1
Introdução

Em seu livro Pressman [22] define processo de software como "um arcabouço para as tarefas que são necessárias para construir software de alta qualidade".

Assim, é-se levado a inferir que o sucesso do software desenvolvido está muito ligado ao seu processo de desenvolvimento. Na Engenharia de Software, há duas abordagens principais de desenvolvimento de software, cada uma com inúmeros exemplos: os modelos de desenvolvimento dirigidos por planos e os métodos ágeis.

Modelos de desenvolvimento dirigidos por planos, como o CMMI [14], surgiram com o intuito de diminuir problemas gerados na construção do software, tentando melhorar o processo de desenvolvimento e visando garantir que a qualidade do produto desenvolvido seja satisfatória. Porém, a grande quantidade de trabalho gerado pela documentação (o que aumenta o tempo esperado de entrega, tornando-se algumas vezes inaceitável pelos clientes) começou a gerar descontentamento em relação a estes tipos de processos de desenvolvimento.

Em 2001, foi publicado o Manifesto Ágil [9], sugerindo uma maneira diferente de desenvolver software. Neste novo modelo, a extensa documentação observada nos métodos de desenvolvimento dirigidos por plano tornou-se menos importante. Em seu lugar, as equipes de desenvolvimento deveriam construir pequenos incrementos de funcionalidade, que convergissem passo a passo para um produto de software completo e satisfatório para todos os interessados (*stakeholders*).

Desta maneira, os métodos ágeis surgiram com o intuito de focar no produto a ser desenvolvido, garantindo que este será útil, utilizável e de qualidade satisfatória sempre do ponto de vista do usuário. Ou seja, um objetivo secundário dos métodos ágeis era diminuir a grande quantidade de documentação de utilidade questionável que estava sendo gerada anteriormente.

Nos métodos ágeis, a cada passo é realizado um controle de qualidade extensivo, tanto das características técnicas, quanto das características de utilidade e usabilidade. Ao final de cada passo é planejado o próximo. Isso torna o processo extremamente adaptativo a possíveis evoluções das expectativas dos usuários, como também torna visível o contínuo crescimento de funcionalidade disponível.

O inconveniente desta abordagem é a dificuldade de determinar no início do desenvolvimento o custo total dele. Porém, baseando-se em uma observação mensurada de Boehm [1], ao se seguir esta proposta tende-se a um desenvolvimento de custo mínimo (na realidade minimal), uma vez que as equipes estão continuamente mantidas sob pressão e bem focadas no problema a resolver. Boehm e Basili [10] mencionam que em torno de 50% do esforço de desenvolvimento é gasto em retrabalho inútil, sendo que este se deve em sua maioria à correção dos artefatos em grande parte devido a erros nas especificações. No desenvolvimento passo a passo, é de se esperar que este volume de correções diminua.

Para alcançar suas metas, os métodos ágeis têm em comum a valorização do desenvolvimento iterativo, realizado por desenvolvedores experientes e com a intensa cooperação dos clientes, sem a definição prévia de um abrangente plano rigoroso a ser seguido durante todo o ciclo de vida do projeto.

Deste modo, os métodos ágeis visam garantir que, em um determinado espaço de tempo (30 dias, por exemplo), novas funcionalidades (aquelas com capacidade de formar um pacote entregável ao cliente) serão desenvolvidas. Ou seja, mesmo que o software ainda não esteja completo, isto é, com todos os requisitos implementados, ele terá algumas funcionalidades – aquelas consideradas as mais prioritárias pelo usuário – que funcionam corretamente e que agregam valor do ponto de vista do usuário.

Esta forma de desenvolver software tem recrutado um grande número de adeptos, o que permite a execução destes métodos em diversos cenários. Enquanto que, em um momento inicial, os processos ágeis eram mais utilizados em sistemas de missão não-crítica, atualmente, o seu uso vem se difundindo também entre projetos de softwares cada vez mais complexos e até em projetos de missão crítica. Assim, as limitações dos métodos ágeis começam a aparecer, bem como diversas dúvidas em relação ao seu uso na prática.

Conseqüentemente, procura-se hoje uma solução de meio termo, em que se adiciona algum grau de formalidade ao processo ágil, assegurando que características não-funcionais de qualidade (ex. testabilidade, evolutibilidade, reuso) sejam alcançadas também [15].

O *Scrum* [11] – um exemplo dos métodos ágeis e objeto de estudo deste trabalho – é uma maneira de gerenciar projetos, que divide o processo de desenvolvimento em duas etapas: uma etapa maior (que dura cerca de um mês), conhecida como *Sprint*, em que é desenvolvida uma série de itens e uma etapa menor, de acompanhamento diário, que faz parte da etapa maior, como pode ser visto na Figura 1.

Assim, o processo de desenvolvimento é constantemente inspecionado, permitindo que adaptações necessárias sejam realizadas tão logo sejam observadas e a baixo custo em virtude da redução do retrabalho inútil.

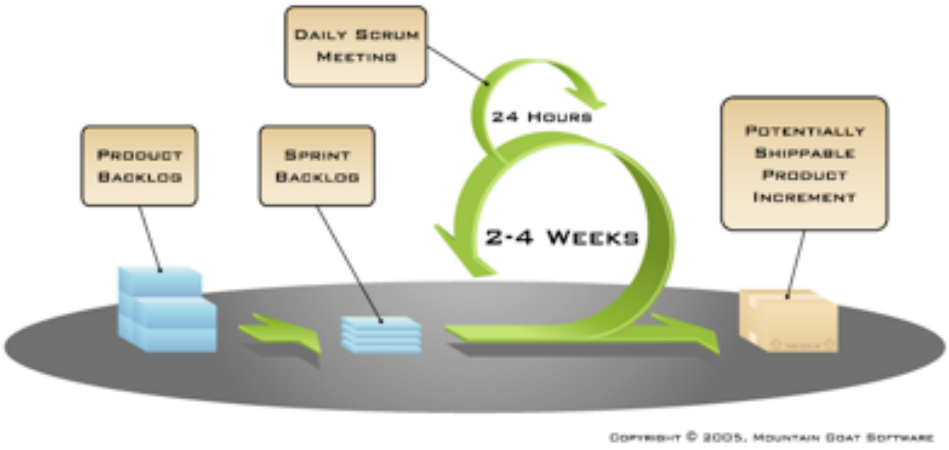

Figura 1 – Conceitos que integram o Scrum (http://digibit.co.uk/AgileScrum.aspx).

Motivação

Este trabalho tem como objetivo relatar a experiência de implantação de *Scrum* em uma equipe responsável pelo *website* de uma organização internacional. A atividade fim desta organização não é o software, entretanto necessita deste, em especial aplicações *Web*, para alcançar os seus objetivos.

O *website* da organização representa um grande aliado de marketing da mesma, porém somente nestes últimos anos é que se tornou mais evidente a necessidade de um time de desenvolvedores trabalhando na equipe responsável pelo mesmo. Anteriormente, esta equipe era composta somente por editores. Por

ser uma necessidade recente, a equipe de desenvolvedores e a sua infraestrutura ainda não estavam bem estabelecidas.

Decidiu-se, então, utilizar o *Scrum* como método de gerenciamento de projetos ali desenvolvidos, porém, este por si só não garante a qualidade dos mesmos. É muito importante que, em conjunto com a implantação do *Scrum*, tenha-se também uma preocupação com a infraestrutura de trabalho e com os métodos de desenvolvimento *Web*. Antes deste estudo, não havia um processo definido de trabalho na equipe estudada, fato que contribuía para gerar uma alta taxa de ocorrência de problemas de retrabalho e de *bugs* no *website*.

No ano de 2010 foi divulgado que o *website* desta organização seria o principal meio de comunicação e *marketing* da mesma. Portanto, era muito importante que houvesse uma estruturação do departamento responsável por este produto para viabilizar a construção e a melhoria de ferramentas relacionadas à gerência de seu conteúdo.

O presente trabalho tem por objetivo, então, relatar a implantação de *Scrum* e da respectiva infraestrutura de trabalho que permita a obtenção dos benefícios deste, utilizando recursos financeiros limitados, em uma equipe multidisciplinar de uma organização tradicional.

Objetivos

Com este trabalho, deseja-se aumentar a qualidade do site da organização estudada e das ferramentas de gerenciamento de seu conteúdo, bem como melhorar o gerenciamento dos projetos em desenvolvimento, através da utilização do *Scrum* e de boas práticas da Engenharia de Software. Porém, para que se possa verificar o impacto das mudanças realizadas no cenário da organização, é necessária a utilização de métricas. Sem a ajuda das métricas, torna-se difícil a comprovação de que um método foi bem sucedido ou não.

O primeiro passo para a definição das métricas a serem adotadas é a escolha dos objetivos a serem atingidos. Este trabalho tem três principais objetivos, que são detalhados nos próximos parágrafos. A partir destes objetivos, algumas métricas foram escolhidas para auxiliar na correta análise dos resultados. Estas métricas estão explicitadas no Capítulo 3.

Uma preocupação constante da gerência é a quantidade de erros que está presente no site da organização. Cada vez que uma página apresenta um erro que não a deixa ser exibida, um email é enviado aos desenvolvedores com os detalhes desse erro. Ou seja, é possível se verificar quais páginas ou módulos possuem mais erros, qual o tipo de erro e qual a causa do mesmo. Muitas vezes, a causa destes erros está relacionada ao processo de atualização do site: o responsável pela atualização não copia todos os arquivos necessários ou duas pessoas alteram simultaneamente um mesmo arquivo, ocasionando perda de trabalho. Com este estudo, a partir da utilização de padrões, frameworks, um controle de versão e testes, deseja-se diminuir a quantidade de erros encontrados. Isto é, a partir da criação da infraestrutura necessária para acompanhar o *Scrum*, pretende-se diminuir estes problemas.

No ano de 2010, o principal meio de comunicação externa da organização passou a ser a Internet, por meio de seu *website* e das ferramentas de rede social (como Facebook, Twitter, Orkut, Youtube e Fliker). Portanto, uma segunda meta a ser atingida é o aumento da popularidade da organização nestas plataformas.

A estratégia a ser adotada para este fim é a propaganda destas ferramentas no próprio site da organização, mostrando atualizações (notícias publicados no Twitter, por exemplo) e eventos correntes (como discussões, entrevistas e seminários no Facebook).

Porém, um problema que existe relacionado às propagandas das redes sociais no site é a tarefa de atualização destas informações: é feito manualmente por um editor *Web*, que necessita alterar dados como foto, título, descrição e link em várias páginas do site. Esta forma de trabalhar, além de estar propensa a erros, não condiz com o espírito das redes sociais – que pregam a instantaneidade de informações. Ou seja, a Engenharia de Software deve ser utilizada para facilitar o gerenciamento deste conteúdo, de modo que este possa ser fácil e rapidamente alterado.

Por último, também se deseja aumentar a popularidade do site em si. Uma das premissas dos métodos ágeis é a preocupação com o cliente, ou seja, a importância de atender às suas expectativas. Neste contexto, a preocupação com o cliente pode se refletir no aumento da qualidade do site desenvolvido, que por sua vez aumenta a popularidade do mesmo.

O aumento da popularidade depende também da utilização de algumas técnicas (como aquelas associadas aos motores de busca, para que as páginas sejam encontradas pelo público alvo em serviços de busca, como Google). Porém, o aumento da popularidade também está relacionado à qualidade do produto apresentado, isto é, do site em si e do conteúdo disponibilizado nas ferramentas de rede social. Ferramentas como o Google Analytics (http://www.google.com/analytics/) e Web Trends (http://www.webtrends.com/) são usadas para verificar a popularidade do site em questão.

Uma vez que a equipe estudada não seguia qualquer método de gerência de projetos, resolveu-se utilizar o *Scrum* para organizar os projetos a serem desenvolvidos e até mesmo as atividades de melhoria de infraestrutura que deveriam ser feitas. Portanto, ao final deste trabalho, é possível verificar o impacto da utilização do *Scrum* pela equipe, as dificuldades e os benefícios encontrados.

Organização da Dissertação

Este documento está assim dividido: no Capítulo 1, tem-se a introdução, a motivação e a proposta.

O Capítulo 2 contém conceitos iniciais de *Web* 2.0, necessários para o correto entendimento do trabalho – uma vez que a equipe estudada é responsável pela *website* da organização e utiliza bastante os conceitos da *Web* 2.0 em seu trabalho diário.

O Capítulo 3 contém conceitos de qualidade de software, garantia de qualidade e controle de qualidade. Estas informações formam a base do trabalho, uma vez que é importante entender os conceitos relacionados à qualidade de software para identificar o que se espera de um software produzido comercialmente.

O Capítulo 4 foca nos processos ágeis, relativamente recentes e ainda desconhecidos por muitos integrantes da comunidade da Engenharia de Software. Nesta seção, é descrito o surgimento de tais processos, sua aplicabilidade, benefícios, desvantagens e alguns exemplos. Entre os exemplos citados, é destacado o *Scrum*, pois este é o método utilizado na organização objeto deste

estudo. Também há neste capítulo uma comparação entre os processos ágeis e aqueles dirigidos por planos.

No Capítulo 5 são estudadas várias ferramentas propostas pela comunidade de desenvolvedores com o intuito de verificar seus pontos positivos, suas restrições e a maneira com a qual elas interagem entre si. A idéia é utilizar o maior número de ferramentas no processo de desenvolvimento da organização estudada, dentre as aqui especificadas, com o objetivo de facilitar o trabalho diário da equipe.

No Capítulo 6 é descrita a configuração inicial do ambiente, explicitando detalhes, como: pessoas envolvidas e suas características, maiores informações em relação ao software desenvolvido e infraestrutura de trabalho. Este capítulo também contém dados do ambiente de desenvolvimento (IDE utilizada, ferramentas e descrição dos computadores).

O Capítulo 7 traz as informações relacionadas às mudanças realizadas ao longo deste trabalho, com o objetivo de atingir as metas iniciais estabelecidas. Isto é, neste capítulo são descritas as mudanças de hardware e de software realizadas, bem como as alterações no modo de gerenciamento dos projetos em andamento.

O Capítulo 8 contém os dados encontrados no início e no final do processo, comparando os mesmos, a fim de verificar os ganhos obtidos com as alterações propostas. Nele também há uma seção com os pontos positivos, os problemas encontrados durante a realização do estudo e suas respectivas críticas.

O Capítulo 9 é a conclusão do trabalho, que apresenta também propostas de trabalhos futuros, seguido pela bibliografia.