R01, R19
US24	As a team member, I can assign members to a task.	Medium	Yes	R01
BACKLOG				
US25	As a team member, I can view and edit the project backlog, prioritized and filtered.	Medium	No	R12
US26	As a team member, I can create an user story in the product backlog.	High	No	R12
US27	As a product owner, I can view/comment/validate a new user story in the project.	Low	No	R12
US28	As a team member, I can add a defect to backlog to be solve later.	Medium	No	R05
US29	As a team member, I can view for each iteration a list of product backlog, goals and the schedule with its iterations.	Medium	No	R12
US30	As a team member, I can view a suggestion in the project backlog for iteration plan.	Medium	No	R18
US31	As a team member, I can assign members to an user story.	Low	No	R12
US32	As a team member, I can view a suggestion of a member to an user story, based on the availability	Medium	No	R18
US33	As a project manager, I can define the length of each iteration depending on the process chosen.	Medium	No	R12
PRODUCTS				
US34	As a team member, I can view and edit a list of work products, prioritized and filtered.	Medium	No	R01
US35	As a team member, I can create a new work product.	Medium	No	R01
US36	As a team member, I can assign members to a work product.	Low	No	R01
PROCESSES				
US37	As a project manager, I can define the process that will be followed in the project (ISP, Scrum, Hybrid)	High	No	R12, R13, R20
US38	As a project manager, I can edit a new process from scratch or based in a old one (Phases, Steps for each phase)	High	No	R12, R13, R20
US39	As a project manager, I can change a process during the life cycle of the project.	Medium	No	R26
US40	As a project manager, I can define the velocity of an item and then calculate the effort of the team based on the size and the velocity of this item.	High	No	R19
US41	As a project manager, I can access to all versions of the processes used in my projects.	Medium	No	R21
CALENDAR				
US42	As a team member, I can edit an event in the calendar.	Medium	Yes	R05, R07
US43	As a project manager, I can add information about the team meetings.	Low	No	R09, R17
US44	As a team member, I can view a Gantt chart about the releases and work products of the project.	Medium	Yes	R05, R07, R27, R28
CHARTS				
US45	As a team member, I can view an iteration and release burndown chart.	Medium	No	R12, R27
US46	As a team member, I can view an Earned value chart per release.	High	No	R04, R12, R15, R27
US47	As a team member, I can view a Resource burnup chart.	Medium	No	R04, R07, R28
GOALS				
US48	As a team member, I can define a list of project goals (Project/Team/Individual, description, status, assigned_to, phase)	Low	No	R01
SETTINGS				
US49	As a project manager, I can assign one member to several roles in one project.	High	Yes	R20
US50	As a project manager, I can add different modules for my projects. (Time, Defect, Graphs, Documents, Repository,...)	Medium	Yes	R26
US51	As an project manager, I can create a new role for my project (id, name, description, Permissions).	Medium	No	R20
US52	As a project manager, I can define for each team member if his role is primary or secondary.	Low	No	R20
US53	As a project manager, I can edit an iteration (name, description, wkld, date, status).	Medium	No	R12
US54	As a project manager, I can create different types for issues (Default: User story, Task, Risk, Goal, Defect)	Medium	No	R05, R06
ADMINISTRATION				
US55	As an admin, I can archive or delete a project.	Low	Yes	R01, R12
US56	As an admin, I can create/delete a user (initials (automatic), name, phone, email).	Low	Yes	R01, R12
US57	As an admin, I can give/remove to a role a several access permissions.	High	Yes	R20
US58	As a project manager, I can CRUD new types for defects (Default: Documentation, Syntax, Build/Package, Assignment, Interface, Checking, Data, Function, System, Environment).	Low	No	R05
US59	As a project owner, I can assign team member availability to the project using hours per week.	High	Yes	R07, R10
OTHERS				
US60	As a member, I can insert a meeting in a template document (date, topics, decisions, risks, actions->new issue)	Low	No	R17
US61	As a member, I can view reports with automated size measurement	Medium	No	R19

Figura 4.3: Lista de *user stories* criada para o Projeto AIMS

4.4 Tecnologias e arquitetura

Em termos tecnológicos foram utilizadas algumas ferramentas de apoio ao desenvolvimento, como:

- **Dokuwiki**¹ para manter o *backlog* do que foi sendo desenvolvido ao longo do semestre e das datas definidas para as entregas principais referentes ao trabalho de investigação associado à dissertação. Contém também os relatórios e apresentações produzidos.
- **Dropbox** para armazenamento de toda a documentação gerada ao longo do projeto AIMS.
- **Redmine** com a versão 2.0, para acompanhamento do desenvolvimento do projeto, e que mantém o registo de todas as *user stories* que vão sendo cumpridas e do respetivo custo associado, bem como dos problemas que foram sendo encontrados e das reuniões diárias, semanais e mensais ao longo do semestre. Para além disso permitiu que a equipa pudesse ir estudando melhor a ferramenta num cenário real. Esta ferramenta encontra-se instalada no servidor virtual para uso comum da equipa de desenvolvimento e é possível aceder a partir do seguinte endereço 192.168.102.203:3001.
- **Ruby on Rails** com a versão 1.9.2 do Ruby, RubyGems 1.3.7 e a versão 3.2.3 de Rails, para o desenvolvimento do *plugin* AIMS.
- **Servidor virtual de produção** instalado e configurado sobre o domínio da FEUP (aims.fe.up.pt) no sistema operativo Ubuntu 10.10 do Linux (Maverick). Neste servidor está mantido um repositório com as versões dos componentes no final de cada iteração, após integração com a restante ferramenta e alguns testes de usabilidade.

O projeto encontra-se a ser desenvolvido segundo uma arquitetura MVC (*Model, View e Controller*) imposta pela tecnologia Ruby on Rails. No entanto é importante mostrar de que forma é feita a ligação entre a ferramenta e o utilizador, do ponto de vista arquitetural.

Na Figura 4.4 é possível ver como está organizada a arquitetura da ferramenta, focando a ligação entre os módulos e a base de dados do Redmine (lado direito da Figura 4.4) e do *plugin* AIMS (lado esquerdo da Figura 4.4).

¹<http://paginas.fe.up.pt/ei07057/tese/doku.php>

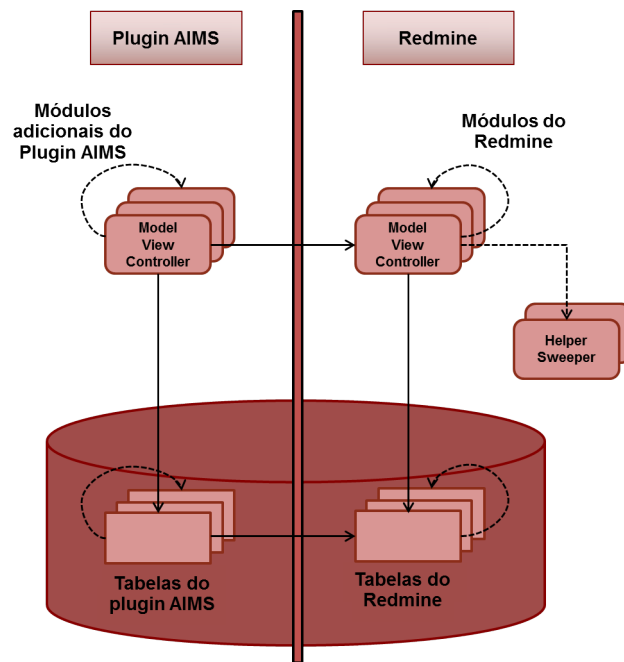


Figura 4.4: Arquitetura em alto-nível do Redmine e do *plugin AIMS*

4.5 Modelo de dados

A Figura 4.5 representa o modelo de dados que contém a relação entre as tabelas do Redmine com as tabelas criadas para o *plugin AIMS*. No diagrama encontram-se encaixilhadas as seis tabelas que dizem respeito ao *plugin AIMS*, enquanto que as restantes são tabelas pré-definidas do Redmine mas que se relacionam com as do *plugin*.

No entanto, para que fosse possível apresentar as relações mais importantes do modelo de dados, foi necessário omitir mais de metade das tabelas pré-definidas no Redmine, tabelas essas que dizem respeito às permissões dos utilizadores, ao repositório, aos documentos, ao controlo de versões, à wiki, ao blog e aos *custom_values*.

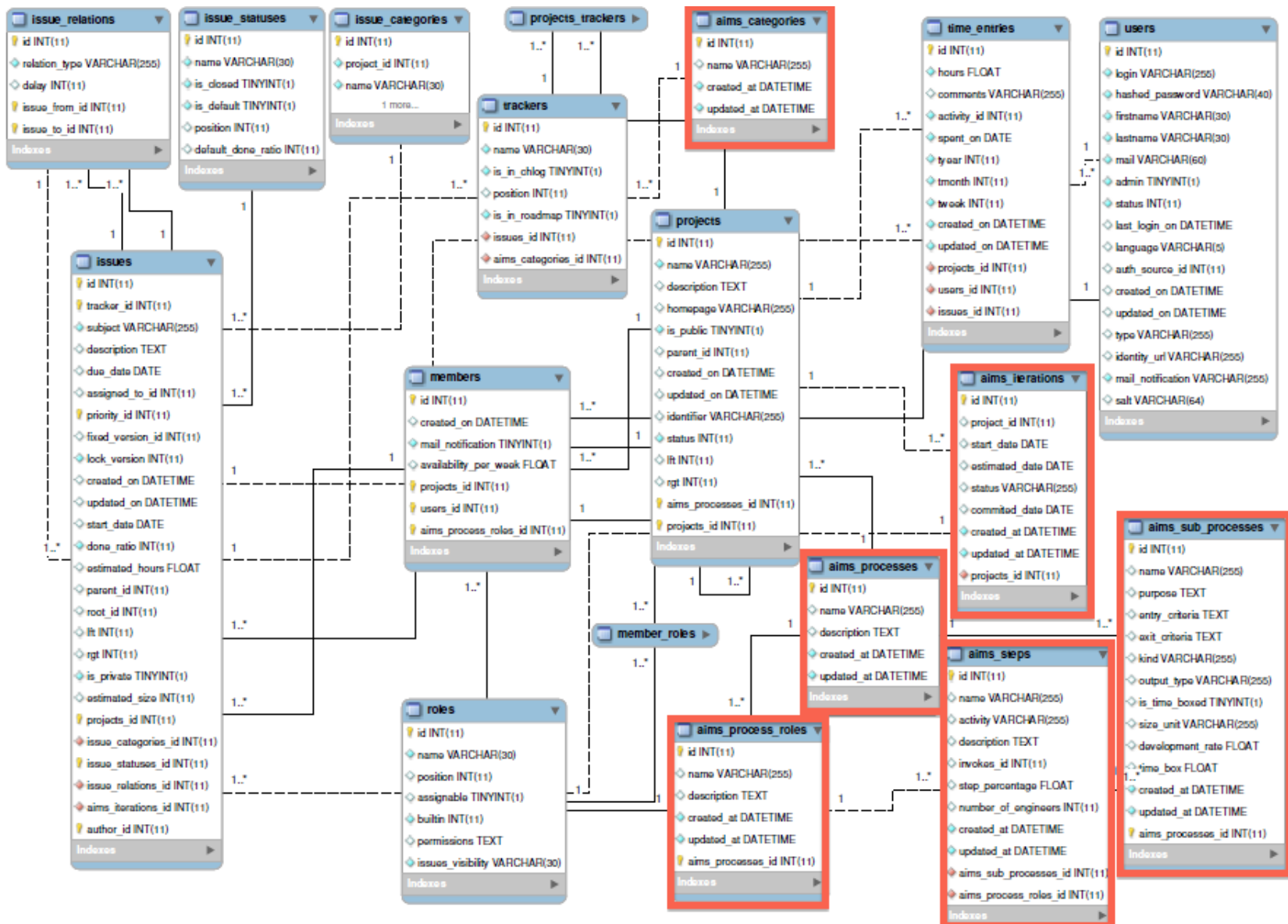
De forma a que o diagrama da Figura 4.5 seja mais compreensível foi criada a Tabela 4.1 com uma breve descrição de cada uma das tabelas apresentadas no modelo de dados.

No modelo de dados da Figura 4.5 é possível ver ligações a tracejado, que representam as relações em que a chave primária de uma tabela é também chave primária na outra tabela. Enquanto que as ligações de traço contínuo representam as relações em que a chave primária de uma tabela é chave estrangeira na outra.

Conceção e Implementação

Tabelas do Redmine	Descrição
projects	Representa cada projeto criado no Redmine.
issues	Representa todas as <i>issues</i> criadas no Redmine, por exemplo, <i>user stories</i> , <i>bugs</i> , documentos, componentes, tarefas, etc.
issue_relations	Representa a relação entre as próprias <i>issues</i> . Por exemplo uma <i>user story</i> pode conter um conjunto de tarefas e no entanto trata-se tudo de <i>issues</i> .
issue_statuses	Representa o conjunto de estados pelos quais uma <i>issue</i> pode passar, nomeadamente, aberta, cancelada, resolvida, atrasada, etc.
issue_categories	Representa a categoria a que uma <i>issue</i> pode pertencer. Apesar de poder ser usado no projeto, é um conceito que no <i>plugin</i> AIMS foi explorado de uma forma diferente do que na tabela "aims_categories".
trackers	Representa o tipo que cada <i>issue</i> pode ter. Por exemplo uma <i>issue</i> pode ser do tipo <i>User story</i> ou do tipo Tarefa.
projects_trackers	Representa a ligação entre os vários tipos de <i>issues</i> e os vários projetos.
users	Representa cada utilizador registado no Redmine.
members	Representa cada utilizador que se encontra registado no Redmine e associado a um ou vários projetos.
roles	Representa os papéis que cada membro pode ter num projeto. Por exemplo, <i>Project Manager</i> , <i>Scrum Master</i> , etc.
member_roles	Representa a ligação entre os vários membros e os vários papéis.
time_entries	Representa o registo da atividade dos utilizadores e das <i>issues</i> que vão sendo alteradas ao longo de um projeto.
Tabelas do Plugin AIMS	Descrição.
aims_iterations	Representa cada iteração criada para um projeto.
aims_processes	Representa cada processo criado no Redmine, por exemplo, TSP, Scrum, etc.
aims_process_roles	Representa os papéis que se encontram associados a um processo.
aims_sub_processes	Representa o conjunto de subprocessos que um processo pode ter, que podem ou não ser equivalentes aos tipos de <i>issues</i> (<i>Trackers</i>).
aims_steps	Representa o conjunto de passos criados e atribuídos a um determinado subprocesso.
aims_categories	Representa a categoria a que pode pertencer um tipo de <i>issue</i> (<i>Tracker</i>). Só existem 3 categorias que, basicamente, representam os módulos principais do <i>plugin</i> AIMS. E são eles: <i>Tasks</i> , <i>Backlog</i> e <i>Products</i> .

Tabela 4.1: Listagem e descrição de cada uma das tabelas apresentadas no modelo de dados da Figura 4.5

Figura 4.5: Modelo de dados do Redmine e *plugin* AIMS

4.6 Módulos da solução

Para provar que é possível desenvolver um *plugin* para Redmine que implemente TSP com Scrum, inicialmente, foram implementados alguns protótipos funcionais para teste e estudo da tecnologia Ruby on Rails, que até à data do projeto era praticamente desconhecida à equipa de desenvolvimento, o que se tornou num fator crítico para que o projeto pudesse avançar mais rapidamente.

Atualmente e usando o Redmine como ferramenta base, juntamente com o respetivo modelo conceptual, encontra-se a ser desenvolvido o *plugin* AIMS. Este *plugin* não se encontra totalmente implementado, mas a solução e interface já estão idealizadas ao pormenor no âmbito desta dissertação. Este *plugin* contém um conjunto de módulos associados aos vários componentes.

Para o componente de gestão de projetos são necessários dez módulos. Estes módulos foram desenhados com o intuito de utilizar um processo que integre TSP com Scrum, mas que ao mesmo tempo permita ao utilizador escolher apenas um dos dois processos para o desenvolvimento do seu projeto. Há ainda um módulo que segundo a arquitetura inicial existiria autonomamente, mas que acabou por se integrar na gestão de projetos, o módulo de Processos.

4.6.1 Processos

Este módulo contém uma lista de processos pré-definida com TSP ou Scrum ou TSP+Scrum ou, até mesmo, a possibilidade de o utilizador criar o seu próprio processo. Dentro de cada processo é possível definir um conjunto de elementos, nomeadamente *user stories*, *bugs*, documentos, componentes de software, etc., sobre os quais se pretende aplicar restrições, permissões e *workflows*. Cada elemento representa um subprocesso e possui um conjunto de passos e papéis associados. Por cada passo é gerada uma tarefa. Cada tarefa pertence a um elemento que pode conter um conjunto de tarefas associadas. Ou seja, uma tarefa está para um elemento, assim como um passo está para um subprocesso de um elemento.

Em relação aos Processos, também foi decidido que seria dada a hipótese ao utilizador de poder trocar de processo a meio do projeto. Embora isto não seja suposto acontecer, a verdade é que nas empresas é comum iniciarem um projeto com um processo e, mais tarde, aperceberem-se de que o processo escolhido não era o mais indicado. No entanto, esta hipótese de mudar de processo só é possível se já não existir nenhuma informação dependente de algum módulo referente ao processo que se está a utilizar.

4.6.2 Visão Geral

Este módulo contém um conjunto de informações gerais referentes ao ponto de vista da organização ou do projeto, consoante o que se estiver a ver no momento, bem como informação sobre os membros da equipa que dele fazem parte e quais os papéis atribuídos. Basicamente aparece informação referente aos campos da tabela *Project, Role, User*.

4.6.3 Equipa

Este módulo contém informação detalhada sobre os elementos da equipa e o seu desempenho.

Em relação ao seu desempenho é possível visualizar o número de horas gasto por cada elemento da equipa, bem como a disponibilidade de cada elemento da equipa. É com base nesta disponibilidade e nos seus papéis no projeto que os elementos são alocados às suas tarefas automaticamente ou manualmente. O número de horas gasto por cada elemento ajuda a que haja um equilíbrio na atribuição de tarefas.

4.6.4 Roadmap

Este módulo contém o estado do projeto em causa, que aparece como atrasado ou a tempo, consoante os dados recolhidos dos prazos de conclusão de todas as tarefas e iterações. Basta um item ter ultrapassado o prazo estipulado para que o estado do projeto passe para atrasado e automaticamente todos os elementos da equipa recebem uma notificação no email a indicar o estado do projeto.

Também é possível visualizar o estado do projeto, não só em relação à calendarização, como em relação ao custo. Ou seja, nas informações do projeto é indicado o orçamento disponível para realização do mesmo e é indicada uma lista de gastos feitos ao longo do projeto. Consoante essa lista é calculado o dinheiro gasto até ao momento e se já ultrapassou ou não o orçamento estipulado.

4.6.5 Tarefas

Este módulo contém uma lista completa de tarefas associadas a *user stories*, *bugs* ou a algum tipo de *work product*. As tarefas encontram-se apresentadas por iteração e uma iteração só pode ser fechada depois de todas as suas tarefas estarem fechadas. Cada tarefa possui uma data de início, uma descrição, um item ao qual está associada, um estado, um tamanho estimado e uma data de fim estimada. Também possui uma prioridade que é representada consoante a ordem em que se encontram apresentadas na lista de tarefas, ou seja, é necessário guardar a posição de cada tarefa na lista de cada iteração.

Dentro deste módulo é possível gerar tarefas automaticamente, ou seja, por cada elemento da lista contida no *Backlog* e no módulo Produtos é gerada uma lista de tarefas. Esse conjunto de tarefas equivale exatamente ao conjunto de passos associados ao elemento em causa, isto de acordo com o processo definido para o projeto. Esta lista de tarefas pode ser gerada para todos os elementos do projeto ou apenas por iteração.

Para além disto, na criação de uma tarefa também é possível ver uma sugestão para o elemento da equipa mais indicado a ser alocado na tarefa, consoante a seguinte ordem, os papéis que possui no projeto, o número de horas gastas na organização ou projeto, a data de fim de cada tarefa, a sua disponibilidade e o número de tarefas por completar em que se encontra alocado.

4.6.6 Backlog

Este módulo contém a lista de *user stories*, *bugs* ou outros elementos, por iteração. Cada elemento contém um conjunto de campos:

- estado que pode ser resolvido, em espera, em progresso ou em atraso;
- descrição;
- tipo de processo que é um campo omitido à vista do utilizador e utilizado pelos processos;
- unidade para o tamanho que varia consoante o tipo de item, este campo vai buscar a informação que está definida nos processos, se houver, senão é um campo editável pelo utilizador;
- velocidade em tamanho por hora, logo não é um campo editável;
- estimativa de esforço que se representa em horas e corresponde à velocidade vezes o tamanho estimado, e também não é um campo editável;
- data estimada para finalizar que corresponde ao número de horas da estimativa de esforço menos o número de horas disponíveis da pessoa atribuída, e depois calcula, consoante a data de criação, qual a data em que prevê terminar o elemento; também indica se esse elemento vai ser entregue atrasado ou não consoante a data definida para a entrega da iteração;
- prioridade que é um campo omitido ao utilizador e funciona da mesma forma que no módulo Tarefas, em que a ordem em que é apresentada na lista é a ordem de cada item;
- data efetiva de conclusão que é um campo editável onde o utilizador indica a data em que deu por terminado um *product backlog item*.

Os itens que não têm iteração atribuída, mantêm-se na *Icebox*, que é para onde vão todos os itens criados inicialmente.

4.6.7 Produtos

Este módulo possui um funcionamento semelhante ao do *Backlog*, mas a lista que contém é de *work products*, que podem ser versões de componentes de software, documentos, etc. Cada elemento contém exatamente os mesmos campos que uma *user story* ou *bug*, referidos no tópico anterior.

Para além disto, possui uma ligação entre cada versão de um *work item* a uma submissão no repositório do projeto.

4.6.8 Atividade

Este módulo já vem de raiz com o Redmine e contém um histórico de todas as alterações que foram sendo feitas ao longo do projeto, funciona como um *Log Time*.

4.6.9 Gráficos

Este módulo contém três tipos de gráficos gerados automaticamente a partir dos dados gerados no planeamento e execução do projeto.

O *earned value* (EV), baseado no TSP, é uma técnica para controlo do desempenho do projeto baseada no trabalho e esforço dispendidos para o mesmo. Na maioria das organizações, os projetos são avaliados em relação a um orçamento, olhando para o dinheiro gasto e a percentagem de trabalho feito do projeto. Ao contrário do que acontece geralmente, o EV é uma estimativa que leva em consideração re-estimativas sobre a evolução do projeto e o que vai ser preciso para concluí-lo. O EV dá aos gestores de projeto uma medição mais precisa do estado do projeto [Wes05] [Sol02].

O EV é calculado através do trabalho feito até ao momento mais o orçamento autorizado para o projeto.

A Figura 4.6 mostra um exemplo de um gráfico de *earned value*, em que a linha vermelha (PV) representa o que foi planeado e a linha azul (AC) o que realmente foi feito e gasto. Com estas duas representações obtém-se a linha verde (EV), que representa o valor ganho (*earned value*). É um gráfico deste tipo que será usado na ferramenta, mas com os dados referentes aos projetos criados, para que a equipa possa consultar a relação do trabalho feito com o que foi planeado até à data.

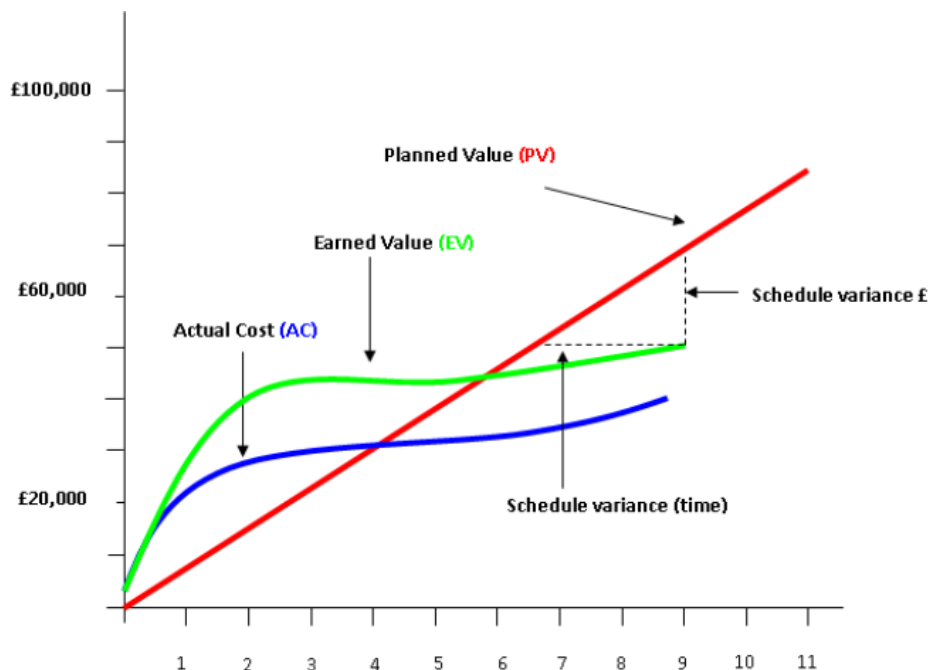


Figura 4.6: Exemplo de um gráfico com o cálculo do *earned value* [EVi12]

O *burndown chart*, baseado no Scrum, ao contrário do *earned value*, apresenta a quantidade de trabalho que ainda resta por terminar, em vez daquele que já foi feito. Este gráfico permite ver a relação entre a quantidade de trabalho que já foi feito até certo ponto no tempo e o progresso da equipa em reduzir o seu trabalho, ou seja, o que está planeado [Sch07].

A Figura 4.7 representa um simples *release burndown chart* que mostra no eixo horizontal as iterações de um projeto e no eixo vertical mostra a quantidade de *story points* ou dias previstos para o projeto. No exemplo da figura está definido que o projeto terá quatrocentos *story points* e é possível concluir que ao fim da sétima iteração já foi possível concluir quase metade do tamanho do projeto estimado inicialmente.

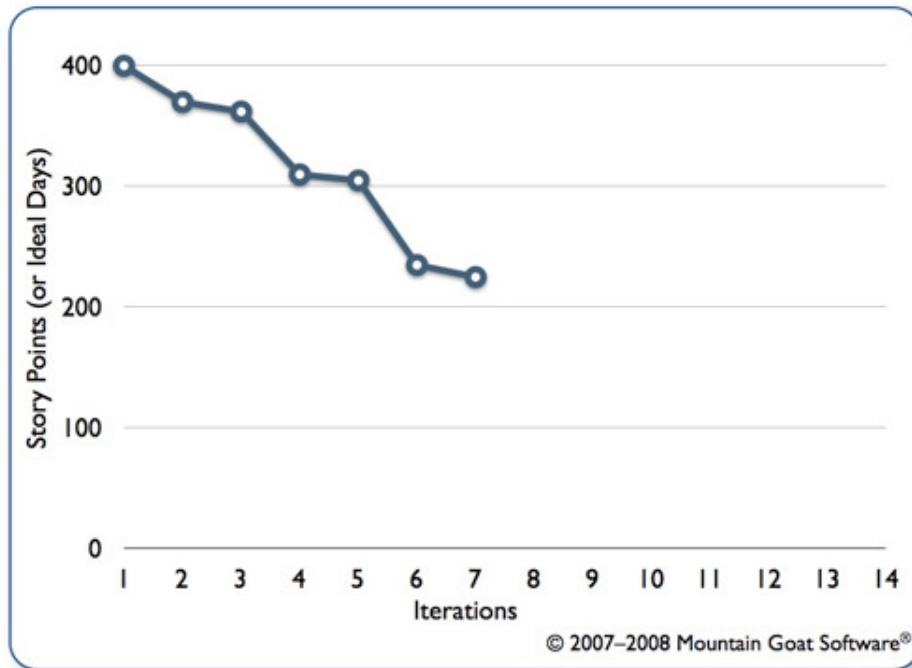


Figura 4.7: Exemplo de um gráfico *burndown* ao longo de várias iterações [Coh08]

Este tipo de gráfico é mais cativante para uma equipa de desenvolvimento, pois tem uma vista mais otimista sobre o que falta para terminar o projeto, ao contrário do *earned value*. Embora a presença de um gráfico não invalide o outro, pois ambos os gráficos são úteis para mostrar os dois tipos de análise sobre o projeto, daí o *plugin* AIMS incluir os dois.

Por último, neste módulo também é apresentado o *resource burnup* que não se limita a assumir o tempo máximo para todos os elementos da equipa, como muitas ferramentas, ele compara a disponibilidade dos elementos da equipa. Isto é bastante útil, porque muitos dos recursos humanos, apesar de trabalharem n horas por semana numa empresa, podem estar envolvidos em simultâneo em vários projetos o que reduz o tempo disponível dos recursos num determinado projeto [Rai06].

A Figura 4.8 representa um exemplo de um gráfico simples de *resource burnup* que representa uma iteração de um projeto com uma equipa de quatro elementos. A barra vermelha (barra mais escura) representa o tempo gasto por um elemento no projeto, em determinado momento de uma iteração e a barra verde (barra mais clara) representa o tempo restante para cada elemento. Ou seja, as duas barras juntas representam a disponibilidade de cada elemento numa determinada iteração do projeto.

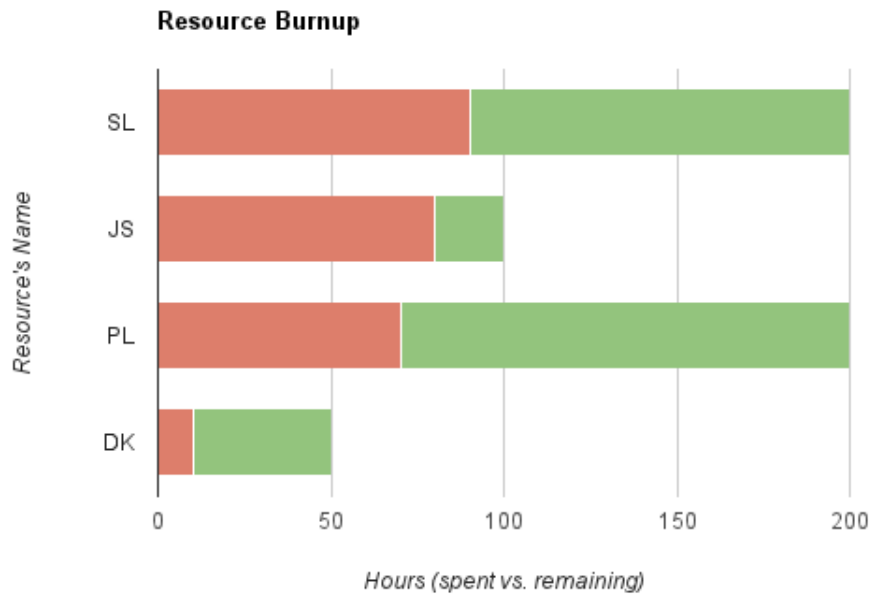


Figura 4.8: Exemplo de um gráfico simples de *resource burnup* para uma iteração

4.6.10 Definições

Módulo em que é possível alterar definições sobre o projeto, por exemplo, adicionar ou retirar módulos ao projeto, criar novos tipos de itens, adicionar novos utilizadores, definir papéis ou alterar as informações do projeto.

Para o *plugin* AIMS apenas serão incluídos os módulos Equipa, *Roadmap*, Tarefas, *Backlog*, Produtos, Gráficos e Processos, porque terão de ser criados pela equipa de desenvolvimento. Alguns de raiz, como é o caso do *Backlog*, Produtos, Equipa e Processos, e outros com base em módulos de Redmine já existentes, como é o caso do módulo Tarefas que será criado com base no módulo *Issues* já existente, e também o caso do módulo *Roadmap* que será criado com base no módulo *Roadmap* já existente no Redmine.

No caso dos módulos Visão Geral, Atividade e Definições, estes já se encontram completamente implementados no Redmine, apenas irão sofrer pequenas alterações. Por exemplo, na Visão Geral onde aparece a informação do projeto, será necessário adicionar o campo de data de início às informações do projeto, o que será feito recorrendo aos *custom fields* do Redmine.

Para além de todos estes módulos, após um estudo do Redmine, também foram considerados para este projeto, os módulos referentes ao mapa de Gantt e ao calendário do projeto.

No capítulo seguinte, Capítulo 5, é possível ver um pouco da implementação de todos os módulos referidos neste capítulo.

Conceção e Implementação

Capítulo 5

Apresentação e Avaliação da Solução

Este capítulo apresenta o que já foi implementado até ao momento, a forma como será avaliada a solução final deste trabalho de dissertação e as conclusões que podem ser retiradas para o trabalho futuro, a partir da investigação feita.

5.1 Cenário implementado

De tudo o que foi referido na implementação, até ao momento já foram desenvolvidos alguns protótipos, para provar que a implementação da solução utilizando o Redmine como base é possível. Como tal, para que seja mais perceptível a forma como o projeto está a ser implementado, são apresentados alguns *printscreens* com o contexto de um cenário realista, mas hipotético, de todas as funcionalidades criadas de raiz até ao momento.

Inicialmente, foi criada uma empresa designada por Strongstep. Numa fase inicial a vista da organização é criada com base na vista de projeto, pois uma organização pode ser vista como um conjunto de projetos e departamentos com restrições e processos diferentes.

Na criação do conceito de organização, Figura 5.1, foi indicado que a organização era pública, ou seja, podia ser acedida por qualquer utilizador da ferramenta. E foram inseridos o nome, a descrição, um identificador único e uma *homepage*. Estas informações já vinham de raiz com o Redmine, mas para além disto, também foi indicado o processo que a organização adota, neste caso o TSP.

Uma vez criada a organização e mais alguns departamentos associados, tal como indica a Figura 5.2, foi também criado o projeto AIMS como um subprojeto do Departamento de Desenvolvimento de Software da empresa Strongstep, inicialmente sem nenhum processo associado. Apesar da empresa utilizar o processo TSP, ela pode optar por escolher processos diferentes para os seus projetos, ou até mesmo criar novos processos. Para criar um processo, basta ir ao módulo Processos, Figura 5.3, onde aparece uma listagem de todos os processos existentes e clicar no link "Novo Processo".

Apresentação e Avaliação da Solução

Redmine

Home My page Projects Administration Help

Logged in as admin My account Sign out

Search:

New project

Name *

Subproject of

Description [Text formatting](#)

Identifier *
Length between 1 and 100 characters. Only lower case letters (a-z), numbers, dashes and underscores are allowed. Once saved, the identifier cannot be changed.

Homepage

Public

Process

Modules

<input checked="" type="checkbox"/> Issue tracking	<input checked="" type="checkbox"/> Time tracking	<input type="checkbox"/> News	<input type="checkbox"/> Documents	<input checked="" type="checkbox"/> Files
<input type="checkbox"/> Wiki	<input checked="" type="checkbox"/> Repository	<input type="checkbox"/> Forums	<input checked="" type="checkbox"/> Calendar	<input checked="" type="checkbox"/> Gantt
<input checked="" type="checkbox"/> Team	<input checked="" type="checkbox"/> Products	<input checked="" type="checkbox"/> Backlog	<input checked="" type="checkbox"/> Tasks	

Trackers

<input checked="" type="checkbox"/> Bug	<input checked="" type="checkbox"/> User Story	<input checked="" type="checkbox"/> Document	<input checked="" type="checkbox"/> Component	<input checked="" type="checkbox"/> Task
---	--	--	---	--

Powered by Redmine © 2006-2012 Jean-Philippe Lang

Figura 5.1: Formulário para criação de um projeto (módulo Visão Geral)

Redmine

Home My page Projects Administration Help

Logged in as admin My account Sign out

Search:

Projects

View all issues | Overall spent time | Overall activity

Strongstep

A Strongstep é uma empresa especializada em Engenharia de Software que contribui para a melhoria de qualidade de software em Portugal e no Mundo. A sua atuação assenta sob princípios pragmáticos de eficiência e excelência, sustentada por uma equipa experiente, pela utilização de melhores praticas e parcerias com instituições de referência mundial...

- Departamento de Desenvolvimento de Software
- Departamento de Recursos Humanos

Also available in: [Atom](#)

Figura 5.2: Vista de todos os projetos públicos

Strongstep

Home My page Projects Administration Help

Logged in as admin My account Sign out

Search:

Overview Backlog Products Team Tasks Activity Issues New issue Gantt Calendar Processes Settings

Processes

	Process Name	Description	Creation Date
<input checked="" type="checkbox"/>	TSP		2012-06-26 23:32:34 UTC
<input type="checkbox"/>	Scrum		2012-06-28 13:06:41 UTC

Processes

New process

Name*:

Description:

Figura 5.3: Vista e formulário para criação de processos (módulo Processos)

Apresentação e Avaliação da Solução

Após criar o processo TSP+Scrum, é possível associá-lo ao projeto AIMS ao editar as suas informações tal como mostra a Figura 5.4.

The screenshot shows the 'Settings' page for a project named 'AIMS' in the 'Departamento de Desenvolvimento de Software' department. The page includes a navigation menu with options like Overview, Backlog, Products, Team, Tasks, Activity, Issues, New issue, Gantt, Calendar, Processes, and Settings. The 'Settings' section is active, showing fields for Name (AIMS), Subproject of (Departamento de Desenvolvimento de Software), Description (with a rich text editor), Identifier (aims), Homepage (http://192.168.102.203:3000), Public (checked), and Process (TSP+Scrum). There are also checkboxes for Trackers: Bug, User Story, Document, Component, and Task, and a 'Save' button at the bottom.

Figura 5.4: Formulário para edição de projetos (módulo Definições)

Com o processo criado, é necessário criar e editar os seus subprocessos e os respetivos passos, tal como mostra a Figura 5.5 e a Figura 5.6, respetivamente. Nestas duas figuras é possível ver que para o processo TSP+Scrum foram adicionados papéis como *Scrum Master*, *Team Member* e *Product Owner* que, num trabalho futuro, irão ter restrições associadas. Para além de papéis também foram criados subprocessos para *user stories*, *bugs* e documentos. Cada subprocesso tem um conjunto de passos próprios a seguir e onde é definida a percentagem de esforço associada. Na Figura 5.6 é possível ver um exemplo de um conjunto de passos já definidos para uma *user story*. Também é possível ver do lado direito da janela, informação criada para este subprocesso, nomeadamente a velocidade prevista para o desenvolvimento de *issues* deste tipo e a unidade em que são medidas, ou seja, a equipa deste projeto demora uma hora a implementar dois *story points*.

The screenshot shows the 'TSP+Scrum' process configuration page. It features two main sections: 'Roles' and 'Subprocesses'. The 'Roles' section has a table with columns for Role Name, Description, and Actions. The 'Subprocesses' section has a table with columns for Subprocess Name, Kind, Description, and Actions. On the right side, there is a sidebar with the title 'TSP+Scrum' and a description 'Integração de TSP com Scrum', along with 'Edit' and 'Delete' buttons.

Role Name	Description	Actions
Scrum Master		[Edit] [Delete]
Team Member		[Edit] [Delete]
Product Owner		[Edit] [Delete]

Subprocess Name	Kind	Description	Actions
User Story			[Edit] [Delete]
Bug			[Edit] [Delete]
Document			[Edit] [Delete]

Figura 5.5: Vista das informações de um processo (módulo Processos)

Uma vez definido o processo para o projeto AIMS, será necessário criar uma iteração que é visível a partir dos módulos de Tarefas, Produtos ou *Backlog*, como tal, também pode ser criada a partir de qualquer um dos módulos. A Figura 5.7 mostra o módulo *Backlog* já com duas iterações criadas e com a *icebox*, que já é criada por defeito e que armazena as *issues* que não têm iterações

Apresentação e Avaliação da Solução

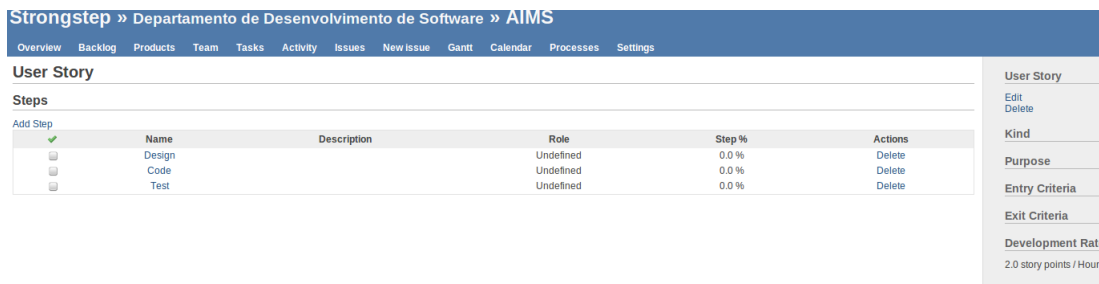


Figura 5.6: Vista do conjunto de passos de um subprocesso (módulo Processos)

atribuídas. Nesta figura também é possível ver três *user stories* e um *bug* já criados e que foram alocados a diferentes iterações através da funcionalidade de *drag and drop*.

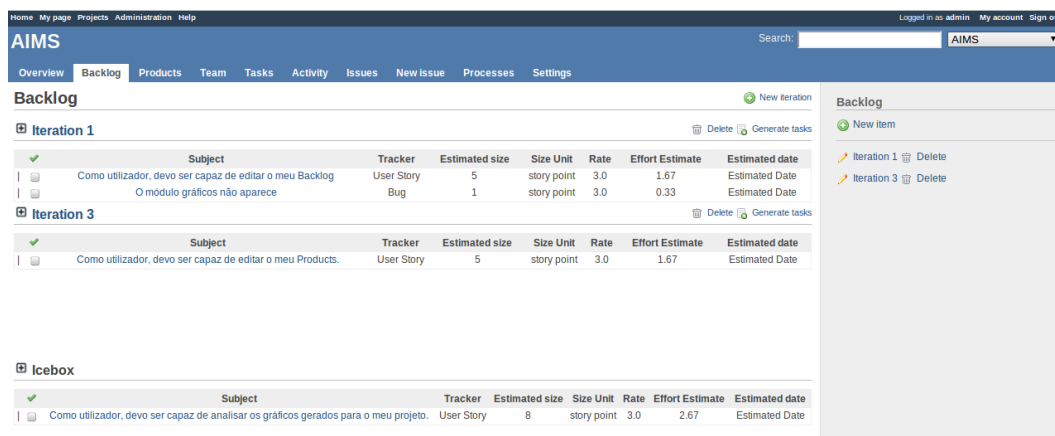


Figura 5.7: Vista do módulo *Backlog* e do respectivo conjunto de iterações

Na Figura 5.7, mais precisamente na tabela da Iteração 1 ou 3, é possível ver os campos atribuídos a cada *issue*.

- O **visto** é apenas para seleção de uma ou várias *issues* em simultâneo;
- A **estrela** é apenas para realçar a importância de alguma *issue*;
- O **nome** de cada *issue*;
- O **tipo** de *issue* que representa. Neste caso, são apresentadas três *user stories* e um *bug*;
- O **tamanho estimado** que se trata de um campo editado *in place* para que seja o utilizador a inserir um valor;
- A **unidade do tamanho** e a **velocidade** são valores retirados diretamente do que foi definido no processo TSP+Scrum;
- O **esforço estimado** representa o tamanho estimado vezes a velocidade de cada *issue*;
- A **data estimada** ainda não se encontra implementada, mas está pronta para ser calculada de forma sequencial consoante a ordem das *issues*.

Apresentação e Avaliação da Solução

Apesar da prioridade não ser um campo diretamente visível ao utilizador, ela é definida internamente consoante a ordem em que se encontram as *issues* dentro da iteração. Esta ordem pode ser alterada dentro da própria iteração através da funcionalidade de *drag and drop*.

Tanto a *user story* como qualquer outra *issue* podem ser criadas através do formulário representado na Figura 5.8. Neste caso, a figura mostra a criação de um "Relatório Técnico" do tipo Documento, onde é possível atribuir um elemento responsável, desde que esteja alocado no projeto.

The screenshot shows the 'New issue' form in the Strongstep application. The form is for creating a 'Document' type issue with the subject 'Relatório Técnico'. The description field contains the text 'Contém a arquitetura da aplicação.' The form includes fields for Status (New), Priority (Normal), Assignee (Redmine Admin), Parent task, Start date (2012-06-28), Due date (2012-07-11), Estimated time, and % Done (0%). There are also buttons for 'Create', 'Create and continue', and 'Preview'.

Figura 5.8: Formulário para criação de uma *issue* de qualquer tipo

Depois de criada uma *issue* do tipo Documento ela passa automaticamente para a icebox do módulo Produtos, como se pode ver na Figura 5.9 com o documento de "Gestão de projetos".

Esta distinção entre o módulo Produtos ou *Backlog* é feita consoante o tipo de *issue*. Inicialmente são definidas *user stories* e *bugs* para o módulo *Backlog*, enquanto que para o módulo Produtos aceita *issues* do tipo documentos ou componentes. Futuramente o utilizador ao criar os seus próprios tipos de *issues* poderá definir em que módulo pretende geri-las.

A Figura 5.9 representa o módulo Produtos e, à semelhança do módulo anterior, também foi criado de raiz e já se encontra implementado com as mesmas funcionalidades que o anterior e com integração com o módulo dos Processos, que se trata da parte mais relevante para este trabalho de investigação.

Depois de definidas as iterações, processos e *issues* que terão de estar completas, é possível gerar tarefas mais específicas de forma automática. No módulo Tarefas existe um botão "Gerar tarefas" que permite gerar e visualizar novas tarefas por iteração. As tarefas são geradas tendo em conta os passos definidos para cada subprocesso. Se um tipo de *issue* não possuir um subprocesso associado, então para esse tipo não são geradas tarefas automaticamente.

Apresentação e Avaliação da Solução

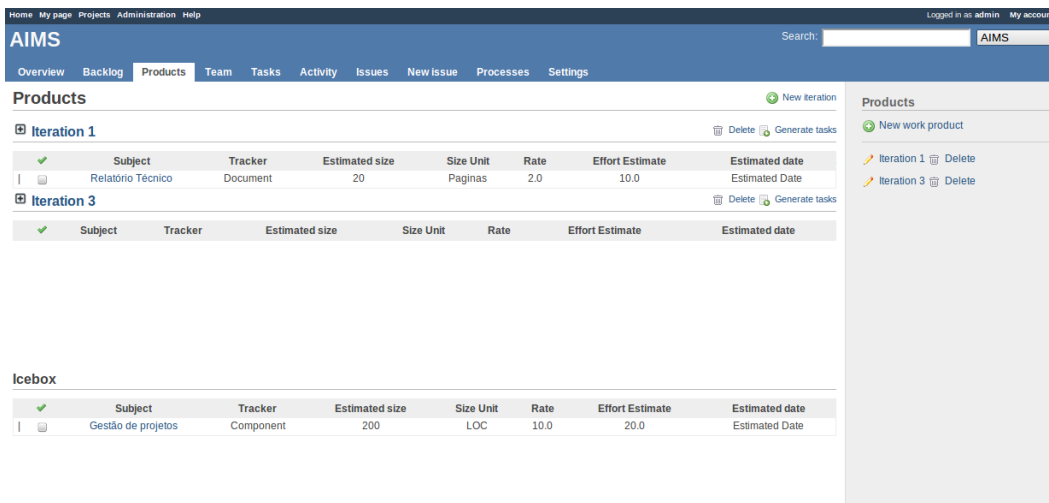


Figura 5.9: Vista do módulo Produtos e do respetivo conjunto de iterações

No módulo Tarefas é possível ver uma lista de tarefas em que repete o nome da *user story* e do documento e após o hífen aparece o nome do respetivo passo definido nos Processos (Figura 5.6). Este módulo também foi criado completamente de raiz.

No final de todo este processo é possível visualizar um conjunto de informações sobre o projeto no módulo Visão Geral, nomeadamente o número de *issues* por resolver. Este módulo já vem implementado no Redmine, tal como mostra a Figura 5.10, e será utilizado nesta ferramenta.

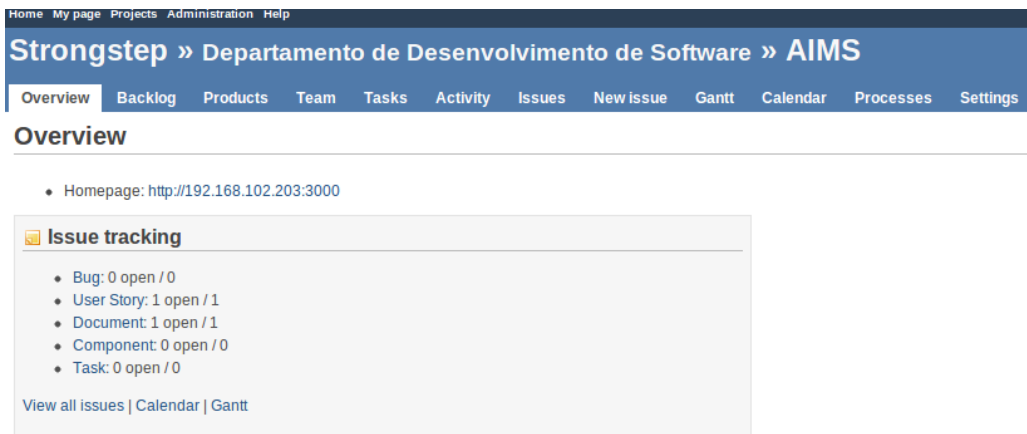
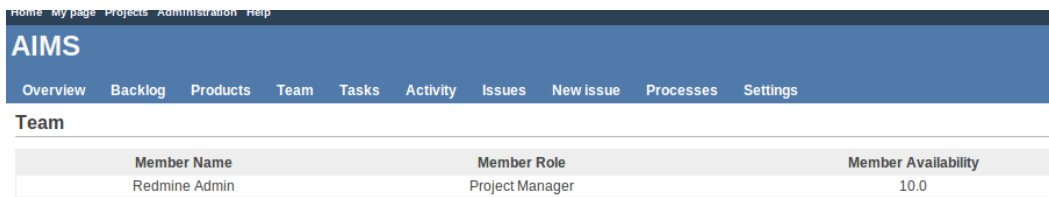


Figura 5.10: Vista do módulo Visão Geral e respetivas informações do projeto

Por último, também já se encontra implementado o módulo Equipa, Figura 5.11, que apesar de ser criado de raiz ainda se encontra numa fase inicial e será uma das próximas funcionalidades a desenvolver, para que seja possível otimizar a vista da equipa e a evolução da mesma em relação ao projeto.

Apresentação e Avaliação da Solução



Member Name	Member Role	Member Availability
Redmine Admin	Project Manager	10.0

Figura 5.11: Vista do módulo Equipa

5.2 Validação da solução

Para validar a solução final é necessário saber quando parar de evoluir o componente de gestão de projetos.

Até ao momento, o consórcio AIMS já validou os protótipos implementados e que continuarão a ser desenvolvidos e integrados até Setembro de 2012. No final desse mês será feita a integração na empresa piloto Multicert, onde será testada, através de testes de usabilidade, a primeira fase da aplicação junto dos elementos da Multicert e validada pela mesma após esta integração final.

Apresentação e Avaliação da Solução

Capítulo 6

Conclusões

6.1 Resultados alcançados

Do ponto de vista deste trabalho de investigação, a lista seguinte contém os objetivos definidos inicialmente para a implementação deste *plugin* (ver secção 1.2) e de que forma foram alcançados:

- Permitir a monitorização de tempo, prazos, defeitos, etc.

A monitorização de tempo e prazos já estavam implementados no Redmine e foram adaptados aos novos *itens* do *plugin* AIMS, ao mesmo tempo que se automatizou o cálculo de alguns dos prazos estimados;

- Permitir a gestão do projeto e da sua qualidade.

A gestão do projeto e da sua qualidade foram alcançadas com a inclusão dos novos processos integrados, TSP e Scrum;

- Conter um vista individual, de equipa e organizacional sobre cada projeto.

A vista individual e organizacional já se encontram implementadas. A vista da equipa também já está implementada mas precisa de mais funcionalidades para ser mais otimizada. Este trabalho será desenvolvido ao longo dos próximos dois meses;

- Suportar metodologias ágeis, como o Scrum.

O suporte de Scrum também foi implementado através da definição de papéis e do módulo do *Backlog*;

- Integrar com as práticas definidas no CMMI.

A integração com as práticas do CMMI nível 2 foi conseguida através do mapeamento com Scrum e TSP. No entanto, ainda necessita de ser melhorada;

- Suportar os formulários, gráficos e relatórios baseados nos parâmetros do TSP.

O suporte de formulários, gráficos e relatórios ainda não se encontra implementado. Este objetivo é alcançado com a implementação do módulo Gráficos, que é um dos trabalhos futuros;

Conclusões

- Integrar com outras ferramentas de gestão de projetos.

A integração com outras ferramentas de gestão de projetos ainda está num estado pouco evolutivo, pois atualmente o *plugin* apenas permite integração com o Redmine. Futuramente irá integrar com o Jira;

- Suportar integração entre Scrum e TSP.

A integração com Scrum e TSP foi conseguida através do mapeamento de conceitos e colocando-os em prática através do uso dos módulos *Backlog* e *Produtos*;

- Integrar gestão de projetos com gestão de processos.

A gestão de processos foi implementada incluindo os conceitos de subprocessos e passos. Este módulo vai permitir agilizar o processo de desenvolvimento de software, pois é o motor das automatizações na ferramenta. Desde a geração de tarefas até ao cálculo de estimativas, quanto melhor estiver um processo definido mais simples e prática se torna a gestão dos projetos.

Concluindo, não foi possível implementar todos os objetivos do componente, sobretudo, porque a seleção da ferramenta base ocupou mais tempo que o previsto inicialmente. Isto aconteceu pois foi feito um estudo inicial a prever a utilização do Jira e não foi possível avançar devido aos custos da licença e como tal foi necessário voltar atrás num dado momento do projeto para utilizar outra ferramenta de base, nomeadamente o Redmine foi a ferramenta escolhida.

No entanto todos os aspetos do componente já estão concebidos e a sua implementação está prevista para os dois próximos meses. Ou seja, segundo o planeamento do projeto AIMS vai ser possível recuperar o tempo gasto na conceção do *plugin* durante os dois próximos meses.

Após a comparação feita entre quatro das muitas ferramentas de gestão de projetos concorrentes existentes, é possível concluir que o cenário mais indicado para a criação desta ferramenta seria o desenvolvimento de raiz, para eliminar qualquer tipo de dependência à partida. No entanto, desenvolver um *plugin* a partir de uma ferramenta base *opensource*, como é o caso do Redmine, permitiu criar um *plugin* que já se encontra a ser desenvolvido de forma independente e em que facilmente é adaptado a qualquer ferramenta de gestão de projetos. Logo, mais tarde, será possível criar de raiz o núcleo da ferramenta do projeto com as funcionalidades básicas e incluir o *plugin* AIMS desenvolvido ao longo desta primeira fase do projeto, referente ao nível 2 de CMMI.

O objetivo principal deste trabalho de dissertação foi provar que a gestão de qualquer projeto numa empresa que já utiliza as práticas do CMMI, se torna mais eficaz se recorrer ao uso do TSP e Scrum, e sobretudo provar que esta integração é possível. Uma vez que o mundo da engenharia de software está em constante evolução também é importante que esta nova ferramenta seja capaz de se integrar com outros componentes ou *plugins* de outras ferramentas de gestão de projetos, para que não fique desatualizado. Por exemplo, uma empresa que esteja a utilizar esta nova ferramenta, não tem de abandoná-la simplesmente porque existem funcionalidades e metodologias mais recentes, pode simplesmente integrar essas novas funcionalidades na ferramenta de base que já se encontra em utilização na empresa. Desta forma também não perde tantos recursos,

tempo ou dinheiro com a formação e adaptação dos seus funcionários, como aconteceria se houvesse uma mudança para uma ferramenta completamente nova de cada vez que surgisse uma nova atualização.

Em relação aos aspetos inovadores do trabalho (ver secção 1.2), o sucesso dos resultados alcançados refletem-se sobretudo na automatização da gestão de projetos através da definição de processos mais flexíveis. Existe um conjunto de aspetos que é comum a qualquer tipo de projeto, nomeadamente, criar tarefas com base nas *user stories* e de forma a seguir um *workflow* próprio. Este processo foi automatizado, juntamente com o cálculo de estimativas, o que vem acrescentar eficácia à gestão de projetos. Também há uma grande preocupação com a usabilidade, pois quanto mais interativo for o uso da ferramenta mais simples se torna a sua aprendizagem e muitas ferramentas no mercado penam por falta dessa usabilidade. Para esta ferramenta já foram implementadas funções de *drag and drop* e de edições de campos *in place*, o que vem contribuir para uma interface mais amigável e intuitiva.

Para terminar, não é útil escolher um único processo para uma empresa, tem de haver hipótese de escolha. É muito comum haver empresas com múltiplos projetos em que cada um se adequa a um processo diferente. No entanto, com a integração do Scrum e TSP com as práticas do CMMI será possível utilizar este único processo em muitos mais contextos. Se a empresa apenas tiver Scrum ou TSP o número de projetos em que se adapta cada um deles é menor do que se houver integração dos dois. Por exemplo, supondo dois cenários na mesma empresa, um projeto com uma equipa distribuída e um projeto com uma equipa local. No primeiro caso seria mais adequado utilizar Scrum por ser um processo simples, enquanto que no segundo caso o SEI recomenda o uso de TSP [Dal09]. Com este novo *plugin* AIMS que integra os dois processos, ao mesmo tempo que permite escolher apenas um deles, ou então criar o seu próprio processo usando outros como base, será possível adaptar um único processo a uma maior quantidade de projetos da empresa, ao mesmo tempo que automatiza algum do trabalho que em muitas ferramentas é feito manualmente.

Desta forma, vem facilitar o trabalho do gestor de projeto, que em vez de perder tempo a trabalhar com a ferramenta, pode realmente investir o seu tempo na melhoria e adaptação do processo em que a equipa está a desenvolver o projeto. Assim como também facilita o trabalho da equipa na inserção de dados e aprendizagem de processos, pois a ferramenta irá encaminhar os seus utilizadores finais consoante o processo escolhido para cada projeto, aliado ao facto de se estar a tornar uma ferramenta bastante interativa e com apenas o mínimo de informação a preencher.

6.2 Trabalho futuro

Tal como já foi referido inicialmente, este projeto foi pensado para dois anos, em que apenas o primeiro ano diz respeito à implementação do CMMI-DEV nível 2 de maturidade. Como tal, a integração desta primeira parte da ferramenta está pensada para meados do mês de Setembro.

Neste momento encontra-se a ser desenvolvido o módulo Gráficos, para apresentação e análise dos dados históricos dos projetos, e o módulo Equipa, já referido na secção 4.6.

Conclusões

Durante os dois próximos meses de trabalho serão melhorados alguns aspetos de usabilidade do projeto e implementadas novas funcionalidades referidas na secção 4.6 onde se explicam os módulos em detalhe. Durante os meses de Setembro, Outubro e Novembro de 2012, esta primeira parte do projeto será integrada na empresa piloto, Multicert, onde serão aplicados testes de usabilidade e formação na utilização da ferramenta e melhorado algum aspeto, consoante a necessidade.

A partir de Novembro serão implementadas novas funcionalidades a pensar na integração com as práticas do CMMI-DEV nível de maturidade 3 e de Janeiro a Março de 2013 serão feitos novos testes de usabilidade e integração com a empresa piloto Multicert.

Anexo A

Elementos do TSP

As tabelas seguintes representam as diretrizes para implementação dos elementos do TSP e são referidas nas tabelas do Anexo B de forma a relacionar os elementos do TSP com as práticas utilizadas nas áreas de processo do nível de maturidade 2 de CMMI.

Grouping / Name	Description	Notes
Launch scripts		
LAU	Team launch: to guide teams in launching a software-intensive project	
LAU1	Launch meeting 1 - launch overview and kick-off	Step 1 in script LAU
LAU2	Launch meeting 2 - roles and goals	Step 2 in script LAU
LAU3	Launch meeting 3 - strategy, process, support	Step 3 in script LAU
LAU4	Launch meeting 4 - overall team plan	Step 4 in script LAU
LAU5	Launch meeting 5 - quality plan	Step 5 in script LAU
LAU6	Launch meeting 6 - detailed next-phase plans	Step 6 in script LAU
LAU7	Launch meeting 7 - risk assessment	Step 7 in script LAU
LAU8	Launch meeting 8 - management meeting preparation	Step 8 in script LAU
LAU9	Launch meeting 9 - wrap-up management meeting	Step 9 in script LAU
LAUPM	Launch postmortem meeting - postmortem on the launch	Step PM in script LAU
REL	Team relaunch	
REL1	Relaunch meeting 1 - status and management update	
Development scripts		
DEV	Overall new development and enhancement process	
MAINT	Overall maintenance and enhancement process	
ANA	Impact analysis process	

Figura A.1: Tabela 1 de *Scripts* [MW05]

Elementos do TSP

Grouping / Name	Description	Notes
HLD	High-level design process	
IMP	Implementation process	
IMP6	Unit test and test development process	Step 6 in script IMP
INS	Inspection process	
PM	Project postmortem process	
REQ	Requirements process	
TEST	Release test process	
TEST1	Product build process	Step 1 in script TEST
TEST2	Integration process	Step 2 in script TEST
TEST3	System test process	Step 3 in script TEST
TESTD	Test defect-handling process	
Other scripts		
MTG	General meeting process	Used as the basis for most meeting scripts
STATUS	Management and customer status meeting	
WEEK	Weekly team meeting	

Figura A.2: Tabela 2 de *Scripts* [MW05]

Grouping / Name	Description	Notes
Launch forms	Asterisk (*) items or equivalents are implemented in the TSP workbook (see Section 5.4)	
GOAL	* Team goals	
INV	Process inventory	
ITL	* Issue/risk tracking log	
MTG	Meeting report form	
PIP	Process improvement proposal	
ROLE	* Team role assignment	
ROLEMX	Role assignment matrix	
SCHED	* Schedule planning template	
STRAT	Strategic planning form	
SUMDI	* Defects injected summary	
SUMDR	* Defects removed summary	
SUMP	* Plan summary form	
SUMQ	* Quality summary form	
SUMS	* Program size summary	
SUMT	* Development time summary form	
SUMTASK	* Task plan summary	
TASK	* Task planning template	
Development forms		
DEFECT	Defect reporting form	
INS	Inspection report	
TESTLOG	Test log	

Figura A.3: Tabela 1 de Formulários e outros artefatos [MW05]

Elementos do TSP

Grouping / Name	Description	Notes
Other forms		
LOGD	* Defect recording log	
LOGT	* Time recording log	
WEEK	* Weekly status report	Modified versions of form WEEK are used in each launch meeting.

Figura A.4: Tabela 2 de Formulários e outros artefatos [MW05]

Grouping / Name	Description	Notes
Role manager specifications	The default set of roles to be assumed by members of the team: customer interface manager, design manager, implementation manager, test manager, planning manager, process manager, quality manager, and support manager	
Customer interface manager	Customer interface manager responsibilities: customer focus, define requirements, manage requirement changes, establish and manage requirement standards, and reporting	A "line" role manager
Design manager	Design manager responsibilities: lead the design, manage design changes, establish and manage design standards, and reporting	A "line" role manager
Implementation manager	Implementation responsibilities: lead the implementation, manage implementation changes, establish and manage the implementation standards, and reporting	A "line" role manager
Test manager	Test manager responsibilities: test planning, test support, test analysis, and reporting	A "line" role manager
Planning manager	Planning manager responsibilities: lead team planning, track team progress, and reporting	A "staff" role manager
Process manager	Process manager responsibilities: process support, tracking, analysis, process problems and process improvement proposal handling and reporting	A "staff" role manager
Quality manager	Quality manager responsibilities: quality support, quality tracking, quality analysis, and reporting	A "staff" role manager
Support manager	Support manager responsibilities: tool support, configuration management, change control, reuse, and reporting	A "staff" role manager
Other role specifications		
Meeting roles	Meeting role descriptions: chairperson, recorder, facilitator/timekeeper, attendees	
Inspection roles	TSP inspection process roles and responsibilities: moderator, producer, recorder, timekeeper, reviewers	
Team leader	TSP team leader responsibilities: leadership, people management, team coaching, quality management, project management, team responsibilities	
Team member	TSP team member roles and responsibilities: personal discipline, personal management, and team responsibilities	

Figura A.5: Tabela de Papéis [MW05]

Elementos do TSP

Grouping / Name	Description	Notes
Preparation checklists		
PREPL	Preparation for launch	
PREPR	Preparation for relaunch	
Launch guidance		
Launch coach	Launch guidelines for the TSP coach	
Marketing	Launch guidelines for marketing management presentation	
Other attendees (2)	Launch guidelines for TSP coach	
Senior Management	Launch guidelines for senior management presentation	
Team leader (2)	Launch guidelines for team leader	
Team members (2)	Launch guidelines for team members	
Other pre-launch assets		
Initial contact letter	TSP launch preparation	
Preparation package cover letter	TSP launch preparation material	
Preparation package instructions	TSP launch preparation material	
Default guidelines		
Planning guidelines	SEI-provided benchmark planning metrics	
Quality guidelines	SEI-provided benchmark quality metrics	
Executive assets		
Plan assessment checklist	Team plan review questions; a quick start for an executive reviewing a TSP team's plan	These assets can be found in <i>Winning with Software</i> [Humphrey 02].
Quarterly review checklist	Project review questions; a quick start for senior managers to probe the status of a TSP project	
TSP introduction strategy	A generic procedure and timeline for TSP implementation in an organization	
Other specifications and assets		
NOTEBOOK	Storage for project artifacts	
STATUS	Management status report	
SUMMARY	Project analysis report	
TSP workbook (individual and consolidated)	Automated individual and team (consolidated) plans and actuals for size, effort, defects, and schedule; functionally equivalent versions of asterisked (*) items above under Forms are included in the TSP Workbook	Excel-based; provided by the SEI as part of the licensed TSP product suite
Checkpoint Review	A review of the project to date conducted by the TSP coach or other process expert	
Weekly Meeting Minutes	Minutes from weekly team meetings	

Figura A.6: Tabela com outros elementos [MW05]

Elementos do TSP

Grouping / Name	Description	Notes
Training and authorization		
SEI training records	SEI-maintained records of everyone reported by SEI-authorized instructors to have finished any of the training classes listed below	
<i>Introduction to Personal Process</i>	Training for team members who are not software engineers (2 days)	
<i>PSP for Engineers</i>	Training for software developers (10 days)	
<i>TSP Executive Seminar</i>	Executive briefing on PSP and TSP, including benefits and the TSP introduction strategy (1 day)	
<i>Managing TSP Teams</i>	Training for people managing TSP teams (3 days)	
<i>PSP Instructor Training</i>	Training to become a PSP instructor (5 days)	Offered only through the SEI; prerequisite is successful completion of <i>PSP for Engineers</i>
<i>TSP Launch Coach Training</i>	Training to become a TSP coach (5 days)	Offered only through the SEI; prerequisite is successful completion of <i>PSP Instructor Training</i>
TSP coach observation	Observation and mentoring of TSP coach during their first TSP launch (4 or 5 days)	Offered only through the SEI; successful completion necessary for SEI authorization

Figura A.7: Tabela de ações de formação [MW05]

Elementos do TSP

Anexo B

Relação das práticas do CMMI com elementos do TSP

As tabelas seguintes contêm a relação dos elementos do TSP com as práticas de cada área de processo do nível de maturidade 2 de CMMI, nomeadamente, Gestão de Requisitos (REQM), Planeamento de Projeto (PP), Acompanhamento e Controlo do Projeto (PMC), Gestão de Configurações (CM), Gestão de Acordo com o Fornecedor (SAM), Medição e Análise (MA) e Garantia da Qualidade de Processo e Produto (PPQA). Existem muitas outras áreas de processo em que os elementos do TSP se relacionam, mas estas são as áreas de processo referentes ao nível 2 de maturidade de um projeto com base no modelo do CMMI.

Antes de apresentar a relação do CMMI com TSP, é importante apresentar uma tabela (Tabela B.1) que mostra a legenda usada para classificar o tipo de mapeamento entre uma prática de CMMI e os elementos do TSP.

Relação das práticas do CMMI com elementos do TSP

Score Value	Description
D	Directly addresses; for TSP practices that meet the intent of the CMMI practice without any significant reservations (can be project or organizational practices)
P	Partially addresses; for project-oriented practices that TSP addresses, but with some significant weakness or omission
S	Supports; for organizational practices that TSP is not intended to fulfill completely, but which TSP supports by providing practices that either feed into the CMMI organization-level practice (e.g., data for a measurement repository) or that create a demand for or use the output of such a practice (e.g., tailoring criteria)
N	Not addressed; for project-related practices that TSP could and possibly should address but doesn't (i.e., a "gap")
U	Unrated; for organizational practices outside the scope of the TSP (e.g., GP 2.1 Establish an organizational policy)

Tabela B.1: Terminologia da classificação atribuída nos mapeamentos das figuras seguintes [MW05]

Relação das práticas do CMMI com elementos do TSP

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
SG1. Estimates of the project planning parameters are established and maintained.				
1.1. Establish a top-level work breakdown structure (WBS) to estimate the scope of the project.	<i>Scripts:</i> LAU3 <i>Forms:</i> STRAT, SUMS <i>Roles:</i> Design manager	The design manager leads the team in identifying the principal products and components of the project in LAU3 and records these on forms STRAT and SUMS.	D	
1.2. Establish and maintain estimates of the attributes of the work products and tasks.	<i>Scripts:</i> LAU3, LAU4, LAU5, LAU6 <i>Forms:</i> STRAT, SUMQ, SUMS <i>Roles:</i> Team leader, planning, design managers	Preliminary estimates are generated in LAU3 and refined as needed in LAU4 and LAU6. Form STRAT is used for developing the estimates in context; form SUMS records results. In LAU5, quality attributes (defect densities and phase yields) are estimated and recorded on form SUMQ. Different steps are led by the team leader, design manager, or planning manager.	D	

Figura B.1: Tabela 1 sobre a área de processo: Planejamento de Projetos (PP) [MW05]

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
1.3. Define the project life-cycle phases on which to scope the planning effort.	<i>Scripts:</i> LAU3, LAU4, LAU6 <i>Forms:</i> STRAT, TASK <i>Roles:</i> Team leader, process manager	The team leader leads the team in defining the project development strategy. The process manager leads the definition of the overall development process up to delivery; results are recorded on form STRAT and reflected in TASK plans generated in LAU4 and refined in LAU6.	D	
1.4. Estimate the project effort and cost for the work products and tasks based on estimation rationale.	<i>Scripts:</i> LAU3, LAU4, LAU6 <i>Forms:</i> STRAT, TASK, TSP workbooks <i>Roles:</i> Planning, design managers, team member	Preliminary estimates are made in LAU3 and refined as needed and to the necessary level of detail in LAU4 and LAU6. Forms STRAT and TASK record results. The design or planning manager leads the way in LAU3 and LAU4. Individual team members make adjustments based on personal historical data if available, otherwise on personal estimated productivity.	D	Dollar-based cost estimates are not explicitly called for; however, in practice, teams generate these if required by the organization.
SG2. A project plan is established and maintained as the basis for managing the project.				

Figura B.2: Tabela 2 sobre a área de processo: Planejamento de Projetos (PP) [MW05]

Relação das práticas do CMMI com elementos do TSP

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
2.1. Establish and maintain the project's budget and schedule.	<i>Scripts:</i> LAU4, LAU6 <i>Forms:</i> TASK, SCHEDULE, TSP workbooks <i>Roles:</i> Team leader, planning manager, team member	Script LAU4 develops an overall schedule, while script LAU6 develops detailed individual schedules for the entire team. These are captured first on the team's overall TASK and SCHED forms (LAU4) and then on each individual's TASK and SCHED form (LAU6), and the individual plans are rolled up in the TSP consolidated workbook (LAU6). The team leader or planning manager leads the discussions.	D	A budget is not specifically addressed in monetary terms; expenditures are generally expressed in terms of person-hours on task. In practice, teams generate a monetary budget if management asks for it.
2.2. Identify and analyze project risks.	<i>Scripts:</i> LAU7 <i>Forms:</i> ITL, team and individual TSP workbooks <i>Roles:</i> Team leader	Script LAU7 guides the team explicitly through identifying and making a preliminary analysis of project risks, capturing them on the issue tracking log (ITL), and filing same in the project NOTEBOOK. The team leader leads the discussion.	D	
2.3. Plan for the management of project data.	<i>Roles:</i> Planning, support managers <i>Other:</i> SUMMARY NOTEBOOK	The planning manager is responsible for maintaining the project NOTEBOOK that holds both launch and ongoing project process data. The SUMMARY specification details a periodic or event-driven rollup of project data.	P/S	Details of how data management is accomplished are not specified by the TSP. If there is no organizational standard in place, the planning or support manager usually sets up a computer-accessible version of the project NOTEBOOK and keeps weekly snapshots of same.

Figura B.3: Tabela 3 sobre a área de processo: Planejamento de Projetos (PP) [MW05]

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
2.4. Plan for necessary resources to perform the project.	<i>Scripts:</i> PREPL, PREPR, LAU (esp. LAU4, LAU6, LAU8, LAU9), REL <i>Forms:</i> TSP workbooks <i>Roles:</i> Team leader, team member	Identification of the team leader and project team prior to the launch represents management's initial thoughts on the necessary resources. The launch itself is the vehicle for the team to determine the necessary resources, develop alternative plans if necessary, and obtain management commitment to a particular plan with particular resources.	D	
2.5. Plan for knowledge and skills needed to perform the project.	<i>Scripts:</i> PREPL, PREPR LAU3, LAU4, LAU6, LAU7 <i>Forms:</i> Team and individual TSP workbooks <i>Roles:</i> Team leader, team member	Management is responsible for assigning a competent team leader and adequate staff to a project. The team leader has a specific responsibility to ensure that individuals on the team have the required knowledge and skills to perform their assigned tasks. Individual team members are responsible for arranging for the education and training necessary to do superior work.	D	

Figura B.4: Tabela 4 sobre a área de processo: Planejamento de Projetos (PP) [MW05]

Relação das práticas do CMMI com elementos do TSP

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
2.6. Plan the involvement of identified stakeholders.	<i>Scripts:</i> LAU, LAU1, LAU9, REL, STATUS <i>Forms:</i> TASK, LOGT, LOGD <i>Roles:</i> Team leader, role managers	Management is explicitly involved, beginning with the launch and continuing with regular STATUS reports, including the results of relaunches. The team leader and role managers are responsible for involving other stakeholders as necessary and appropriate.	P	There is no explicit guidance in the launch to plan for stakeholder involvement, nor is there a designated place to record such information. However, in practice, ensuring the involvement of relevant parties is a strength of TSP teams. Launches and relaunches are a common point of involvement for relevant stakeholders.
2.7. Establish and maintain the overall project plan content.	<i>Scripts:</i> LAU, REL <i>Forms:</i> WEEK <i>Roles:</i> Team leader, planning manager <i>Other:</i> NOTEBOOK	The entire launch sequence and subsequent relaunches create, update, and extend project plan artifacts for inclusion in the project NOTEBOOK. The plan is often revised during execution, both at the individual and team levels. For example, the weekly team meetings (script and form WEEK) result in frequent plan adjustments in response to the team's progress and understanding of the work.	D	See Notes above for SP 2.3.
SG3. Commitments to the plan are established and maintained.				
3.1. Review all plans that affect the project to understand project commitments.	<i>Scripts:</i> LAU6, LAU7, LAU8, LAU9 <i>Forms:</i> SUMQ, SUMP, SUMS, TASK, SCHED	The quality plan is reviewed in LAU6 after individual plans have been created and consolidated into a team plan. The team reviews its plan against the team's and management's desired goals in LAU6 and LAU8 and creates alternative plans if necessary. During LAU9, the team presents	D	If there are any ancillary plans that affect the team's plan, such as facilities issues or a support group, the team will either secure necessary commitments before they present to management or make the case for their needs during LAU3.

Figura B.5: Tabela 5 sobre a área de processo: Planejamento de Projetos (PP) [MW05]

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
	<i>Roles:</i> Team leader, team member	the plan and any alternatives and asks for management's approval of a specific plan. Management reviews the plan according to the plan assessment checklist.		
3.2. Reconcile the project plan to reflect available and estimated resources.	<i>Scripts:</i> LAU4, LAU6, LAU7, LAU8, LAU9 <i>Forms:</i> TASK, SCHED <i>Roles:</i> Team leader, team member	The TSP team compares and adjusts its plans frequently against existing and potential resources during the launch (LAU4, LAU6, LAU7, and LAU8). This includes preparation of alternative plans, where appropriate, that make different assumptions about available resources, critical milestone dates, and delivered functionality. In LAU9, management chooses a plan based, among other considerations, on resource availability.	D	
3.3. Obtain commitment from relevant stakeholders responsible for performing and supporting plan execution.	<i>Scripts:</i> LAU, REL <i>Forms:</i> TASK, LOGT, LOGD <i>Roles:</i> Team leader, role managers	The entire launch process elicits commitment by the project team for the team's plan (built in LAU2 to LAU8) to meet management's presented goals and by management to one of the plan alternatives presented by the team (LAU9). Relaunches revisit all commitments for feasibility, potential alternate approaches, and, if necessary, renegotiation with management.	P	TSP projects frequently invite significant stakeholders to participate in launches and typically check with external groups as necessary during the planning process; however, getting commitments from other "relevant stakeholders" is not explicitly called for. For TSP multi-teams, the component teams of a larger project explicitly negotiate commitments to support each other.

Figura B.6: Tabela 6 sobre a área de processo: Planejamento de Projetos (PP) [MW05]

Relação das práticas do CMMI com elementos do TSP

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
SG1. Actual performance and progress of the project are monitored against the project plan.				
1.1. Monitor the actual values of the project planning parameters against the project plan.	<i>Scripts:</i> WEEK, PM, REL1, STATUS <i>Forms:</i> WEEK, TSP workbooks <i>Roles:</i> Team leader, planning manager, other role managers	TSP teams typically examine actual values during the weekly status meeting, postmortems, and meeting 1 of relaunches. The team leader or planning manager leads the team in comparing these data to estimates (for productivity and time on task) and the actual work products (for size and defect density). Other role managers weigh in as appropriate.	D	

Figura B.7: Tabela 1 sobre a área de processo: Monitorização e Controlo de Projetos (PMC) [MW05]

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
1.2. Monitor commitments against those made in the project plan.	<i>Scripts:</i> WEEK, STATUS <i>Forms:</i> GOAL, WEEK, PM <i>Roles:</i> Team leader, team member, role managers <i>Other:</i> Quarterly review checklist	Team members, usually in association with their designated role manager responsibilities and as captured on form GOAL, monitor the team's status with respect to its goals and commitments weekly (WEEK).	D	
1.3. Monitor risks against those identified in the project plan.	<i>Scripts:</i> WEEK, STATUS <i>Forms:</i> IRTL <i>Roles:</i> Role managers	Team members, usually in conjunction with one or more role assignments as captured on form IRTL, monitor the status of identified plan risks weekly and report to the team weekly and management regularly.	D	The team assigns risk monitoring responsibilities in LAU7.
1.4. Monitor the management of project data against the project plan.	<i>Scripts:</i> WEEK <i>Forms:</i> WEEK, Team and individual TSP workbooks <i>Roles:</i> Planning manager	The planning manager is responsible for ensuring that individual program plans are updated weekly and revised as needed and for consolidating these data weekly into a team view.	P/S	Details of how data is managed on an ongoing basis are not specified by the TSP; however, this is clearly a planning manager responsibility.

Figura B.8: Tabela 2 sobre a área de processo: Monitorização e Controlo de Projetos (PMC) [MW05]

Relação das práticas do CMMI com elementos do TSP

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
1.5. Monitor stakeholder involvement against the project plan.	<i>Scripts:</i> PM, STATUS	In addition to the launch, the team obtains stakeholder evaluations during the postmortem.	P	As in PP SP 2.6 and 3.3, opportunities for stakeholder involvement are obvious and implicitly encouraged, but not explicitly called for outside of the PM activity.
	<i>Roles:</i> All role descriptions, esp. team leader	The team leader regularly provides status to management and other designated stakeholders. Role managers involve relevant stakeholders during execution of the plan, as required.		
1.6. Periodically review the project's progress, performance, and issues.	<i>Scripts:</i> WEEK	The team reviews its status weekly. The team leader reports team data and issues regularly, usually weekly, to management. Management holds quarterly reviews of project status.	D	
	<i>Forms:</i> WEEK, STATUS, SUMMARY			
	<i>Roles:</i> Team leader, planning manager, team member <i>Other:</i> Quarterly review checklist			
1.7. Review the accomplishments and results of the project at selected project milestones.	<i>Scripts:</i> PM, REL1	The team leader reviews current status with the team during REL1 and with management as requested. The team leader and several of the role managers lead a thorough review of the team's performance, processes, and other important aspects of the project during postmortems for each launch cycle and at the end of the project.	D	
	<i>Forms:</i> STATUS			
	<i>Roles:</i> Team leader, role managers			

Figura B.9: Tabela 3 sobre a área de processo: Monitorização e Controlo de Projetos (PMC) [MW05]

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
SG2. Corrective actions are managed to closure when the project's performance or results deviate significantly from the plan.				
2.1. Collect and analyze the issues and determine the corrective actions to address the issues.	<i>Scripts:</i> WEEK	Significant deviations from the plan, changes in the status of identified risks, and any other relevant issues are flagged during weekly status meetings. The effect on the achievement of team goals is of paramount concern. Team consensus is typically sought for corrective actions where the solution is not obvious from the data.	D	
	<i>Forms:</i> WEEK, associated weekly meeting minutes, IRTL, TSP workbooks			
	<i>Roles:</i> Team leader, role managers, team member			
2.2. Take corrective action on identified issues.	<i>Scripts:</i> WEEK	The team leader, a designated role manager, or other team members take corrective actions as necessary, usually as a result of decisions made at the weekly meeting (WEEK) and recorded as action items in the minutes.	D	
	<i>Forms:</i> WEEK and weekly meeting minutes, TASK, LOGT, LOGD			

Figura B.10: Tabela 4 sobre a área de processo: Monitorização e Controlo de Projetos (PMC) [MW05]

Relação das práticas do CMMI com elementos do TSP

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
	<i>Roles:</i> Team leader, role managers, team member			
2.3. Manage corrective actions to closure.	<i>Scripts:</i> WEEK, meeting minutes <i>Forms:</i> IRTL, TASK, LOGT, LOGD <i>Roles:</i> Team leader, role managers, team member	The team, led by the team leader or the appropriate role manager, monitors on at least a weekly basis how effective its corrective actions are and whether or not they need to be adjusted. Decisions are recorded either in meeting minutes or on the IRTL.	D	

Figura B.11: Tabela 5 sobre a área de processo: Monitorização e Controlo de Projetos (PMC) [MW05]

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
SG1. Preparation for validation is conducted.				
1.1. Select products and product components to be validated and the validation methods that will be used for each.	<i>Scripts:</i> LAU3, LAU4, LAU6, REQ, ANA, TEST, TEST1, TEST2, TEST3 <i>Forms:</i> TASK, LOGT, LOGD <i>Roles:</i> Customer interface, design, and test managers	System testing is specified early in the project (REQ, ANA) and refined (TEST, TEST3) throughout the project. Early build and integration tests (TEST1, TEST2) incorporate system tests as appropriate. The referenced role managers ensure that the appropriate products, components, and methods are selected and available. Specific tasks appear in individual TASK plans and are logged against in LOGT and LOGD.	D	

Figura B.12: Tabela 1 sobre a área de processo: Gestão de Requisitos (REQM) [MW05]

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
1.2-2. Establish and maintain the environment needed to support validation.	<i>Scripts:</i> LAU3, LAU4, LAU6 <i>Forms:</i> TASK, LOGT, LOGD <i>Roles:</i> Support, customer interface, design, and test managers	The support manager has the responsibility to ensure that the verification environment is adequate. LAU3 has a specific step for identifying needed items in a support plan. The other referenced role managers communicate their needs and the entire team ensures that their individual TASK plans support the validation effort.	P	No specific activities are called for in any of the scripts to actually "establish and maintain," but it is strongly implied by the support manager role description and the LAU3 activity.
1.3-3. Establish and maintain procedures and criteria for validation.	<i>Scripts:</i> LAU3, LAU4, LAU6, REQ, ANA, TEST3 <i>Forms:</i> TASK, LOGT, LOGD <i>Roles:</i> Customer interface, support, design, implementation, and test managers	A system test plan is developed during requirements development (REQ) or analysis (ANA), refined, and extended as necessary for system test (TEST3). The referenced role managers take responsibility as appropriate for the various aspects of validation.	D	Specific validation criteria are not called for in the scripts but are typically identified throughout the project as appropriate.
SG2. The product or product components are validated to ensure that they are suitable for use in their intended operating environment.				

Figura B.13: Tabela 2 sobre a área de processo: Gestão de Requisitos (REQM) [MW05]

Relação das práticas do CMMI com elementos do TSP

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
2.1. Perform validation on the selected products and product components.	<i>Scripts:</i> REQ, ANA, TEST1, TEST2, TEST3 <i>Forms:</i> TASK, LOGT, LOGD <i>Roles:</i> Team member, design, implementation, support, and test managers	TEST1 and TEST2 aim specifically at validating product components, while TEST3 specifies verifying proper operations under normal and abnormal operating conditions, presumably as specified previously in the customer's validated requirements (REQ, ANA). Each role manager ensures that appropriate activities have been included in individual TASK plans.	D	
2.2. Analyze the results of the validation activities and identify issues.	<i>Scripts:</i> TEST, TEST1, TEST2, TEST3, TESTD, PM <i>Forms:</i> TASK, LOGT, LOGD, SUMP, SUMQ <i>Roles:</i> Process, quality, and test managers	The referenced role managers are responsible for specific activities as indicated in the scripts. SUMP, SUMQ, and TASK summarize relevant data gathered in LOGT and LOGD. All test defects are attributed to source (product or test) and, if in the product, further analyzed using TESTD. Postmortems (PM) look specifically at quality and other issues related to validation efforts.	D	

Figura B.14: Tabela 3 sobre a área de processo: Gestão de Requisitos (REQM) [MW05]

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
SG1. Agreements with the suppliers are established and maintained.				
1.1. Determine the type of acquisition for each product or product component to be acquired.	<i>Scripts:</i> LAU1, LAU3, LAU4, LAU6, LAU7, LAU8, HLD <i>Forms:</i> TASK, LOGT, LOGD <i>Roles:</i> Team leader, design manager	Management directive (LAU1), the development strategy (LAU3), the need to develop alternative solutions (LAU4, LAU6, LAU8), or risk mitigation (LAU7) for the project may determine a need for an outside supplier. The design manager would be responsible for HLD activities to determine which products or components would be built by or obtained from a supplier. The team leader would be involved in decisions to determine acquisition type.	D	
1.2 Select suppliers based on an evaluation of their ability to meet the specified requirements and established criteria.	<i>Scripts:</i> LAU4, LAU6, LAU7 <i>Forms:</i> TASK, LOGT, LOGD	If a supplier is needed, tasks to determine viable suppliers would be placed in the team's project plan (LAU4, TASK, LAU6, and LAU7) and monitored (TASK, LOGT, LOGD). The team	D	

Figura B.15: Tabela 1 sobre a área de processo: Gestão de Acordo com o Fornecedor (SAM) [MW05]

Relação das práticas do CMMI com elementos do TSP

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
	<i>Roles:</i> Team leader, support manager	leader and design manager, at a minimum, would likely be involved in supplier selection.		
1.3 Establish and maintain formal agreements with the supplier.	<i>Forms:</i> TASK, LOGT, LOGD <i>Roles:</i> Team leader, support manager	If a supplier is needed, tasks to determine viable suppliers would be placed in the team's project plan (LAU4, TASK, LAU6, and LAU7) and monitored (TASK, LOGT, LOGD). The team leader or support manager is typically responsible for monitoring such activities for the team.	D	
SG2. Agreements with the suppliers are satisfied by both the project and the supplier.				
2.1. Review candidate COTS products to ensure they satisfy the specified requirements that are covered under the supplier agreement.	<i>Scripts:</i> HLD, IMP <i>Forms:</i> TASK, TSP workbooks <i>Roles:</i> Design and implementation managers	Individual TASK plans would likely include investigations of COTS products and would be tracked in individual workbooks. Tailored versions of HLD and/or IMP would reflect a customized design approach. The design or implementation managers would typically lead or coordinate such activities.	D	
2.2 Perform activities with the supplier as specified in the supplier agreement.	<i>Scripts:</i> WEEK, STATUS <i>Forms:</i> TASK, TSP workbooks	The supplier agreement activities would be reflected in one or more TASK plans and reflected in the corresponding TSP workbooks. Significant activities would be reported by one	D	

Figura B.16: Tabela 2 sobre a área de processo: Gestão de Acordo com o Fornecedor (SAM) [MW05]

<i>Specific Practice</i>	<i>TSP Reference</i>	<i>Observation</i>	<i>Rating</i>	<i>Notes</i>
	<i>Roles:</i> Team member, role managers	or more team members in the weekly meeting (WEEK), often in conjunction with one or more of the role manager functions. Problems and milestones would likely be reported during the weekly STATUS meeting.		
2.3 Ensure that the supplier agreement is satisfied before accepting the acquired product.	<i>Scripts:</i> WEEK, STATUS, IMP <i>Forms:</i> TASK, TSP workbooks <i>Roles:</i> Team leader, team member, role managers	Tasks for testing of supplier products would be reflected in individual TASK plans, probably during an implementation phase (IMP), and tracked in the TSP workbooks. Status of those tasks would be reviewed in the weekly team meeting (WEEK) and STATUS meeting. The various affected role managers would likely be involved. The team leader typically has final approval authority.	D	
2.4 Transition the acquired products from the supplier to the project.	<i>Scripts:</i> IMP, TEST1, TEST2 <i>Forms:</i> TASK, TSP workbooks <i>Roles:</i> Test and implementation managers	This would typically be an implementation (IMP) or build/integration (TEST1, TEST2) activity reflected in one or more TASK plans and tracked in the TSP workbook. The implementation or test manager would take responsibility for completion of these activities.	D	

Figura B.17: Tabela 3 sobre a área de processo: Gestão de Acordo com o Fornecedor (SAM) [MW05]

Anexo C

Relação das práticas do CMMI com práticas do Scrum

As tabelas seguintes representam um ponto de orientação para a implementação das práticas do CMMI através do Scrum, apenas referindo as áreas de processo principais (gestão de requisitos, planeamento de projetos e monitorização e controlo de projetos).

REQM	CMMI Practice	Scrum Practice
SP 1.1	Develop an understanding with the requirements providers on the meaning of the requirements.	<ul style="list-style-type: none">• Review of Product Backlog (requirements) with Product owner and team.
SP 1.2	Obtain commitment to the requirements from the project participants.	<ul style="list-style-type: none">• Release planning and Sprint planning sessions that seek team member commitment.
SP 1.3	Manage changes to the requirements as they evolve during the project.	<ul style="list-style-type: none">• Add requirements changes to the Product Backlog.• Manage changes in the next Sprint planning meeting.
SP 1.5	Identify inconsistencies between the project plans and work products and the requirements.	<ul style="list-style-type: none">• Daily standup meeting to identify issues.• Release planning and Sprint planning sessions to address inconsistencies.• Sprint burndown chart that tracks effort remaining.• Release burndown chart that tracks story points that have been completed. This shows how much of the product functionality is left to complete.

Figura C.1: Tabela de métodos para a Gestão de Requisitos do CMMI através do Scrum [PS09]

Relação das práticas do CMMI com práticas do Scrum

PP	CMMI Practice	Scrum Practice
SP 1.1	Establish a top-level work breakdown structure (WBS) to estimate the scope of the project.	<ul style="list-style-type: none"> The standard tasks used in a Scrum process combined with specific project tasks (Scrum Backlog).
SP 1.2	Establish and maintain estimates of the attributes of the work products and tasks.	<ul style="list-style-type: none"> Story points, used to estimate the difficulty (or relative size) of a Story (requirement).
SP 1.3	Define the project life-cycle phases upon which to scope the planning effort.	<ul style="list-style-type: none"> The Scrum process.
SP 1.4	Estimate the project effort and cost for the work products and tasks based on estimation rationale.	<ul style="list-style-type: none"> Scrum Ideal Time estimate (similar to billable hours or Full-time Equivalents).
SP 2.1	Establish and maintain the project's budget and schedule.	<ul style="list-style-type: none"> Scrum estimates (in Ideal Time). Estimates of what work will be in each release. Sprint Backlog. Project Taskboard.
SP 2.4	Plan for necessary resources to perform the project.	<ul style="list-style-type: none"> Scrum estimates in Ideal Time Release plan, Sprint Backlog and assignments.
SP 2.6	Plan the involvement of identified stakeholders.	<ul style="list-style-type: none"> Scrum process roles (including team, Scrum Master, Product Owner). [Note: The stakeholders listed in Scrum might not be the complete list of stakeholders for the project, e.g., customers, other impacted teams.]
SP 2.7	Establish and maintain the overall project plan content.	<ul style="list-style-type: none"> Scrum release plan. Sprint Backlog. Project Taskboard. [Note: The term "plan" in CMMI refers to additional plan components (such as risks and data management) that are not called out specifically in Scrum.]
SP 3.1	Review all plans that affect the project to understand project commitments.	<ul style="list-style-type: none"> Sprint planning meeting. Daily Scrum meeting.
SP 3.2	Reconcile the project plan to reflect	<ul style="list-style-type: none"> Sprint planning meeting.

Figura C.2: Tabela 1 de métodos para o Planejamento de Projetos do CMMI através do Scrum [PS09]

	available and estimated resources.	<ul style="list-style-type: none"> Daily Scrum meeting.
SP 3.3	Obtain commitment from relevant stakeholders responsible for performing and supporting plan execution.	<ul style="list-style-type: none"> Sprint planning meeting. Daily Scrum meeting. [Note: The stakeholders listed in Scrum might not be the complete list of stakeholders for the project.]

Figura C.3: Tabela 2 de métodos para o Planejamento de Projetos do CMMI através do Scrum [PS09]

Relação das práticas do CMMI com práticas do Scrum

PMC	CMMI Practice	Scrum Practice
SP 1.1	Monitor the actual values of the project planning parameters against the project plan.	<ul style="list-style-type: none"> • Sprint burndown chart that tracks effort remaining. • Release burndown chart that tracks completed story points. This shows how much of the product functionality is left to complete. • Project Task Board used to track stories (requirements) that are done, in progress, or ones that need verification.
SP 1.2	Monitor commitments against those identified in the project plan.	<ul style="list-style-type: none"> • Discussions on team commitments at the: <ul style="list-style-type: none"> – Daily Scrum meeting. – Sprint review meeting. • Sprint burndown chart that tracks effort remaining. • Release burndown chart that tracks story points that have been completed. This shows how much of the product functionality is left to complete.
SP 1.5	Monitor stakeholder involvement against the project plan.	<ul style="list-style-type: none"> • Discussions at the: <ul style="list-style-type: none"> – Daily Scrum meeting. – Sprint review meeting. • [Note: The stakeholders listed in Scrum might not be the complete list of stakeholders for the project, e.g., customers, other impacted teams.]
SP 1.6	Periodically review the project's progress, performance, and issues.	<ul style="list-style-type: none"> • Daily Scrum meeting. • Sprint review meeting. • Retrospectives.
SP 1.7	Review the accomplishments and results of the project at selected project milestones.	<ul style="list-style-type: none"> • Sprint review meeting.
SP 2.1	Collect and analyze the issues and determine the corrective actions necessary to address the issues.	<ul style="list-style-type: none"> • Notes from the: <ul style="list-style-type: none"> – Daily Scrum meeting. – Sprint review meeting. • [Note: Some teams track their outstanding actions on the Product Backlog. It doesn't matter where or how the items are tracked, as long as they are.]
SP 2.2	Take corrective action on identified issues.	<ul style="list-style-type: none"> • Actions from the: <ul style="list-style-type: none"> – Daily Scrum meeting. – Sprint review meeting.
SP 2.3	Manage corrective actions to closure.	<ul style="list-style-type: none"> • Tracking of actions from: <ul style="list-style-type: none"> – Daily Scrum meeting. – Sprint review meeting. • [Note: This assumes that teams will track (and not lose) actions.]

Figura C.4: Tabela de métodos para a Monitorização e Controlo de Projetos do CMMI através do Scrum [PS09]

Relação das práticas do CMMI com práticas do Scrum

Referências

- [CKS11] Mary Beth Chrissis, Mike Konrad e Sandy Shrum. *CMMI: Guidelines for Process Integration and Product Improvement*. Addison-Wesley, 2011.
- [Cla12] Clarizen: Features. <http://www.clarizen.com/project-software/features.html>, consulted on January 2012.
- [CM12] Timothy A. Chick e Gene Miluk. *Getting the performance you need from processes that work: The CMMI Accelerated Improvement Method*. SEI-CMU, 2012.
- [Coh08] Mike Cohn. *Improving On Traditional Release Burndown Charts*. Mountain Goat Software, 2008.
- [Dal09] Jeff Dalton. *Broadsword - Process Innovation at the speed of life*. 2009.
- [DavPAa] Noopur Davis. *Using Scrum in a TSP Measurement Framework*. presented at the 2008 TSP Symposium Pittsburgh, PA.
- [DavPAb] Noopur Davis. *Lessons Learned Using Agile Practices with TSP*. presented at the 2010 TSP Symposium Pittsburgh, PA.
- [DM03] Noopur Davis e Julia Mullaney. *The Team Software Process (TSP) in Practice: A Summary of Recent Results*. 2003.
- [EVi12] Project management systems - Earned value management - part 4. http://www.project-management-basics.com/project_management_173_Project_management_systems_part_10_Earned_value_management_d.shtml, consulted on June 2012.
- [Gre12] GreenHopper. <http://www.atlassian.com/software/greenhopper/overview>, consulted on January 2012.
- [Gy11] Hu Guang-yong. *Study and practice of import Scrum agile software development*. IEEE, 2011.
- [Jir12a] JIRA: features. <http://www.atlassian.com/software/jira/features>, consulted on January 2012.
- [Jir12b] JIRA: overview. <http://www.atlassian.com/software/jira/overview>, consulted on January 2012.
- [MCM10] James Mchale, Tim Chick e Gene Miluk. *Implementation Guidance for the Accelerated Improvement Method*. 2010.

REFERÊNCIAS

- [MCNM07] James Mchale, Tim Chick, Davis Noopur e Gene Miluk. *Accelerating CMMI Adoption with PSP/TSP*. 2007.
- [Mel09a] SEI Partner Network | Carnegie Mellon. *TSP Product Suite: Materials and Use Restrictions*. 2009.
- [Mel09b] SEI Partner Network | Carnegie Mellon. *TSP Product Suite: Procedures*. 2009.
- [Mul12] Multicert. <http://www.multicert.com>, consulted on June 2012.
- [MW05] James McHale e Daniel Wall. *Mapping TSP to CMMI*. 2005.
- [Ove10] James Over. *Introduction to the Team Software Process*. 2010.
- [Pad08] Alan Padula. *TSP-Agile Showdown*. 2008.
- [Pro12a] The Software Process Dashboard Initiative. <http://www.processdash.com/>, consulted on February 2012.
- [Pro12b] The Software Process Dashboard Initiative: Functionality. <http://www.processdash.com/functionality>, consulted on February 2012.
- [PS09] Neil Potter e Mary Sakry. *Implementing Scrum (AGILE) and CMMI Together*. 2009.
- [QRE12] QREN. <http://www.qren.pt>, consulted on June 2012.
- [Rai06] J.B. Rainsberger. *Augmenting the Agile Planning Toolbox*. 2006.
- [Red12a] Redmine. <http://www.redmine.org/>, consulted on May 2012.
- [Red12b] Redmine: Main features. <http://www.redmine.org/projects/redmine/wiki/Features>, consulted on May 2012.
- [Roy08] Daniel Roy. *Agile, TSP, CMMI pick one, pick two, pick all three!* 2008.
- [Sch04] Ken Schwaber. *Agile project management with Scrum*, volume 7 of *Microsoft Professional*. Microsoft Press, 2004.
- [Sch07] Ken Schwaber. *What is Scrum?* 2007.
- [Sch10] David Scherb. *Accelerated Improvement Method (AIM)*. 2010.
- [Smi04] Karen J Smiley. *Agility and The Team Software Process (TSP)*. 2004.
- [Sol02] Paul Solomon. *Using CMMI to Improve Earned Value Management*. 2002.
- [SS11] Ken Schwaber e Jeff Sutherland. *The Definitive Guide to Scrum: The Rules of the Game*. 2011.
- [Str12] Strongstep. <http://www.strongstep.pt>, consulted on June 2012.
- [Tfs12] Visual Studio Team Foundation Server 2010. <http://www.microsoft.com/visualstudio/en-us/products/2010-editions/team-foundation-server/overview>, consulted on February 2012.
- [Tur11] Jennifer Turgeon. *SCRUMP (Scrum + RUP) and CMMI: The Story of a Harmonious Process and Product Deployment*. 2011.
- [Wes05] Cynthia West. *Using Earned Value Management for Improving Processes*. 2005.