

7 Mudanças Realizadas

Este capítulo tem o objetivo de detalhar as mudanças realizadas no ambiente de trabalho da equipe estudada. Ele cita as alterações no produto de software utilizado pela equipe, que recebeu algumas novas funcionalidades para facilitar o trabalho da mesma; cita também mudanças na maneira de gerenciamento dos projetos, a partir da introdução do *Scrum* e alterações na infraestrutura de trabalho da equipe, que necessitava de melhorias (de hardware, de software e de utilização de boas práticas).

7.1.Melhorias no Gerenciamento de Projetos

A primeira ação tomada na tentativa de melhorar o processo de trabalho da equipe estudada foi a realização de workshops envolvendo conceitos como métodos ágeis (*Scrum*) e a importância do controle de versão dos arquivos. Em seguida, era importante estabelecer a lista de projetos e suas respectivas prioridades, para verificação do que deveria ser desenvolvido em seguida, uma vez que a equipe tinha um fluxo de trabalho que não poderia ser esquecido durante as atividades de mudança e melhoria.

Uma vez que os projetos nesta unidade dificilmente envolvem todos os componentes da equipe (isto é, para cada projeto, um grupo de pessoas é escolhido e são aquelas que irão trabalhar durante todo o projeto para que ele seja concluído) as equipes são quebradas em tamanhos menores, mais fáceis de gerenciar e de aplicar conceitos do *Scrum*.

Alguns projetos foram escolhidos para utilizar *Scrum*, desenvolvendo na equipe a experiência e segurança necessárias para a utilização deste método ágil. Em conjunto, atividades de padronização e organização da infraestrutura de trabalho foram desenvolvidas.

Entre o time dos desenvolvedores foi escolhida uma série de padrões de desenvolvimento e codificação, além do uso de um *framework* e de ferramentas que permitem uma melhor integração entre si.

Após a conclusão das mudanças, que são detalhadas no próximo capítulo, foram verificadas as métricas estabelecidas, com o objetivo de visualizar as vantagens obtidas.

7.1.1. Adoção do *Scrum*

Decidiu-se utilizar o *Scrum* como forma de gerenciamento de três grandes projetos em andamento, dentre os vários que estão sendo executados. Assim, pode-se verificar se houve algum ponto positivo gerado pela utilização de *Scrum*, quando comparados aos projetos que não tiveram a sua maneira de gerenciamento alterada. Com relação à melhoria da infraestrutura de desenvolvimento – integrando ferramentas e mudando algumas rotinas de trabalho – todos os projetos são afetados igualmente, ou seja, independe da utilização de um método ágil para o seu gerenciamento.

Decidiu-se utilizar o *Scrum* em projetos mais longos (duração média de dois ou três meses), que envolvessem principalmente atividades de programação, realizadas pelos desenvolvedores. A formação destes é mais técnica, portanto eles conhecem alguns conceitos utilizados neste trabalho e assim se tornam pessoas mais adeptas a mudanças do que os editores. Deste modo, é mais fácil convencê-los a tentar algo novo e, uma vez que o sucesso do método seja verificado, mais integrantes da equipe podem se tornar confiantes a utilizá-lo. Porém, como pode ser visto na Seção 7.3, o *Scrum* foi utilizado anteriormente em dois pequenos incrementos de funcionalidade para que alguns integrantes da equipe se familiarizassem com os conceitos deste método ágil.

Decidiu-se utilizar *Sprints* de uma semana, uma vez que os projetos não tinham duração muito longa e que muitas vezes os requisitos não estavam claramente especificados, ocorrendo muitas alterações durante o processo de desenvolvimento, inclusive de prioridades. Portanto, com *Sprints* de uma semana, o PO pode verificar o andamento do projeto e ter uma idéia mais concreta do que gostaria de ver nas próximas iterações e do que é possível ser feito neste espaço de tempo, tornando mais fácil o gerenciamento.

7.1.2. Utilização de Ferramenta para a Gerência de Projetos

Em relação à ferramenta de gerenciamento de projetos, decidiu-se por utilizar uma ferramenta de suporte ao *Scrum* nos projetos que utilizariam este tipo de método ágil. O desejo de se utilizar uma ferramenta se deve ao fato de que relatórios constantes de acompanhamento dos projetos devem estar sempre disponíveis aos gerentes.

Assim, com a utilização de uma ferramenta que registre o andamento do projeto, com a opção de impressão de relatórios, este requisito seria atendido. Outro ponto muito importante é que o PO gostaria de poder ter acesso ao *Sprint Backlog* e ao *Burndown Chart* online, sem precisar ir ao local em que se eles encontravam fisicamente. Algumas ferramentas disponíveis no mercado foram pesquisadas e se optou por utilizar o Banana Scrum (<http://www.bananaScrum.com/>).

Banana Scrum é uma ferramenta *web* para gerenciamento de projetos que usam *Scrum*, desenvolvida pelo Code Sprinters. Há também uma versão para instalação nos servidores locais das empresas, caso sejam requeridos mais segurança e controle. Esta ferramenta é bastante interessante, uma vez que implementa todos os requisitos de utilização do *Scrum*: é possível criar vários projetos independentes; *Backlogs* com a lista de itens e seus respectivos esforços; *Sprints* com a lista de itens e respectivas tarefas, além do tempo necessário para completá-las; cadastrar usuários com diferentes papéis (time, SM ou PO) dentro dos diferentes projetos; há vários gráficos disponíveis, entre eles o *Burndown Chart*; é possível cadastrar impedimentos; há a opção de gerenciar *tags*, de modo que itens possam ser agrupados e mais facilmente encontrados posteriormente; além de possuir uma interface muito fácil de utilizar. Esta ferramenta também permite o anexo de documentos, importação e exportação de arquivos em formato CSV e geração de relatórios.

No início deste trabalho, quando passou a ser utilizada, a ferramenta tinha uma interface distinta e estava em processo de desenvolvimento. Problemas eram encontrados com uma certa frequência, uma vez que estava em fases de testes. Porém, atualmente, ela é uma ferramenta paga, com diferentes opções de plano, com uma nova interface e sem as limitações encontradas anteriormente.

Há também uma versão não-paga, que não possui todas as funcionalidades das versões pagas (por exemplo, não há o *Timeline*, funcionalidade que apresenta o histórico dos *Sprints* e as suas evoluções), além de possuir uma limitação maior de espaço e de quantidade de usuários.

Para este trabalho, uma vez que se estava utilizando o *Scrum* de uma maneira experimental, optou-se por utilizar a versão educacional da ferramenta, que possui todas as funcionalidades das versões profissionais, porém, não é paga. Caso a organização acredite que a ferramenta agrega um valor positivo à rotina de trabalho, ao final do estudo, poderá adquiri-la.

7.2.Melhorias na Infraestrutura de Trabalho

Uma vez que o *Scrum* necessita da utilização de boas práticas da Engenharia de Software para alcançar seus objetivos, foram realizadas alterações também na infraestrutura de trabalho da equipe.

7.2.1.Utilização de Controle de Versão

A primeira mudança realizada foi a criação de um sistema de controle de código, utilizando Subversion. A implantação é relativamente simples, o maior impacto, porém, foi a falta de experiência da equipe que passaria a utilizá-lo. Alguns treinamentos foram realizados, além de algumas semanas de apoio a dúvidas que surgiam constantemente, principalmente dos editores.

Escolheu-se criar o repositório de código no principal servidor de desenvolvimento utilizado pela equipe e assim os problemas de perda de trabalho e serialização na alteração de um determinado arquivo foram resolvidos.

Também foi criada uma política de atualização de versão. A cada mudança grande – adição de uma funcionalidade ou a alteração de mais de dez arquivos simultaneamente – do site, envolvendo uma nova funcionalidade adicionada, é lançada uma nova versão do mesmo. Para mudanças menores, não há lançamento de versão e os arquivos são copiados para os outros servidores. Porém, quando há lançamento de versão, é necessário incluir esta informação no repositório do Subversion, criando um *commit*.

Com o objetivo de facilitar o trabalho de sincronização e atualização dos servidores, criou-se uma ferramenta para este fim. Esta ferramenta foi desenvol-

vida em Java, com o auxílio da biblioteca SVNKit (<http://svnkit.com/>) – uma biblioteca open-source que implementa todas as funcionalidades do SVN, provendo uma API que manipula o repositório do Subversion dentro da própria aplicação Java.

Esta aplicação funciona da seguinte forma: o usuário entra com os dados da versão para a qual deseja atualizar o site e escolhe os servidores que serão sincronizados. A aplicação sabe em qual versão se encontra cada servidor (a partir de uma informação guardada no banco de dados) e copia todos os arquivos necessários para a atualização nos servidores escolhidos.

Os arquivos copiados são buscados a partir do histórico de alterações do Subversion, com a utilização da API do SVNKit. Também é possível escolher uma determinada versão do histórico do Subversion para atualizar o site com os arquivos que foram alterados naquela versão.

Desta maneira, automatizou-se o processo de atualização e sincronização dos arquivos presentes nos servidores, de extrema importância para o trabalho desenvolvido na unidade. Muitos erros relacionados a problemas durante a atualização dos servidores – ou porque os arquivos eram colocados em lugares incorretos, ou porque se esquecia de atualizar determinados arquivos – foram bastante minimizados com esta prática.

7.2.2.Melhorias na IDE Utilizada

Em relação à infraestrutura de desenvolvimento, o principal passo a ser tomado era a utilização de um sistema de controle de versão de código. Era fundamental que este sistema estivesse integrado ao ambiente de desenvolvimento dos membros da equipe. Optou-se por utilizar o Subversion, uma vez que o time de desenvolvedores se sentia mais seguro com esta ferramenta, devido às suas experiências anteriores.

Porém, o Dreamweaver 3, como visto no Capítulo 5, não dá suporte ao controle de versão, limitação esta que impossibilitou a continuidade de seu uso. Decidiu-se então, fazer uma experiência com o Eclipse (utilizando-se o *plugin* para Coldfusion, CFEclipse), com o intuito de verificar se a equipe poderia utilizar este novo ambiente. Uma vez que estas ferramentas são *open source*, este teste poderia ser feito sem muitos problemas de recursos. Caso funcionasse, a

organização compraria o Coldfusion Builder, que é uma versão paga e melhorada deste ambiente.

Por duas semanas utilizou-se o Eclipse com o CFEclipse, porém algumas limitações foram encontradas. Os editores, principalmente, não se adaptaram à nova ferramenta, uma vez que o Eclipse está mais direcionado ao mercado de desenvolvedores. Estes, entretanto, não tiveram grandes problemas para utilizar esta nova ferramenta.

Desta maneira, a decisão tomada foi de upgrade para o Dreamweaver 4, uma vez que este tem suporte ao controle de versão. Os desenvolvedores, por sua vez, poderiam utilizar o Coldfusion Builder, pois o controle de versão possibilita essa liberdade de escolha de ferramenta – o ponto fundamental é que esta tenha suporte ao Subversion.

7.2.3.Melhorias de Hardware

Uma das medidas tomadas em relação à estrutura física de trabalho foi a solicitação de mais memória para as máquinas, para que estas possam executar programas mais pesados, sem se tornarem lentas. A configuração mínima de 2 GB de memória RAM se tornou um requisito fundamental para a melhoria da infraestrutura.

7.3.Melhorias no Produto

O gerenciador de conteúdo utilizado pelos editores da equipe já se encontrava desenvolvido quando este trabalho foi iniciado. Porém, ao se estudar as práticas de *Search Engine Optimization* (SEO), verificou-se que a ferramenta deveria passar por algumas modificações, com o objetivo de se adequar às melhores práticas.

Portanto, novas funcionalidades foram adicionadas ao gerenciador de conteúdo: o preenchimento de campos, como título, descrição e cabeçalhos do site e a reescrita de *url* para a apresentação de *urls* amigáveis. Desta forma, alguns dos principais requisitos relacionados à encontrabilidade de um site foram atendidos.

Outro ponto de mudança foi o relacionado às mídias sociais. Foi criada no gerenciador de conteúdo a possibilidade de gerenciar também as informações relativas às ferramentas de mídias sociais, como citado na Seção 7.3.2.

7.3.1. Melhorias Relacionadas à SEO

Como visto na Seção 2.2, há uma série de ações a serem realizadas para aumentar a popularidade de um site. Este conjunto de melhorias requer muito trabalho e consumo de recursos e nem mesmo os próprios buscadores conseguem atender a todos os requisitos.

Em um estudo das próprias páginas de produtos Google, verificou-se que há muito o que melhorar em suas páginas quando se verificam as técnicas de SEO [2]. Por exemplo, os títulos das páginas estão inadequados em 90% das páginas, a *meta-tag* “description” só é usada em 33% das páginas, a utilização da *tag* “alt” para as imagens é de 58% e a utilização de cabeçalhos <h1> adequados é de 43%.

Na organização estudada, antes de se verificar as técnicas de SEO, não havia qualquer tipo de preocupação em relação aos fatores que a alteram, vistos na Seção 2.2. Portanto, muito trabalho teve que ser feito e ainda há muitos pontos de melhoria.

Uma vez que a equipe utiliza um gerenciador de conteúdos para suas páginas, desenvolvido internamente pelos desenvolvedores da mesma, este teve que sofrer algumas alterações: adição de campos que representem a *tag* <title>, cabeçalhos e *meta-tag* “description”. Também foi desenvolvido um módulo de reescrita de *urls*, para que elas deixassem de serem compostas por números e códigos e passassem a possuir uma descrição amigável do conteúdo da página.

As alterações feitas na ferramenta que gerencia o conteúdo do site foram controladas por meio de *Scrum*. Três desenvolvedores foram envolvidos nas tarefas de melhoria propostas e em apenas um *Sprint* (de um mês), foi possível desenvolver todas as funcionalidades requeridas. Este foi o primeiro contato da equipe com *Scrum* e foi ser considerado bem-sucedido, uma vez que no final do ciclo, as funcionalidades estavam prontas, aprovadas e passaram a ser utilizadas pelo restante da equipe.

Uma vez que as mudanças não seriam muito grandes (as funcionalidades requeridas não eram muito complexas), esta foi uma boa maneira de se começar a utilizar o *Scrum* pela equipe. As tarefas criadas para se terminar com sucesso o ciclo eram simples, o que facilitava a estimativa de horas e de trabalho a ser realizado. Deste modo, em apenas um *Sprint* foi possível entregar todas as

funcionalidades e, ao final do mesmo, os desenvolvedores já estavam mais familiarizados com os conceitos do *Scrum*.

7.3.2.Melhorias Relacionadas aos Conteúdos das Mídias Sociais

A mudança aqui realizada foi a criação de uma funcionalidade no gerenciador de conteúdo utilizado pela equipe para facilitar a atualização das informações relacionadas às ferramentas de redes sociais.

Deste modo, os editores apenas precisam atualizar informações como título, descrição, imagens e links relacionados dentro da ferramenta, sem alterar os arquivos das páginas presentes nos servidores. Ou seja, esta seria mais uma funcionalidade requerida para o gerenciador de conteúdo utilizado pelos editores da equipe de Internet.

Resolveu-se utilizar novamente o *Scrum* para o desenvolvimento desta nova entrega e desta vez foram envolvidos apenas dois desenvolvedores. Visto que a quantidade de trabalho necessária seria menor, pelo fato de a mudança ser pequena, o *Sprint* também foi diminuído para apenas duas semanas. Os dois desenvolvedores responsáveis por esta nova funcionalidade participaram do desenvolvimento citado na seção anterior, que utilizava o *Scrum*. Este fato também contribuiu para a decisão de diminuir a duração desta nova iteração.

Os dois desenvolvedores trabalharam por duas semanas para o desenvolvimento desta nova funcionalidade, sempre no período da manhã apenas, uma vez que possuíam outras atividades à tarde. Durante estas duas semanas, trabalharam sempre em pares para a entrega da funcionalidade, o que acelerou bastante o desenvolvimento e permitiu a entrega com sucesso ao final do ciclo.

7.4.Resumo

Neste capítulo foi possível verificar as mudanças realizadas no ambiente de trabalho da equipe de Internet durante o desenvolvimento deste estudo. Três mudanças fundamentais foram implantadas: a mudança no produto de software utilizado pelos integrantes da equipe, mudanças na infraestrutura de trabalho (alterações de hardware e implantação de um conjunto de ferramentas para

auxiliar o trabalho) e mudanças no gerenciamento dos projetos, com a adoção do *Scrum*.

A implantação do *Scrum* trouxe benefícios, uma vez que os gerentes conseguem ter um melhor acompanhamento dos projetos. Para os desenvolvedores também há melhorias, pois eles conseguem planejar seu trabalho, concentrando-se nas atividades mais prioritárias, além de possuírem requisitos mais claros do que deve ser implementado.

Uma infraestrutura que apoiasse o trabalho diário da equipe, formada por ferramentas que automatizassem processos sujeitos a falhas, era também de extrema importância. Assim, utilizaram-se conceitos da Gerência de Configuração para auxiliar no gerenciamento de mudanças, de problemas e de incidentes. Passou-se a utilizar um sistema de controle de versão de código e foi desenvolvida uma ferramenta de atualização dos servidores. Somente estas melhoras já foram extremamente positivas, uma vez que substituíram procedimentos manuais, muitas vezes demorados e que geravam problemas posteriores.

É importante observar o grau de integração das ferramentas introduzidas por este estudo, fato muito importante para a melhora da infraestrutura de trabalho. Ferramentas corretamente utilizadas e que auxiliam o trabalho diário das pessoas podem ser um problema quando utilizadas em conjunto – se não puderem ser integradas corretamente. Isto é, uma equipe pode saber utilizar corretamente várias ferramentas individualmente, que auxiliam bastante o seu trabalho. Porém, se este conjunto de ferramentas for usado simultaneamente, ele pode comprometer o trabalho da equipe (caso as ferramentas não sejam integradas). O que ocorre é que as alterações feitas em uma ferramenta acabam não sendo refletidas na outra, o que pode gerar retrabalho e inconsistência.

Assim, neste trabalho, por exemplo, foi necessário criar um módulo de gerenciamento de configuração para controlar parte das mudanças realizadas, copiando arquivos do repositório de dados para os servidores. Esta ferramenta comunica-se com o repositório de dados, o que facilita bastante o trabalho de atualização dos servidores. Porém, outra melhora que poderia ser feita é a criação de um *plugin* para o Eclipse que faça esta tarefa. Ou seja, por dentro da própria IDE utilizada, seria possível atualizar os servidores.

Neste capítulo também se destacou a importância de treinamento para o aumento da produtividade. Uma vez que a principal ferramenta utilizada não

sofreu grandes alterações (Dreamweaver 3 para o Dreamweaver 4), não foram necessárias muitas horas de treinamento. O treinamento se concentrou na utilização do controle de versão (Subversion) e da ferramenta para atualização de servidores. A curva de aprendizado neste caso foi bem pequena, em torno de duas semanas.

Em relação à utilização do *Scrum*, apenas o time de desenvolvedores teve contato com este método ágil, no desenvolvimento de alguns projetos. Sua adoção foi bastante positiva, pois organizou o trabalho desenvolvido e permitiu uma maior visibilidade dos projetos. Isto é, os gerentes tinham a oportunidade de acompanhar o andamento das tarefas.

No próximo capítulo, cada uma dessas mudanças é citada novamente, para que se possam perceber seus benefícios, as dificuldades encontradas para implantá-las e as melhorias que ainda precisam ser realizadas.