Integrando a Gestão de Configuração do CMMI com o Gerenciamento de Liberação do ITIL

Marlon Gracietti de Amorim, Cláudio Ratke

Departamento de Sistemas e Computação
Universidade Regional de Blumenal (FURB) – Blumenau, SC – Brasil

amorim.mga@gmail.com, ratke@inf.ufsc.br

Abstract. *This study demonstrates the aggregation of concepts in an application that presents information relevant to the planning and monitoring of the release of new software releases. Using the concepts of the Capability Maturity Model Integration (CMMI) in side of the software house and integrates with release management of the Information Technology Infrastructure Library (ITIL) in perspective of the customer. The results reflect in a defined and controlled process that starts with the change request until the safe and planned update in production environment.*

Resumo. *Este trabalho demonstra a agregação dos conceitos em uma aplicação que apresenta informações relevantes ao planejamento e acompanhamento da liberação de novos releases de software. Utilizando os conceitos do Capability Maturity Model Iflntegration (CMMI) no lado da software house e integra com a gestão de liberação do Information Technology Infrastructure Library (ITIL) na perspectiva do cliente. Os resultados obtidos refletem em um processo definido e controlado que inicia desde a solicitação da requisição de mudança até a atualização segura e planejada no ambiente de produção.*

1. Introdução

Com o aumento da complexidade da infraestrutura de TI e da dependência das organizações em relação aos serviços de TI, torna cada vez mais necessário o gerenciamento detalhando a liberação de softwares para uso pela organização (MAGALHÃES; PINHEIRO, 2007).

Por tanto, as empresas desenvolvedoras de softwares precisam adotar boas práticas para obter o mínimo de impacto e a máxima agilidade na entrega de seus serviços de software. Segundo Magalhães e Pinheiro (2007) são muito frequentes o emprego de processos simplificados para o gerenciamento de liberação, muitas vezes gerando falhas na entrega do serviço de software. Causando transtornos e desgaste ao cliente, bem como custos e trabalhos não planejados para a equipe de desenvolvimento de software.

Nesta solução uniu-se a utilização de práticas recomendadas pelo *Capability Maturity Model Integration* (CMMI) integradas as melhores práticas de *Information Technology Infrastructure Library* (ITIL). A utilização destas duas metodologias resulta em um processo que inicia com o planejamento da liberação da versão até a implantação e integração da versão liberada no ambiente de produção do cliente.

A escolha por estas duas metodologias, deve-se ao fato de serem bastante difundidas no âmbito de gerenciamento de serviço e deterem um reconhecimento internacional. Freqüentemente elas são exigidas como requisitos para indicação de qualidade de prestação de serviço.

2. CMMI (Capability Maturity Model Integration)

O CMMI é um modelo de melhoria de processos e programa de avaliação de serviços. Composto por práticas e processos recomendados durante o ciclo de vida de um produto. No âmbito da metodologia, os processos são classificados de acordo com seus níveis de maturidade, que são definidas como: inicial, gerenciado, definido, quantitativamente gerenciado e otimização. O CMMI é administrado e comercializado pela Carnegie Mellon University.

2.1. Gestão de configuração

Segundo Sommerville (2003), gestão de configuração é a utilização de modelos e padrões para gerenciar um software em desenvolvimento. Alterações em suas funcionalidades, correções e adaptações, geram diferentes versões do sistema. A gerência de configuração serve para evitar conflitos nos itens de configuração modificados.

A gestão de configuração de software pode ser entendida como uma disciplina que permite manter a evolução de produtos de software sobre controle e contribuir para atender as exigências de qualidade e prazo (ESTUBLIER, 2000).

Segundo Mellon (2006) os produtos de trabalho colocados sob a gestão de configuração incluem os produtos que são entregues ao cliente, produtos de trabalho internos selecionados, produtos adquiridos, ferramentas e outros itens que são utilizados para criar e descrever esses produtos de trabalho.

2.2. Gerenciamento de mudanças

As necessidades e requisitos organizacionais modificam o tempo de vida útil de um sistema. Isso requer que mudanças sejam feitas no software. Um processo definido de gerenciamento de mudanças associado a ferramentas de apoio garantem que essas mudanças sejam registradas e aplicadas ao sistema de maneira econômica (SOMMERVILLE, 2003).

Os procedimentos de gerenciamento de mudança devem ser concebidos pra assegurar que os custos e os benefícios das mudanças sejam adequadamente analisados e as mudanças em um sistema sejam feitas de maneira controlada (SOMMERVILLE, 2003).

O primeiro estágio do processo de gerenciamento de mudanças é o preenchimento de um formulário de solicitação de mudança *Change Request Form* (CRF), em que o solicitante estabelece a mudança requerida no sistema. Tem como objetivo registrar a solicitação da mudança, recomendações, custo estimado, aprovação. Ele pode também incluir uma seção na qual o engenheiro de manutenção faz um esboço de como a mudança deverá ser implementada. As solicitações de mudança devem ser registradas no banco de dados de configuração (SOMMERVILLE, 2003).

Uma vez que o formulário de solicitação de mudança tenha sido submetido, ele é analisado a fim de ser verificado se a mudança é válida. Algumas solicitações de mudanças podem ocorrer em virtudes de erros de compreensão, e não de defeitos do sistema. Outras podem se referir a defeitos já conhecidos (SOMMERVILLE, 2003, p. 555).

Quando um conjunto de mudanças é aprovado, ela é encaminhada para a equipe de desenvolvimento ou manutenção, para implementação. À medida que os itens de

configuração são modificados, deve ser mantido um registro de mudança feito em cada um deles (SOMMERVILLE, 2003, p. 556).

3. ITIL (IT Infrastructure Library)

O ITIL consiste em uma série de documentos que são utilizados para auxiliar a implementação de uma estrutura de ciclo de vida de serviços de TI. Este quadro personalizável define como gerenciamento de serviço é aplicada dentro de uma organização. Ele também alinhado com o padrão internacional, ISO 20000. (AXELOS, 2014).

3.1. Gerenciamento de liberação

O gerenciamento de liberação é o processo de planejar, compilar, testar e implantar uma versão de distribuição de software, bem como o controle de versionamento e a sua armazenagem. Tem como objetivo garantir a qualidade do ambiente de produção usando procedimentos formais e verificações quando se implementam novas versões (BON, 2006).

Um release de sistema é uma versão que é distribuída para os clientes. Cada release de sistema deve incluir nova funcionalidade ou se destinar a uma diferente plataforma de hardware. Há sempre muito mais versões de um sistema do que releases, uma vez que as versões são criadas dentro de uma organização para o desenvolvimento interno ou testes, e nunca são liberadas para clientes (SOMMERVILLE, 2003, p. 557).

Ter uma política de gerenciamento de liberação também é muito importante para o sucesso de um projeto. Como o desenvolvimento de um projeto é bastante dinâmico, torna-se freqüente a liberação de várias versões do produto (BECTA, 2004).

O processo de Gerenciamento de Liberação também contribui para aumentar a eficiência da introdução das mudanças no ambiente de produção, combinando-as em uma única liberação e realizando a implementação em conjunto (MAGALHÃES; PINHEIRO, 2007).

3.2. Ramos de desenvolvimento (Branches)

A forma mais comum de lançamento de versões nos projetos é o “congelamento” do código, isto é, a criação de uma linha de base. A partir disto, nenhuma funcionalidade é adicionada ao código base, apenas os *bugs* são corrigidos para o lançamento de uma versão estável. Em seguida é criada uma nova liberação que compreendem uma ou mais mudanças autorizadas (BON, 2006, p. 96).

Segundo Sussman, Fitzpatrick e Pilato (2011) um ramo é uma linha de desenvolvimento que existe independente de outra linha, e ainda, partilham um histórico em comum. Um ramo sempre se inicia como cópia de outra linha de desenvolvimento e segue rumo próprio a partir deste ponto, gerando seu próprio histórico. Um exemplo é exibido na Figura 1.

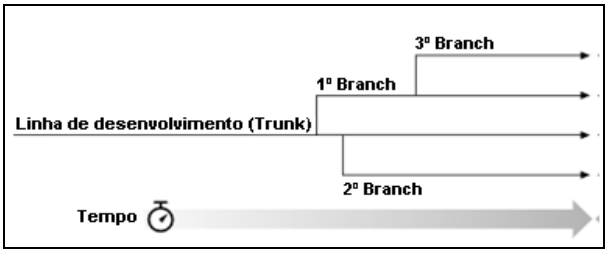

Figura 1 - Ramos de desenvolvimento adaptado de Sussman, Fitzpatrick e Pilato (2011, p. 72).

3.3. Biblioteca Definitiva de Software

A Biblioteca Definitiva de Software é o local onde todas as versões autorizadas e definitivas de software da organização são armazenadas. Ela armazena as cópias-mestras de todos os softwares comprados (junto com os documentos de licenciamento), assim como as dos softwares desenvolvidos internamente (MAGALHÃES; PINHEIRO, 2007).

4. Ferramentas de apoio

De acordo com Sommerville (2003) os processos de gerenciamento de configuração são padronizados e envolvem a aplicação de procedimentos predefinidos. Eles exigem no gerenciamento cuidados de grande quantidade de dados, e a atenção aos detalhes é essencial. Quando se constrói uma nova versão de software, um único erro de gerenciamento de configuração pode significar que o software não funcionará adequadamente. Como consequência, a ferramenta de apoio é essencial para o gerenciamento de configuração.

A seguir são exibidas algumas das ferramentas de apoio para gerência de configuração utilizada no desenvolvimento deste trabalho.

4.1 Subversion (SVN)

O Subversion é um software livre para controle de versão. É utilizado tanto para o desenvolvimento de software livre como para fins corporativos (SUSSMAN, FITZPATRICK, PILATO, 2011).

Tem o objetivo de gerenciar arquivos e diretórios, e as modificações feitas neles ao longo do tempo. Isto permite que você recupere versões antigas de seus dados, ou que examine o histórico de suas alterações (SUSSMAN, FITZPATRICK, PILATO, 2011, p. 17).

O Subversion tem comandos para ajudar a controlar versões paralelas de um arquivo ou diretório. Ele permite você criar ramos copiando seus dados, e ainda lembra que as cópias têm relação entre si. Ainda é possível duplicar cópias de um ramo para outro. Finalmente, ele pode fazer com que partes de sua cópia de trabalho reflitam ramos diferentes, assim é possível "misturar e combinar" diferentes linhas de desenvolvimento no trabalho de dia-a-dia (SUSSMAN, FITZPATRICK, PILATO, 2011, p. 72).

4.2 Redmine

O Redmine é um software livre, gerenciador de projetos baseados na web e ferramenta de gerenciamento de mudança. Ele contém calendário e gráfico de *Gantt* para ajudar na

representação visual dos projetos e seus prazos de entrega. Ele pode também trabalhar com múltiplos projetos (REDMINE, 2012).

5. Metodologia

Para reunir informações e automatizar o processo de liberação, o trabalho desenvolvido faz a integração com um sistema de gestão empresarial, gerenciamento de configuração e gerenciamento de mudança. Utilizando práticas recomendas pelo CMMI a equipe de liberação, inicia o processo de liberação e faz a indicação dos arquivos de atualização. Automaticamente os arquivos indicados são disponibilizados em um FTP pelo servidor de atualizações.

O servidor ainda tem a responsabilidade de autorizar o download dos arquivos solicitado pelos clientes via *webservice*. A partir do momento que a atualização é autorizada, os clientes fazem o download automático dos arquivos. Seguindo recomendações do processo de gerenciamento de liberação definido pelo ITIL, é dado inicio ao processo de atualização. Primeiramente em um servidor de homologação em seguida a atualização é enviada para o servidor de produção. A Figura 2 exibe uma visão geral do trabalho desenvolvido.

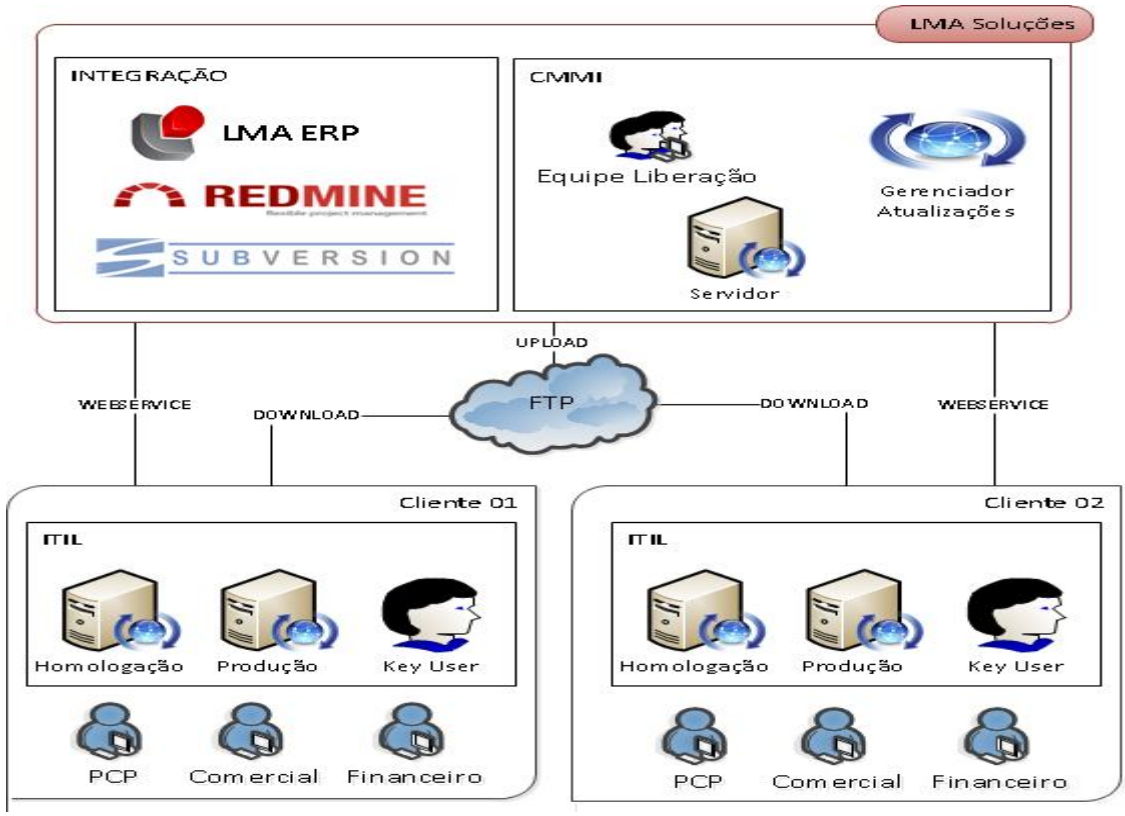

Figura 2 - Visão geral do trabalho desenvolvido

Foram inseridos alguns processos para ampliar o controle sobre as soluções desenvolvidas. A Figura 3 exibe uma visão geral do ciclo de vida de uma requisição de mudança realizada pelo cliente.

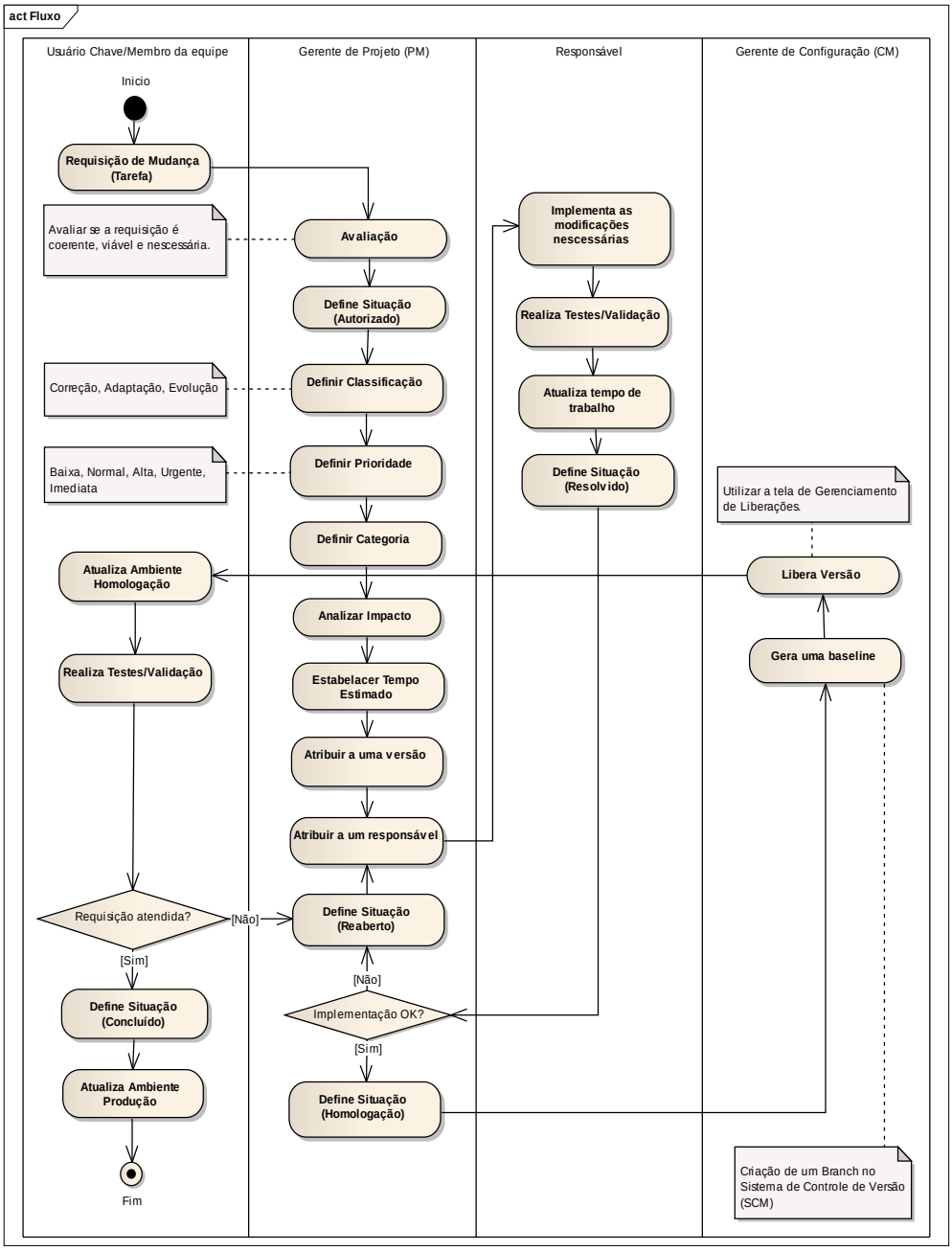

Figura 3 - Fluxo proposto

A partir do momento que o cliente informa a necessidade de uma nova funcionalidade no sistema, o Analista faz o levantamento dos requisitos e regras de negócio para a alteração solicitada. Em seguida é realizada a estimativa de tempo e custo da alteração. Caso a solicitação seja aprovada pelo cliente, ela é registrada no sistema de controle de mudanças (*redmine*) e atribuída a uma equipe ou programador. Com o término do seu desenvolvimento, são realizados testes unitários até que a correção esteja em condições de ser liberada para o cliente.

O objetivo deste processo é garantir que possíveis erros possam ser corrigidos antes de entrar para o ambiente de produção do cliente, fazendo os ajustes necessários antes da sua liberação oficial

Após serem realizados todos os testes de integração a nova versão está pronta para ser liberada. É neste momento que o aplicativo desenvolvido neste trabalho é utilizado. A

equipe responsável pela liberação inicia o processo de liberação através da tela de Gerenciamento de liberações exibida na Figura 4.

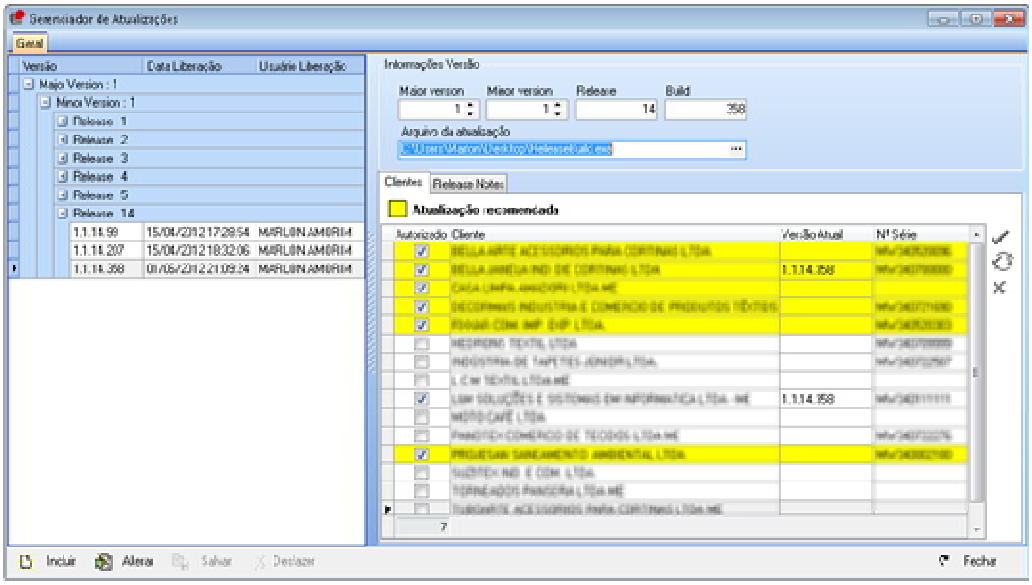

Figura 4 - Tela de gerenciamento de liberações

O aplicativo desenvolvido armazena um histórico com todas as liberações realizadas e os clientes que receberam cada atualização. Ao incluir uma nova liberação é exibida uma lista com todos os clientes envolvidos no projeto, o sistema faz a recomendação dos clientes aptos a receber a nova atualização exibindo-os na cor amarela, como é exibido na Figura 4.

6. Resultados

Neste trabalho foi realizado o desenvolvimento de um aplicativo que automatiza o processo de liberação de software. Auxiliando os membros da equipe de liberação a identificar os clientes aptos a receber a nova atualização, monitorar a versão utilizada por cada cliente, além de manter um histórico de todas as liberações efetuadas pela empresa. O aplicativo ainda faz a integração com o sistema de projetos *redmine*, extraindo informações utilizadas na criação de *release notes* de cada liberação. Para concluir foi adicionado a possiblidade de enviar por e-mail a relação de modificações contidas em cada liberação.

Utilizando conceitos recomendados pelas metodologias ITIL e CMMI o trabalho alcançou todos os seus objetivos, além de auxiliar na elaboração de um novo fluxo de trabalho, que proporciona objetividade e define responsabilidade sobre cada etapa do processo de requisição de mudança, fornecendo um ganho de qualidade e maior controle dos serviços prestados.

As ferramentas e tecnologias utilizadas foram adequadas, atendendo todas as exigências para o sucesso deste projeto.

A realização deste trabalho contribuiu para a expansão dos conhecimentos sobre as metodologias ITIL e CMMI, que foram elementos essenciais para guiar o objetivo deste trabalho. Além disso, novas oportunidades foram identificadas, despertando o interesse pelas demais áreas tratadas por estas metodologias.

Referências

Axelos. Axelos. [S.l.],(2014). http://www.axelos.com/.

Beca, Advice. (2012) "FITS Release Management". [S.I.], 2004. http://www.becta.org.uk/tsas/docs/fits_release.pdf, Agust.

Bon, Jan V. (2006) "Fundamentos do gerenciamento de serviços em TI baseado no ITIL"**.** 1. ed. Tradução Van Haren Publishing. Holanda.

Couto, Ana Brasil. (2007) "CMMI: integração dos modelos de capacitação e maturidade de sistemas". Rio de Janeiro: Ciência Moderna, 2007. XIII, 276 p.

Estublier, J.(2000) Software Configuration Management: Proceedings of the Conference on The Future of Software Engineering. Ireland: Limerick,.

Magalhães, Ivan Luizio; Pinheiro, Walfrido Brito. (2007) "Gerenciamento de serviços de TI na prática: uma abordagem com base na ITIL". São Paulo: Novatec.

Mellon, Carnegie. (2012) "CMMI para desenvolvimento – Versão 1.2." [S.l.], 2006. http://www.sei.cmu.edu/library/, December.

Redmine. Redmine. [S.l.],(2012). http://www.redmine.org, may.

Sommerville, Ian.(2003) "Engenharia de software". 6. ed. São Paulo: Addison Wesley, 2003. XVI, 592 p.

Sussman, Ben; Fitzpatrick, Brian; Pilato, C. (2011) "Version control with subversion: For subversion" 1.7. California: TBA, 2011. 441 p.