

Utilização da Ferramenta Redmine para Auxiliar na Obtenção da Certificação MPS-BR Nível F

Thiago Detomi¹, Paulo Afonso Júnior², Heitor Costa²

¹Laboratório de Projetos e Pesquisas - Universidade Federal de Lavras - MG - Brazil

²Departamento de Ciência da Computação - Universidade Federal de Lavras - MG - Brazil

thiago.detomi@lemaf.ufla.br, {pauloa.junior, heitor}@dcc.ufla.br

Resumo. Neste artigo, é relatada a experiência na utilização de uma ferramenta de controle de projetos na implantação dos processos do nível F do MPS-BR em uma empresa desenvolvedora de software. Para a escolha dessa ferramenta, alguns critérios foram definidos para avaliar as ferramentas existentes, os quais devem ser atendidos para ser considerada na implantação dos processos. Após essa avaliação, a ferramenta Redmine foi selecionada e utilizada pela gestão de projetos e pelas equipes de desenvolvimento de forma a auxiliar no processo de implantação dos processos e da obtenção da certificação. A seleção foi mediante a Redmine passar por uma análise crítica de viabilidade para entender suas características e suas limitações dentro do contexto da organização. Os resultados evidenciam os pontos fortes e limitações da Redmine quanto ao processo de certificação MPS-BR nível F neste cenário.

Abstract. In this paper, we reported the experience of using a project control tool in the implementation of MPS-BR level F processes in a software development company. To choose this tool, we defined some criteria to evaluate the existing tools, which must be considered in the implementation of the processes. After this evaluation, Redmine tool was selected and used by project management and development teams to assist in the process implementation and certification process. The selection was through Redmine undergo a critical analysis of feasibility to understand its characteristics and its limitations within the context of the organization. The results highlights Redmine's strengths and limitations regarding the MPS-BR level F certification process in this scenario.

1. Introdução

As mudanças estão cada vez mais frequentes nos ambientes corporativos, motivando as empresas a melhorarem seus processos de forma a maximizar sua capacidade produtiva. Dessa forma, as certificações mostram-se cada vez mais presentes nas empresas, tanto pelos benefícios esperados quanto pelo desafio de melhorar seus processos em diversos aspectos, especialmente no setor de Tecnologia da Informação, na qual certificações são cada vez mais necessárias. Dentro das diversas certificações de melhoria de processos possíveis para empresas desse setor, destacam-se os modelos de maturidade.

Os modelos de maturidade oferecem informações estratégicas para avaliar o desempenho dos processos organizacionais como uma forma de auxiliar a gestão dos projetos na empresa. Os modelos de maturidade em gestão de projetos servem como uma estrutura para comparação entre práticas atuais exercidas por uma organização e o que são consideradas como melhores práticas de gestão de projetos pelo mercado [Ibbs; Kwak, 2002]. Os modelos e maturidade em gestão de projetos consistem em um arquétipo de crescimento que estabelece estágios pré-definidos, permitindo autoavaliações e aperfeiçoamentos [Prado, 2008]. São exemplos de modelos de maturidade *Capability Maturity Model Integration* (CMMI), *Project Management Maturity Model* (PMMM), *Organizational Project Management Maturity Model* (OPM3) e Melhoria de Processo de Software Brasileiro (MPS-BR), abordado neste trabalho.

MPS-BR é um programa criado pela SOFTEX com o apoio do Ministério da Ciência e Tecnologia, Inovações e Comunicações (MCTIC) iniciado em 2003, cujo objetivo é melhorar a capacidade de desenvolvimento de software, serviços e práticas de gestão na indústria de Tecnologia da Informação (SOFTEX, 2016). O modelo MPS-BR está em conformidade com as normas internacionais ISO/IEC 122007 e ISO/IEC 330xx e o modelo CMMI, adequando-se à realidade das empresas brasileiras [Koscianski; Soares, 2007]. Esse programa baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de software e serviços correlatos e para a melhoria da qualidade e produtividade dos serviços prestados. Esse programa possui quatro componentes: i) Modelo de Referência MPS para Software (MR-MPS-SW); ii) Modelo de Referência MPS para Serviços (MR-MPS-SV); iii) Método de Avaliação (MA-MPS); e iv) Modelo de Negócio para Melhoria de Processo de Software e Serviços [SOFTEX, 2016].

No processo de implantação dos processos de MPS-BR em uma empresa de software, sobretudo em uma empresa que utiliza métodos ágeis no desenvolvimento, é comum a utilização de ferramentas para auxiliar no processo de gerenciamento dos projetos. Neste trabalho, o objetivo é apresentar uma análise crítica da ferramenta Redmine¹ durante o processo de certificação MPS-BR Nível F na empresa GT4W² (*Geotechnology for Web*), de forma a avaliar a sua efetividade durante e após o processo.

O restante do artigo está organizado da seguinte forma. A necessidade de realizar a adequação na empresa por utilizar Scrum e desejar a implantação de processos de qualidade do MPS-BR no desenvolvimento de sistemas de software é apresentada na Seção 2. A escolha de uma ferramenta computacional (apoio computacional) para apoiar o controle de projetos é destacada na Seção 3. A utilização da ferramenta escolhida - Redmine - é mostrada na Seção 4. Resultados obtidos com a implantação são discutidos na Seção 5. Resumo de alguns trabalhos relacionados é apresentado na Seção 6. Considerações finais são apresentados na Seção 7.

2. MPS-BR Nível F em Conjunto com o Scrum na GT4W

A GT4W é uma empresa de Tecnologia da Informação sediada em Lavras/MG, mas teve seu primeiro contato com o processo de implantação do MPS-BR em Belo Horizonte/MG. A empresa tem como principal produto projetos de desenvolvimento de sistemas de software para o setor público. A empresa é parceira do Laboratório de Estudos e Projetos

¹ <http://www.redmine.org>

² <http://www.gt4w.com.br/>

em Manejo Florestal (LEMAF) da Universidade Federal de Lavras (UFLA) em projetos para controle e monitoramento ambiental em âmbito federal e estadual, que utilizam como base o Georreferenciamento e a metodologia ágil Scrum para controle de seus projetos, possuindo equipes alocadas na UFLA. Scrum é uma metodologia ágil utilizada para gerenciar o desenvolvimento de produtos complexos criado na década de 90 [Schwaber; Sutherland, 2016], sendo um *framework* no qual as pessoas podem empregar diversas técnicas e processos.

O Scrum possui as seguintes práticas e ritos: i) *Product Backlog*; ii) *Sprint Planning*; iii) *Sprint*; iv) *Daily Stand-up Meeting*; iv) *Sprint Review*; e v) *Sprint Retrospective*. Além disso, possui três papéis principais: i) *Product Owner*; ii) *Scrum Master*; e iii) Time de desenvolvedores. O *framework* (Figura 1) tem início com a definição do *Product Backlog*, que contém a lista priorizada e refinada dos requisitos do produto, sendo de responsabilidade do *Product Owner* o seu gerenciamento. Definido o *Product Backlog*, o *Product Owner* e o Time de desenvolvedores, em uma reunião chamada *Sprint Planning*, definem e estimam o que pode ser realizado em uma iteração com um tempo definido, chamada *Sprint*.

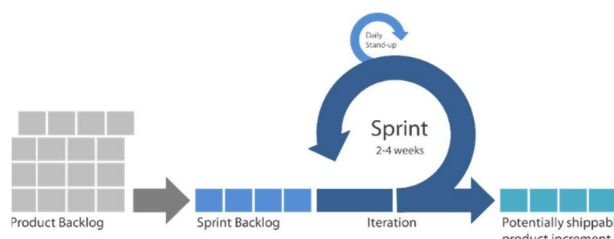


Figura 1 - Scrum Tradicional

Durante o progresso da *Sprint*, é realizada uma reunião diária entre o Time de desenvolvedores, chamada *Daily Stand-up Meeting*, na qual cada desenvolvedor reporta em que está trabalhando e se há algum impedimento que precisa ser tratado. Ao final da iteração, que normalmente leva de 2 a 4 semanas, é realizada a *Sprint Review*, na qual o Time de desenvolvedores apresenta o resultado da *Sprint* para o *Product Owner* que informa se o resultado foi ou não favorável.

Ao final do processo, é feita a *Sprint Retrospective*, uma reunião reflexiva entre os desenvolvedores para analisar os pontos positivos e o que pode ser melhorado nas próximas iterações. O processo retorna para a *Sprint Planning*, quando uma nova iteração é planejada, até o produto estar funcional. O *Scrum Master* é responsável por garantir que esse processo ocorra e remover qualquer impedimento de um membro do Time de desenvolvedores.

Para acompanhar o trabalho sendo desenvolvido na iteração (chamado na literatura de *Work in Progress - WIP*), é utilizada uma ferramenta chamada *kanban* que exhibe a lista das atividades necessárias, apresentadas em um quadro, para alcançar o objetivo da *Sprint*. Esse quadro é dividido em colunas, por exemplo, listando as atividades a fazer, as atividades sendo trabalhadas no momento e as atividades finalizadas (podendo ser personalizado de acordo com a cultura de cada equipe, colocando mais colunas). O *kanban* é atualizado em tempo real, à medida que o Time de desenvolvedores realiza as atividades, movendo as atividades de colunas.

A GT4W emprega uma utilização customizada do Scrum em seus projetos e seus resultados mostraram-se satisfatórios no decorrer do tempo. Porém, a necessidade do mercado, aliada ao crescimento da empresa, convenceu a gestão executiva a buscar a certificação MPS-BR para melhorar seus processos e ser um diferencial competitivo para a empresa, uma vez que grande parte de seus projetos são originados de licitações provenientes do setor público.

A customização do Scrum na GT4W possui algumas particularidades. Antes de iniciar o projeto, o Gerente de Projetos em conjunto com o *Product Owner* elabora o Documento de Visão do projeto, cujo objetivo é nortear a gestão quanto aos objetivos e ao escopo do projeto a ser desenvolvido, fornecendo requisitos macros que o projeto deve atender, do ponto de vista funcional e não-funcional. De posse do Documento de Visão, o *Product Owner* refina os requisitos macro em forma de histórias de usuários para serem executadas nas *sprints*. Havia também um papel adicional na equipe, chamado Líder Técnico, sendo um desenvolvedor mais experiente que orienta os menos experientes e lida com tarefas mais complexas. Outro ponto importante a destacar é o fato de o *Scrum Master* fazer parte do Time de desenvolvedores, ao contrário do Scrum tradicional, cuja figura do *Scrum Master* fica alocada integralmente em remover impedimentos e garantir os ritos do *framework*.

Uma instituição implementadora foi contratada para acompanhar o andamento do processo e constatou que a empresa possuía certa organização e padronização em seus processos, sendo aconselhada a buscar o Nível F, implantando os processos referentes aos níveis G e F do MPS-BR. O nível G (Parcialmente Gerenciado) possui dois processos: i) Gerência de Requisitos (GRE); e ii) Gerência de Projetos (GPR). O nível F (Gerenciado) possui cinco processos: i) Aquisição (AQU); ii) Gerência da Configuração (GCO); iii) Garantia da Qualidade (GQA); iv) Gerência de Portfolio de Projetos (GPP); e v) Medição (MED). O processo de Aquisição não foi considerado no escopo da avaliação, pois a GT4W se enquadra como uma fábrica de software.

3. Seleção da Redmine para Apoiar o Controle de Projetos

Com o advento da necessidade de implantar o MPS-BR na empresa e a definição do nível F como objetivo primário, foi recomendada pelo consultor responsável da instituição implementadora, a utilização de um sistema computacional (apoio computacional) para ajudar a controlar os projetos para atendimento a algumas exigências da implantação, por exemplo, a rastreabilidade bidirecional entre requisitos e atividades. O controle era realizado utilizando planilhas personalizadas para cada equipe sem padronização. Diante dessa necessidade, a comissão formada pelos gestores responsáveis e a instituição implementadora estabeleceram, baseados na experiência prévia da instituição implementadora em certificações anteriores, um conjunto de nove critérios categorizados de acordo com a necessidade para a implantação da certificação que a ferramenta deveria atender, sendo eles:

- **Integração com o Git:** a empresa utiliza o Git para controle de versões e de *releases* dos sistemas desenvolvidos, sendo fundamental a integração com os repositórios de forma a ser possível acessar o código fonte gerado no projeto;
- **Open source:** a ferramenta deve possuir código aberto para futuras customizações pela equipe de desenvolvimento;
- **Rastreabilidade bidirecional de requisitos:** permitir identificar cada item desenvolvido presente no Documento de Visão do projeto (mais alto nível de

abstração), as histórias de usuários do *Product Backlog* associadas ao item de visão, as tarefas associadas às histórias de usuários e os *commits* no repositório e vice-versa, possibilitando navegar entre os diversos níveis de atividades, fornecendo ampla visibilidade do projeto;

- **Possibilidade de trabalhar com Scrum:** Scrum é utilizado pelas equipes de desenvolvimento;
- **Histórico de atualização de tarefas:** manter um histórico que permitisse visualizar quem alterou o estado da atividade e quando foi realizada a alteração;
- **Controle da equipe:** possibilidade de configurar a equipe, gerenciar membros de projetos e permissões de acesso, de forma a limitar o acesso a determinadas áreas do sistema para usuários;
- **Autenticação de usuários:** *login* utilizando uma credencial para acesso ao sistema;
- **Gratuita:** GT4W é uma microempresa, não tendo recursos para investir em uma ferramenta proprietária;
- **Self-hosted:** a ferramenta deveria ser implantada nos servidores locais da empresa, não dependendo de serviços em nuvem para seu funcionamento. Essa não dependência é explícita em alguns contratos com clientes.

Esses critérios foram categorizados de acordo com seu grau de relevância para a certificação (Tabela 1), sendo: i) 1: pouco relevante; ii) 2: relevante; e iii) 3: muito relevante. O grau de relevância “pouco relevante” é considerado pouco importante, não inviabilizando a utilização da ferramenta, caso não seja atendido. O grau de relevância “relevante” é de relevância média, considerado importante, mas não chega a inviabilizar o uso da ferramenta, caso não seja atendido. O grau de relevância “muito relevante” é considerado muito importante, que inviabiliza o uso da ferramenta, caso não seja atendido. Assim, os critérios *Integração com o Git*, *Rastreabilidade bidirecional de requisitos*, *Possibilidade de trabalhar com Scrum*, *Gratuita* e *Self-hosted* são considerados **muito relevantes**. Por outro lado, o critério *Open source* e *Histórico de atualização de tarefas* são considerados **pouco relevantes**. Os dois critérios restantes, *Controle da equipe* e *Autenticação de usuários*, são considerados **relevantes**.

Tabela 1. Critérios e Grau de Relevância para a Seleção da Ferramenta de Controle de Projetos

ID	Critério	Grau de Relevância
1	Integração com o <i>Git</i>	3
2	<i>Open source</i>	1
3	Rastreabilidade bidirecional de requisitos	3
4	Possibilidade de trabalhar com <i>Scrum</i>	3
5	Histórico de atualização de tarefas	1
6	Controle da equipe	2
7	Autenticação de usuários	2
8	Gratuita	3
9	<i>Self-hosted</i>	3

Uma vez definida a lista de critérios, foi feita uma pesquisa para identificar possíveis ferramentas computacionais (apoio computacional) candidatas a serem

utilizadas para apoiar no processo de certificação baseada nessa lista. As ferramentas encontradas foram:

- **JIRA** é amplamente conhecida na gestão ágil de projetos, mas foi descartada pelo fato de não atender o critério 8 (*Gratuita*) e inclusive possui alto custo para implantação e customização;
- **Open Project** é uma ferramenta para controle de projetos que possui uma versão gratuita, mas foi descartada pelos envolvidos não possuírem contato prévio com ela, ocasionando na necessidade de esforço (tempo) para estudo do seu comportamento prático. Dessa forma, a sua utilização tornou-se inviável por conta do curto tempo entre a implantação da ferramenta nos projetos e a avaliação final da certificação;
- **Trello** consiste em uma ferramenta *online* para criação e controle de *kanbans*. Foram realizados testes práticos com essa ferramenta anteriormente, porém a *Rastreabilidade bidirecional de requisitos* (critério 3) não se mostrou ineficiente, apesar dos *plug-ins* existentes para a ferramenta. Além disso, Trello não é *Self-hosted* (critério 9), sendo descartada;
- **Redmine** é uma aplicação web para gerenciamento de projetos escrita utilizando o *framework* Ruby on Rails. Com as pesquisas realizadas na Internet e em fóruns de discussão, essa ferramenta apresentou bons resultados baseados nos critérios e nos graus de relevância definidos. O principal ponto que contribuiu para a sua escolha foi a indicação da instituição implementadora que acompanhava a implantação dos processos, pois o consultor responsável havia tido contato com a ferramenta em processos de certificação anteriores, atingindo seu o objetivo.

Selecionada a ferramenta, três equipes de desenvolvimento foram selecionadas para executarem um piloto com a Redmine, cujo objetivo era analisá-la na prática, consistindo inicialmente em utilizá-la em conjunto com as planilhas que faziam parte da cultura da empresa, a fim de evitar possíveis perdas de informações e mitigar o impacto das mudanças.

4. Utilização da Redmine nos Projetos

Para a certificação MPS-BR Nível F, a empresa precisava apresentar quatro projetos candidatos à avaliação, dois finalizados e dois em andamento. Os projetos seriam avaliados quanto ao escopo da avaliação do Nível F: i) Gerência de Projetos; ii) Gerência de Requisitos; iii) Gerência da Configuração; iv) Garantia da Qualidade; v) Gerência de Portfolio de Projetos; e vi) Medição. A Redmine foi selecionada para auxiliar na Gerência de Projetos e na Gerência de Requisitos.

Durante o período da certificação, quatro projetos haviam sido finalizados seguindo as recomendações presentes no MPS-BR e três projetos estavam sendo iniciados por três equipes diferentes alocadas no LEMAF/UFLA, em Lavras, Minas Gerais. Os projetos estariam finalizados no período da avaliação final, sendo selecionados como projetos candidatos aqueles com os melhores resultados baseado em análises pela equipe de implantação em conjunto com a instituição implementadora contratada. Para os projetos iniciantes, foi proposta a utilização da Redmine para fins de experiência e de *feedback*, de forma a avaliar a sua efetividade no processo estabelecido.

Uma vez definidas as equipes e os projetos que participariam da experiência com a Redmine, foram realizadas reuniões com a instituição implementadora para definir como seria a estrutura da Redmine. Nessas reuniões, foram discutidos diversos aspectos

de como a ferramenta seria utilizada para atender aos requisitos do MPS-BR. Dentre os vários pontos, os mais importantes foram a estruturação dos requisitos de projeto, a rastreabilidade bidirecional de requisitos e o acesso ao repositório Git.

Inicialmente, para adaptar-se ao contexto da organização, foi necessário inserir uma extensão na Redmine que permitia trabalhar com metodologias ágeis. Essa extensão permitia a utilização do *Kanban* e as demais características do Scrum, como a criação de *Product Backlog* e *Sprints*. A Redmine precisou ser configurada quanto ao tipo das tarefas, de forma a adequar-se ao processo, podendo adicionar tarefas dos seguintes tipos ao *Product Backlog* dos projetos:

- **Itens de visão:** tipo de tarefa que faz referência aos itens macro destacados no documento de visão do projeto;
- **Histórias de usuário:** tipo de tarefa que reflete uma função do sistema. Um conjunto de histórias de usuários são vinculadas a um item de visão;
- **Atividade:** tipo de tarefa que detalha o que será necessário para construir uma história de usuário. Um conjunto de tarefas é vinculado a uma história de usuário;
- **Bugs:** tipo de tarefa que consiste em problemas encontrados em determinada história de usuário durante o andamento da *Sprint*, relatado pelo *tester*.

Após os tipos de tarefas do projeto terem sido configurados, foi necessário configurar os papéis dos usuários do sistema. A Redmine contém os papéis padrão do Scrum, tais como, *Scrum Master*, *Product Owner* e *Time*. Mas, para adequação ao processo definido, foi necessário incluir os papéis “Líder Técnico” e “Analistas de Qualidade”, atribuídos aos respectivos responsáveis pelas funções. As *Sprints* foram configuradas com 10 dias úteis de desenvolvimento (tempo médio da iteração, empregado pela organização).

Em seguida, os projetos em andamento deveriam ser cadastrados pelo respectivo *Product Owner* de cada equipe selecionada, cabendo a ele a tarefa de cadastrar sua equipe e o *Backlog* do projeto. A estrutura da configuração dos projetos na Redmine é apresentada na Figura 2. Os repositórios Git eram adicionados ao projeto na Redmine pela equipe de infraestrutura da empresa. Como a Redmine não permite criar uma referência para o repositório Git, sua atualização deveria ser realizada com um *job* implementado pela própria equipe de infraestrutura, que atualizava diariamente os repositórios dos projetos, excluindo as versões anteriores e armazenando apenas a versão mais atual do repositório do projeto.

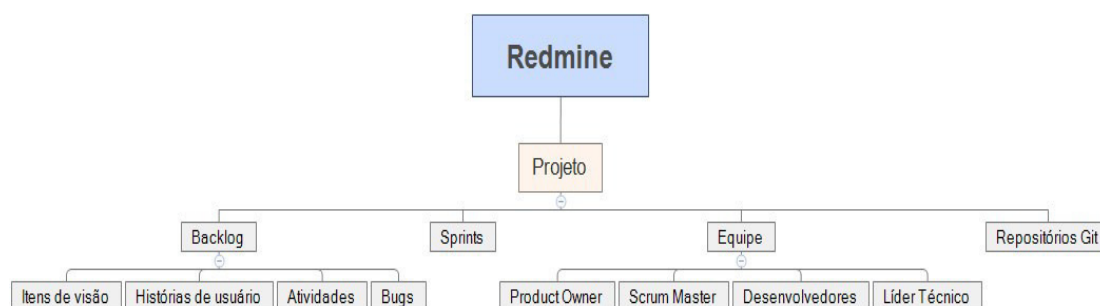


Figura 2 - Estrutura dos Projetos na Redmine

As equipes envolvidas foram treinadas pela equipe de qualidade e pelo Gerente de Projetos antes de iniciar de fato a utilização da Redmine. O treinamento consistiu em uma apresentação da ferramenta juntamente com sua política de uso e os padrões de nomenclaturas e de projeto. Ainda nesse treinamento, os desenvolvedores foram orientados a identificar a atividade sendo trabalhada utilizando comentários nos *commits* realizados ao finalizar determinada atividade. Para fins de identificação de tarefas no Git, a Redmine exige informação do caractere “#” seguido do código identificador (id) da atividade trabalhada no comentário do *commit*, estabelecendo um vínculo entre a tarefa e o *commit* no repositório. Dessa forma, o critério *Rastreabilidade bidirecional de requisitos* (Seção 3) é atendido.

Após esse treinamento, as equipes foram capazes de iniciar a utilização da ferramenta Redmine em seus projetos. Foi aconselhado manter o uso das planilhas em paralelo a essa ferramenta no primeiro momento, de forma que o impacto nas equipes fosse mitigado, visto que as planilhas eram utilizadas desde os primeiros projetos da empresa, sendo a primeira vez que uma ferramenta estava sendo implantada para controle dos projetos. O fluxo realizado para a implantação da Redmine para ser utilizada pelas equipes é apresentado na Figura 3.

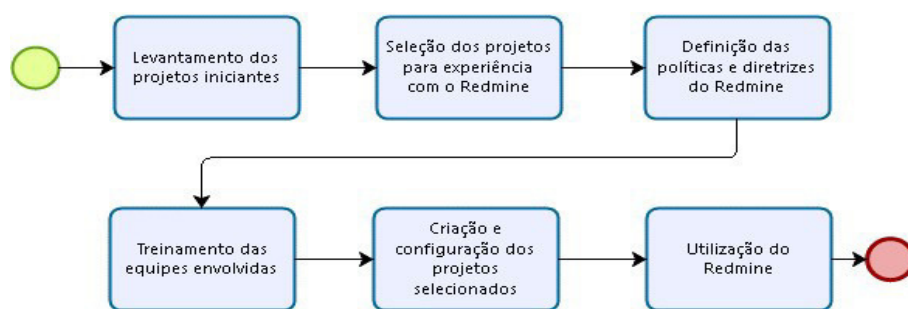


Figura 3 - Passos para Implantação do Redmine

Em um primeiro momento, houve certa resistência quanto ao uso da Redmine pelos desenvolvedores, com a justificativa de que ela era ineficiente em alguns aspectos, como a impossibilidade de vincular mais de uma pessoa a uma tarefa (característica comum no Time de desenvolvedores quando se desenvolve em par). Outro ponto levantado pelo Time de desenvolvedores está relacionado com a estimativa das histórias de usuários, que a ferramenta exigia uma estimativa em horas de desenvolvimento, algo que não condizia com a realidade do Time de desenvolvedores, pois utilizava a escala de Fibonacci para estimar as histórias de usuários que entram na *Sprint*. Essa é uma característica comum no Scrum, empregada por grande parte das equipes da empresa. Para relatar casos em que a ferramenta não se adaptava ao processo de desenvolvimento da equipe, foi solicitado às equipes envolvidas que relatassem (via *e-mail*) aos responsáveis pela certificação para que alternativas fossem adotadas. Com a realização das *Sprints* posteriores, a resistência foi menor.

Ao longo do processo de certificação, cada projeto teve a duração média de três iterações. Alguns projetos foram finalizados após a instituição avaliadora auditar os projetos, sendo considerados (auditados) como projetos em andamento.

5. Resultados Obtidos

Durante a utilização da Redmine pelas equipes envolvidas no processo de implantação da qualidade, foi possível ter uma visão mais ampla dos projetos em execução, algo que se tornava ineficiente quando as planilhas eram a principal forma de controle utilizada. O desempenho dessa ferramenta foi medido com base nos critérios descritos na Seção 3, os quais foram analisados quanto à sua efetividade no processo, atendendo, não atendendo ou atendendo parcialmente o critério em questão.

O critério 1 (*Integração com o Git*) foi considerado de alto grau de relevância para a obtenção da certificação, o qual a Redmine atendeu parcialmente, de forma a conseguir ser efetiva para a instituição avaliadora quanto ao acesso ao repositório, mas, do ponto de vista de infraestrutura, mostrou-se tecnicamente inviável. Isso ocorreu, pois a Redmine exige que os repositórios Git sejam clonados e inseridos em seu diretório para serem acessados, sendo necessário ter um espaço cada vez maior no *storage*, à medida que novos projetos são criados. Além disso, foi necessário criar uma rotina de atualização para realizar a cópia do repositório diariamente na Redmine, dependendo constantemente da intervenção da equipe de infraestrutura para dar suporte à ferramenta. Isso tornava um procedimento de alto custo para a empresa, que conta apenas com dois analistas de infraestrutura e uma alta quantidade de projetos, além do risco de informações sensíveis se perderem por causa do espaço limitado nos *storages* disponíveis. O modelo ideal seria a ferramenta ter uma referência para os repositórios dos projetos, de forma a não ser necessário criar cópias adicionais ou rotinas para substituir os repositórios por versões mais atualizadas.

O critério 2 (*Open source*), de baixa relevância, está relacionado à licença da ferramenta, que preferencialmente deveria ser de código aberto, para que melhorias futuras pudessem ser propostas pelas equipes de desenvolvimento e pela equipe de infraestrutura. A Redmine atendeu esse critério por ser de código aberto, escrita utilizando o *framework* Ruby on Rails publicada sob a licença GPL³. É importante frisar que, no período da certificação, não havia desenvolvedores na empresa com conhecimento pleno em Ruby on Rails, mas poderia ser cogitada a contratação de alguém com o perfil para melhorar a ferramenta.

O critério 3 (*Rastreabilidade bidirecional de requisitos*) é um dos critérios de maior grau de relevância para o processo de certificação, sendo considerado crítico pela instituição implementadora e profundamente analisado durante a auditoria da instituição avaliadora. Com esse critério atendido, para o MPS-BR Nível F, pode-se identificar cada item desenvolvido no projeto, desde seu nível mais abstrato até a atividade concluída pelo desenvolvedor na *Sprint* e vice-versa. Em linhas gerais, acessando um item de visão, deveria ser possível acessar cada história de usuário associada àquele item. Acessando uma história de usuário, deveria ser possível acessar cada tarefa associada àquela história de usuário e, em cada tarefa, deveria ser possível acessar o *commit* realizado pelo desenvolvedor relacionado àquela tarefa e vice-versa (Figura 4). Esse critério foi atendido parcialmente, pois a Redmine não faz a vinculação do código gerado pelo *commit* dos desenvolvedores armazenado nos repositórios do projeto no Gitlab⁴ da empresa e o projeto da Redmine. Para atender esse critério, o *Product Owner* adicionava aos comentários de cada atividade os comentários dos *commits* realizados pelos

³ Gnu Genreal Public Licence v2 - <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.

⁴ Gerenciador de repositório de software - <https://about.gitlab.com>

desenvolvedores e cabia ao gerente de projetos gerar relatórios periódicos baseados nesses comentários dos *commits*. Isso permitia criar um mapeamento que associava a atividade realizada a seu respectivo *commit* por meio do código identificador da atividade da Redmine no comentário realizado, conforme orientado no treinamento realizado.

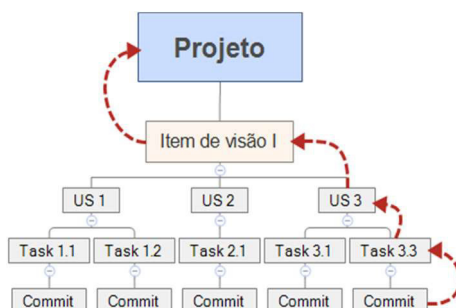


Figura 4 - Rastreabilidade Bidirecional de Requisitos

O Scrum é adotado pela GT4W no desenvolvimento de software, então o critério 4 (*Possibilidade de trabalhar com Scrum*) foi elencado pelos envolvidos e atribuído alto grau de relevância. Assim, a Redmine atendeu esse critério, pois, apesar de não possuir o *framework* Scrum como padrão, existem extensões criadas pela comunidade que permitem a utilização de Scrum. No processo para obtenção da certificação, foi utilizado o *plug-in* Redmine Plugin Scrum⁵ que permitiu adaptar a Redmine ao Scrum. Mesmo com a utilização desse *plug-in*, foi necessário configurar a ferramenta por causa da customização do Scrum empregado pela empresa, sendo necessário criar alguns papéis adicionais (Seção 2). Um *feedback* negativo das equipes quanto a esse critério foi a impossibilidade de utilizar estimativa por pontos de complexidade nas histórias de usuários, sendo a Redmine restrita à estimativa por horas de desenvolvimento.

O critério 5 (*Histórico de atualização de tarefas*) possui baixo grau de relevância, não inviabilizando a utilização da ferramenta. Esse critério foi atendido pela Redmine pois mantém a informação de quem moveu a atividade no *kanban*, bem como a data e o horário que seu estado foi alterado. Isso permitiu saber exatamente quais membros da equipe trabalharam em quais atividades.

O critério 6 (*Controle da equipe*) também foi levantado pelos envolvidos com médio grau de relevância. Esse critério consiste no controle da equipe do projeto, como o cadastro dos usuários e a atribuição de perfis e de permissões. A Redmine atendeu, pois possui configurações de perfis (que apresentavam com os perfis padrão de Scrum diante da instalação do *plug-in*) e de permissões de acesso que podem limitar ou conceder o acesso a determinada ação mediante a permissão relacionada. No processo de obtenção da certificação foi importante o cadastro de novos perfis para membros da equipe, tal como o Líder Técnico e o *Tester*, bem como as permissões relacionadas.

O critério 7 (*Autenticação de usuários*), de média relevância, foi atendido pela Redmine por fornecer *login* e senha para cada usuário que utilizar a ferramenta. Dessa forma, a Redmine “exigia” do membro da equipe ou gerente de projetos sua respectiva

⁵ <https://redmine.ociotec.com/projects/redmine-plugin-scrum>.

credencial, formada pelo *e-mail* corporativo e uma senha de acesso, criada no momento do cadastro.

O critério 8 (*Gratuita*) foi atendido por a Redmine ser gratuita. O fato de ser gratuita foi levantado por ser uma ferramenta piloto na organização, podendo adequar ou não ao processo. Assim, para evitar custos com ferramentas que poderiam ser descartadas, optou-se pela utilização de ferramentas gratuitas.

O critério 9 (*Self-hosted*) foi considerado de alto grau de relevância, sendo necessária a hospedagem internamente na empresa. Esse critério exige que a ferramenta esteja alocada nos servidores internos da empresa, sem a necessidade da utilização de nuvem para acessar o conteúdo dos projetos. Esse critério foi definido por questões de segurança e disponibilidade, uma vez que o acesso à Internet não é 100% estável no ambiente organizacional. A Redmine foi hospedada em um servidor interno dedicado na rede interna da organização, atendendo esse critério.

Um resumo dos critérios elaborados e se eles foram atendidos, não atendidos ou atendidos parcialmente pela ferramenta Redmine é apresentado na Tabela 2.

Tabela 2. Atendimento aos Critérios

ID	Critério	Grau de Relevância	Atendimento
1	Integração com o <i>Git</i>	3	Atendeu parcialmente
2	<i>Open source</i>	1	Atendeu
3	Rastreabilidade bidirecional de requisitos	3	Atendeu parcialmente
4	Possibilidade de trabalhar com <i>Scrum</i>	3	Atendeu
5	Histórico de atualização de tarefas	1	Atendeu
6	Controle da equipe	2	Atendeu
7	Autenticação de usuários	2	Atendeu
8	Gratuita	3	Atendeu
9	<i>Self-hosted</i>	3	Atendeu

Como descrito, a Redmine não atendeu totalmente todos os critérios, mas, para os pontos que não atendeu com eficácia, foram criados mecanismos que permitiram sua utilização durante o processo de certificação, atingindo desempenho satisfatório no processo de obtenção da certificação. A gestão de projetos passou a adotar a Redmine em conjunto com o Microsoft Project® como fonte de controle dos projetos, automatizando seu processo, trazendo como benefícios a agilidade aos ativos dos projetos, o controle de acesso, a geração de relatórios e a rastreabilidade de requisitos e tornando a gestão de projetos mais eficaz.

O objetivo macro da utilização da Redmine foi atingido e a certificação MPS-BR Nível F foi obtida com êxito pela empresa, sendo os processos auditados pela instituição avaliadora da SOFTEX e o resultado auditado posteriormente para a confirmação da certificação.

6. Trabalhos Relacionados

Há estudos anteriores a este sobre a implantações da certificação MPS.BR em empresas que necessitaram do apoio de ferramentas para serem bem-sucedidas na implantação dos

processos. Em um dos trabalhos [Catunda *et al.*, 2011], os autores destacam a implantação dos processos do nível F do MPS.BR em uma empresa que desejava organizar melhor seu processo de desenvolvimento de softwares. Nesse trabalho, é possível verificar a metodologia utilizada para implantar os processos necessários para o nível F em conjunto com o Scrum. Também, foi elencada a necessidade de utilizar a uma ferramenta para a gestão de projetos, sendo utilizada Redmine em conjunto com o Microsoft Word® e o Microsoft Project®. Como resultado, foi destacado que, sem o apoio de ferramentas automatizadas, o Scrum pode ser prejudicado ao implementar os processos do MPS.BR e a ferramenta Redmine permitiu mais agilidade.

Em outro trabalho [França *et al.*, 2009], foi relatada a utilização da ferramenta automatizada WebAPSEE na implementação dos processos do nível G do MPS.BR. Nesse trabalho, os autores descreveram a experiência sob o ponto de vista da utilização da WebAPSEE para auxiliar o processo de implantação em um órgão ligado à reitoria da Universidade Federal do Pará (UFPA). Os autores destacam que a ferramenta permitiu a modelagem e a execução dos processos de software de forma flexível, permitindo implantar os processos de Gestão de Projetos (GPR) e Gestão de Requisitos (GRE), obtendo a certificação MPS.BR nível G. Como lições aprendidas, foram ressaltadas situações ocorridas durante a implantação dos processos.

Em mais um trabalho [Oliveira *et al.*, 2010], foi realizado um estudo cujo objetivo foi a criação do projeto Spider, que consiste em uma suíte de ferramentas livres para apoiar a implantação dos processos de níveis G, F e E do MPS.BR. Esse projeto tem como justificativas a alta demanda por profissionais e empresas qualificados, a crescente quantidade de terceirizações envolvendo empresas nacionais e internacionais, os resultados obtidos por algumas empresas públicas e privadas do setor de Tecnologia da Informação no estado do Pará, entre outras. O trabalho elencou, dentre uma série de resultados, o levantamento de ferramentas livres e *open source* para apoio aos processos de Gerência de Projetos, Gerência de Requisitos, Gerência de Configuração e Gerência de Portfólio; o desenvolvimento de ferramentas para apoio a estimativas de projetos, quanto ao tamanho e custo dos mesmos.

Este trabalho tem como foco a análise da ferramenta Redmine na prática, juntamente com a descrição de seu modo de utilização como apoio à implantação dos processos do nível F do MPS.BR em uma empresa cuja principal linha de produtos é relacionada a projetos de desenvolvimento de sistemas de software de gestão ambiental que utilizam como base o Georreferenciamento utilizando Scrum customizado para atender aos processos organizacionais. A empresa em questão possui algumas particularidades como a execução de múltiplos projetos simultaneamente e as necessidades descritas nos critérios na Seção 2 deste artigo.

7. Considerações Finais

Este trabalho explorou o desempenho da ferramenta Redmine durante o processo de implantação de processos para obtenção da certificação MPS-BR Nível F na empresa GT4W. Em reuniões de consultorias, a comissão responsável, formada pela instituição implementadora e a diretoria executiva elaboraram nove critérios a serem atendidos pela ferramenta e, posteriormente, avaliadas na prática, analisando se ela atendeu, atendeu parcialmente ou não atendeu os critérios.

No que concerne à utilização do Scrum, a ferramenta mostrou-se eficiente durante a implantação, mas, em seguida, foi identificada a necessidade de trabalhar com múltiplos projetos em uma mesma *Sprint* na equipe, sendo uma característica que ainda não existe na Redmine. Isso inviabilizou a empresa a manter a utilização da ferramenta como fonte única de controle dos projetos. Outro ponto que contribuiu para a decisão de descontinuar a utilização de Redmine na empresa foi a integração com o Git, uma vez que a integração exige fazer uma cópia dos repositórios no diretório da Redmine para serem referenciados no projeto. Assim, há a necessidade de cada vez mais ter um espaço maior no *storage* à medida que novos projetos são criados, tornando inviável do ponto de vista de infraestrutura ao longo do tempo, uma vez que os repositórios são alocados localmente.

Do ponto de vista das equipes que utilizaram a ferramenta, o *feedback* em geral foi positivo, sendo relatados alguns problemas de adaptação, tais como, a impossibilidade de mais de uma pessoa estar associada a uma tarefa e a recomendação de, no primeiro momento, atualizar o andamento das atividades nas planilhas e na Redmine. Outro ponto criticado pelos desenvolvedores está relacionado com a estimativa das atividades, em que a equipe tinha como costume estimar com base em pontos de complexidade e a Redmine exige que a estimativa seja informada em horas de trabalho.

Durante o processo de implantação, ocorreu também uma falha no *storage* da Redmine que ocasionou a perda de informações de projetos (*Backlog* e o histórico de atualização das tarefas). Mais tarde, foi identificada que a causa do problema estava relacionada à instabilidade do ambiente no qual a ferramenta estava hospedada. Após o episódio, políticas de *backups* diários foram implementadas pela equipe de infraestrutura para evitar problemas futuros.

Um aspecto que deve ser considerado é o fato da Redmine ser uma ferramenta *open source*. Desenvolvedores podem criar ou alterar extensões e *plug-ins* para adaptá-la ao ambiente da organização, sendo necessário desenvolvedores com conhecimentos em Ruby on Rails. Assim, pode ser que alguns pontos negativos citados tenham sido resolvidos pela comunidade após a realização deste trabalho.

De modo geral, o processo de implantação do MPS-BR Nível F na GT4W foi um êxito e a Redmine permitiu a automatização de alguns processos da gestão dos projetos, que dependia integralmente de planilhas de controle, promovendo, com a utilização da Redmine, mais agilidade, flexibilidade, disponibilidade e a rastreabilidade necessárias do foi realizado nos projetos e melhorando significativamente a eficácia da gerência de projetos. Ainda existem algumas limitações relacionadas à ferramenta a serem tratadas, conforme descrito neste trabalho, mas, apesar de não atender 100% de todos os critérios, a ferramenta Redmine não deve ser descartada como uma opção para controle dos projetos, sobretudo em processos de certificação MPS-BR.

Referências

- Catunda, E.; Nascimento, C.; Cerdeiral, C.; Santos, G.; Nunes, E.; Schots, N. C. L.; Schots, M.; Rocha, A. R. "Implementação do Nível F do MR-MPS com Práticas Ágeis do Scrum em uma Fábrica de Software." X Simpósio Brasileiro de Qualidade de Software (SBQS'11), Curitiba–Brasil (2011).
- França, B. B. N. de; Sales, E. de O.; Reis, C. A. L.; Reis, R. Q. Utilização do Ambiente WebAPSEE na implantação do nível G do MPS. BR no CTIC-UFPA. 2009. VIII Simpósio Brasileiro de Qualidade de Software. pp 310- 317

- Ibbs, C. W.; Kwak, Y. H. (2002). "Assessing Project Management Maturity". Project Management Journal 31.1. pp 32-43.
- Koscianski, A.; Soares, M. dos S. Qualidade de Software. 1ª edição. Editora Novatec. 2007.
- Oliveira, S. R. B., et al. "SPIDER-Um Suite de Ferramentas de Software Livre de Apoio à Implementação do modelo MPS. BR." Anais do VIII Encontro Anual de Computação–ENACOMP (2010).
- Prado, Darci. "Maturidade em gerenciamento de projetos." Nova Lima: INDG Tecnologia e Serviços Ltda 7 (2008).
- Schwaber, K.; Sutherland, J. The Scrum Guide. Disponível em: <<http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100>> Acesso em 19 abr de 2016
- SOFTEX, 2016. Disponível em: <<http://www.softex.br/mpsbr/#>>. Acesso em: 18 abr 2016.
- SOFTEX. Guia Geral do Software, 2016. Disponível em: <http://www.softex.br/wp-content/uploads/2016/04/MPS.BR_Guia_Geral_Software_2016-com-ISBN.pdf?x15632>. Acesso em: 18 abr 2016.