

**Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Informática e Estatística
Graduação em Sistemas de Informação**

**Implementação de Uma Ferramenta de Apoio a
Gerência de Configuração do MPS.br**

Henrique Lima Leite

Florianópolis, março de 2010

HENRIQUE LIMA LEITE

**Implementação de Uma Ferramenta de Apoio a
Gerência de Configuração do MPS.br**

Trabalho apresentado ao Curso de
Sistemas de Informação da Universidade
Federal de Santa Catarina como parte
dos requisitos para obtenção do título de
Bacharel em Sistemas de Informação.
Prof. Dr. Ricardo Pereira Silva – orientador

Florianópolis, março de 2010

“Há mais coisas entre o céu e a terra do que supõe nossa vã filosofia”

Willian Shakespeare

SUMÁRIO

Lista de Siglas.....	8
Lista de Figuras	9
Lista de Tabelas	11
Resumo.....	12
Abstract.....	13
1 Introdução.....	14
1.1 Objetivos	14
1.1.1 Objetivo Geral	14
1.1.2 Objetivos Específicos.....	15
1.1.3 Escopo e delimitação do trabalho.....	15
1.2 Metodologia.....	16
2 Fundamentação Teórica.....	16
2.1 Processo de Software.....	17
2.2 Modelos de Melhoria de Software	17
2.2.1 ISO/IEC 15504.....	18
2.2.2 CMMI-DEV V1.2	22
2.2.3 MPS.BR.....	26

2.3	Gerência de Configuração	29
2.3.1	Itens de configuração.....	31
2.3.2	Controle de mudança.....	31
2.3.3	Controle de Versão	32
2.3.4	Baseline	33
2.3.5	Auditoria de Configuração.....	34
2.4	VSS – Visual Source Safe	35
3	A Empresa H2L Solutions	37
4	A ferramenta FAC.....	38
4.1	Requisitos	38
4.1.1	Requisitos funcionais	39
4.1.2	Requisitos não funcionais	40
4.2	Workflow da FAQ.....	40
4.3	Casos de Uso.....	42
4.3.1	Caso de uso – Cadastrar Item de Configuração	43
4.3.2	Caso de uso – Cadastrar Baseline	43
4.3.3	Caso de uso – Gerar Baseline	44
4.3.4	Caso de uso – Cadastrar Projetos.....	46

4.3.5	Caso de uso – Recuperar Baseline	47
4.3.6	Caso de uso – Cadastrar Integrantes	48
4.3.7	Caso de uso – Solicitar Mudança	48
4.3.8	Caso de uso – Avaliar Mudança	49
4.3.9	Caso de uso – Auditar	52
4.4	Diagrama de Implantação (Deployment)	52
4.5	Comunicação com a API do Visual Source Safe	54
5	Conclusão	55
5.1	Resultados obtidos	56
5.2	Limitações	56
5.3	Trabalhos futuros	57
5.4	Considerações finais	57
6	Referencias Bibliográficas	58

Lista de Siglas

API	Application Programming Interface
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CVS	Concurrent Version System
GCS	Gestão de Configuração de Software
HTTPs	HyperText Transfer Protocol Secure
ISO	International Standards Organization
PDF	Portable Document Format
MPS.BR	Melhoria de Processo de Software Brasileiro
RIA	Rich Internet Application
SEI	Software Engineering Institute
WWW	World Wide Web

Lista de Figuras

Figura 1 - Perfil da capacitação de área de processo CMMI (PRESSMAN, 2007)	25
Figura 2 - Componentes do Modelo MPS (SOFTEX, 2009)	27
Figura 3 - Processos e níveis de maturidade (SOFTEX, 2009).....	28
Figura 4 - Entendendo uma Baseline (MOLINARI, 2007).....	34
Figura 5 - Tela inicial da ferramenta FAC.....	38
Figura 6 - Workflow da FAC.....	41
Figura 7 - Diagrama geral de casos de uso	42
Figura 8 - Tela de cadastro de item de configuração	43
Figura 9 - Tela de cadastro de baseline	44
Figura 10 - Tela de criação de baseline.....	44
Figura 11 - Relatório da geração de baseline.....	45
Figura 12 - Diagrama de seqüência da geração de baseline.....	46
Figura 13 - Tela de cadastro de projetos.....	47
Figura 14 - Tela de recuperação de baseline	47
Figura 15 - Tela de cadastro de integrantes	48
Figura 16 - Tela de solicitação de mudança	49

Figura 17 - Tela de lista de solicitação de mudança.....	50
Figura 18 - Tela de aceitação de solicitação de mudança.....	50
Figura 19 - Diagrama de estados da solicitação e aprovação de mudança.....	51
Figura 20 - Tela de auditoria de configuração	52
Figura 21 - Diagrama de implantação.....	54
Figura 22 - Classe de Conexão com o VSS	55

Lista de Tabelas

Tabela 1: Atributos de processo e níveis de capacidade (ISO, 2005).....	20
---	----

Resumo

O Processo de Gerência de Configuração é de extrema importância para o ciclo de desenvolvimento de software. Garantir a consistência de uma entrega ao cliente e posteriormente poder rastrear os itens de configuração que a compõem. Ainda poder gerenciar as mudanças nesses artefatos de forma eficiente. E, por fim, auditar esses mecanismos a fim de garantir que o processo como um todo foi executado com qualidade. Essas são as tarefas da gerência de configuração, e ainda há integração com ferramentas de mercado como integração contínua e repositório de dados.

Nesse contexto é proposta uma ferramenta chamada *FAC (Ferramenta de Automação de Gerência de Configuração)* que contempla um suporte para o processo de gerência de configuração, com a possibilidade de geração de baselines, controle de mudanças e auditoria de configuração.

A ferramenta FAC utiliza-se da API provida pela ferramenta de controle de versão VSS (Visual Source Safe) para realizar a leitura dos itens de configuração e suas versões, liberar ou restringir permissões para desenvolvedores e por fim emitir relatórios de auditoria.

Abstract

The configuration management process is extremely important for the software lifecycle development. Ensuring consistency of delivery to the customer and then be able to track the configuration items that comprise it. Still being able to manage changes in these artifacts efficiently. Finally, audit these mechanisms in order to ensure that the whole process was executed with quality. These are the tasks of configuration management, and there's integration with marketing tools such as continuous integration and data repository.

In this context we propose a tool called FAC (Tool Automation Configuration Management) which includes support for the process of configuration management, with the possibility of generation of baselines, change control and configuration auditing.

FAC The tool uses the API provided by the tool version control VSS (Visual Source Safe) for reading the configuration items and their versions, release or restrict permissions to developers and ultimately issue an audit report.

1 Introdução

Um modelo de maturidade tem o propósito de prover um guia para a melhoria dos processos de uma organização, da sua habilidade de gerenciar o desenvolvimento, a aquisição e a manutenção de produtos e serviços (FREDERICO, 2007)(SEI, 2006).

Dentre os processos de um modelo de maturidade, é estudado o processo de Gerência de Configuração, e é proposta uma ferramenta para a automatização desse processo tomado como base o modelo MPS.br.

O processo de Gerência de Configuração tem com finalidade garantir a integridade dos produtos de software, através do controle efetivo do processo de desenvolvimento. A configuração do software compreende todos os itens de configuração que são obtidos durante o seu ciclo de vida. O estabelecimento e a manutenção da integridade desses itens de configuração constituem o processo de gerenciamento da configuração de software. (BARBARESCO, 2000).

Este capítulo apresenta o modo operante deste trabalho e a implementação da ferramenta chamada FAC (*Ferramenta de Automatização de Gerência de Configuração*). São apresentados neste capítulo: objetivos, resultados esperados e metodologia deste trabalho.

1.1 Objetivos

A partir da apresentação do contexto do problema, são definidos os objetivos gerais e específicos deste trabalho:

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é o desenvolvimento de uma ferramenta para automatizar o processo de gerência de configuração do modelo MPS.BR

e ao final espera-se que a ferramenta atenda a todos os resultados esperados do processo de gerência de configuração do MPS.BR.

1.1.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

Objetivo 1: Detalhar modelos que antecederam e deram subsídios ao MPS.BR, como CMMI e ISO/IEC 15504.

Objetivo 2: Elucidar os conceitos de processos de software, modelo de melhoria de software e gerência de configuração.

Objetivo 3: Explanar alguns conceitos relacionados à gerência de configuração, por exemplo: Itens de configuração, controle de versão, controle de mudança, baseline e auditoria de configuração.

Objetivo 4: Criar uma ferramenta em ambiente 100% Web, utilizando conceito de RIA (Rich Internet Application), atendendo alguns requisitos não funcionais como escalabilidade e segurança.

Objetivo 5: integrar a ferramenta com a API da aplicação Visual Source Safe.

1.1.3 Escopo e delimitação do trabalho

O escopo do trabalho limita-se a estudar o conceito de gerência de configuração aplicado ao MPS.BR, analisando também outros modelos de referência que deram subsídio à criação do MPS.BR. Também são apresentados conceitos que compõem a gerência de configuração.

A ferramenta proposta se integrara com a ferramenta Visual Source Safe, esta proposta não contempla integração com outras ferramentas de controle de versão como por exemplo : Subversion ou CVS.

Espera-se que os requisitos do processo de gerência de configuração do MPS.BR sejam atendidos em sua totalidade.

1.2 Metodologia

Este trabalho está organizado em cinco capítulos. Os três primeiros apresentam a introdução e a revisão teórica dos principais conceitos relacionados ao trabalho. Os dois restantes tratam dos resultados do trabalho realizado.

O capítulo 2 introduz o conceito de gerência de configuração, aborda modelos de melhoria como o CMMI-DEV v1.2, ISO-IEC 15504 e MPS.BR.

O capítulo 3 apresenta a empresa ou laboratório que implantará a ferramenta proposta nesse trabalho.

O capítulo 4 apresenta a ferramenta proposta (*FAC*), e a documentação relacionada em UML: diagrama de implantação, requisitos, máquina de estado e casos de uso.

O capítulo 5 apresenta as conclusões, limitações, sugestões para trabalhos futuros e as considerações finais.

2 Fundamentação Teórica

Antes que se possa definir uma proposta para implementação de uma ferramenta para automatizar o processo de Gerência de Configuração, é importante que sejam estudados os conceitos envolvidos. Por isso, neste capítulo são descritos os conceitos básicos necessários ao entendimento do processo de Gerência de Configuração.

2.1 Processo de Software

Segundo [ABNT, 2001]. Processo é um conjunto de atividades inter-relacionadas ou interativas, que transforma insumos (entradas) em produtos (saídas).

(PRESSMAN, 2007) define processo de software como: Um arcabouço para as tarefas que são necessárias para construir softwares de alta qualidade. Processo de software define a abordagem que é adotada quando o software é elaborado.

A escolha de um modelo de processo (ou modelo de ciclo de vida) é o ponto inicial para a definição de um processo de desenvolvimento de software. Um modelo de processo preocupa-se com atividades macros, e estabelece precedência e dependência entre as mesmas. Os processos de software definem o conjunto de atividades conduzidas no contexto do projeto, por exemplo: análise e gestão dos requisitos, planejamento de projeto, desenvolvimento, planejamento e execução de testes e implantação. (PRESSMAN, 2007).

2.2 Modelos de Melhoria de Software

A evolução da engenharia de software resultou no estabelecimento de normas, modelos e guias, como por exemplo: (ANSI, 1998; PMI, 2004; SWEBOK, 2004; ISO/IEC, 2005; SEI, 2006; SOFTEX, 2009), que auxiliam o gerente de projeto a adaptar os conceitos e práticas de gerência de projetos à realidade e desafios da gerência de projetos de software. (HAUKE, 2007)

Neste capítulo alguns modelos com maior expressividade serão apresentados, são eles: CMMI-DEV v1.2 (SEI, 2006), MPS.BR (SOFTEX, 2009) e ISO/IEC.

2.2.1 ISO/IEC 15504

É uma norma elaborada pela ISO (International Organization for Standardization) e pelo IEC (International Electrotechnical Commission), e constitui-se de um padrão para a avaliação do processo de software, objetivando determinar a capacitação de uma organização. (EMA, 1998) (HAUKE, 2007)

Os objetivos da norma ISO/IEC 15504 são: (ISO, 2005) (HAUKE, 2007)

1. Melhoria dos processos: gerando um perfil dos processos, identificando os pontos fracos e fortes que serão utilizados para a elaboração de um plano de melhorias;
2. Determinação da capacidade dos processos: viabilizando a avaliação de um fornecedor em potencial, obtendo o seu perfil de capacidade.

Esta norma pode ser utilizada tanto por uma organização que deseja avaliar a capacidade de possíveis fornecedores de produtos de software, quanto por organizações que desejam determinar a capacidade de seus processos (ISO, 2005).

O modelo de referência possui duas dimensões: (1) Capacidade e (2) Processo.

(1) Capacidades são divididas em níveis de capacidade de processo, determinado pelo modelo (ISO, 2005):

Nível 0 (incompleto): o processo não é executado nem consegue alcançar a sua finalidade. Neste nível não há quase nenhuma evidência de uma realização sistemática da finalidade do processo.

Nível 1 (executado): o processo é executado e consegue alcançar sua finalidade.

Nível 2 (gerenciado): o processo é executado de uma forma controlada (de forma planejada, monitorada e ajustada) e seus produtos de trabalho são apropriadamente estabelecidos, controlados e mantidos.

Nível 3 (estabelecido): o processo controlado é executado utilizando um processo definido, baseado em um processo padrão, que seja capaz de alcançar os resultados esperados.

Nível 4 (previsível): o processo estabelecido opera-se agora dentro dos limites definidos para conseguir seus resultados esperados.

Nível 5 (otimização): o processo previsível é melhorado continuamente para alcançar os objetivos de negócio relevantes atuais e projetados.

(2) A norma ISO/IEC 15504 possui nove atributos de processo para avaliar a capacidade dos processos. A função desses atributos é para determinar se um processo atingiu um nível determinado de capacidade.

Atributo de Processo	Níveis de Capacidade e Atributos de Processo
	Nível 0: Processo Incompleto
	Nível 1: Processo Executado
PA 1.1	Desempenho do Processo
	Nível 2: Processo Gerenciado
PA 2.1	Gerenciamento do Desempenho do Processo
PA 2.2	Gerenciamento dos Produtos de Trabalho

	Nível 3: Processo Estabelecido
PA 3.1	Definição do Processo
PA 3.2	Distribuição do Processo
	Nível 4: Processo Previsível
PA 4.1	Mensuração do Processo
PA 4.2	Controle do Processo
	Nível 5: Processo Otimizado
PA 5.1	Inovação do Processo
PA 5.2	Otimização Contínua

Tabela 1: Atributos de processo e níveis de capacidade (ISO, 2005)

2.2.1.1 Gerência de Configuração na ISO/IEC 15504

A norma ISO/IEC 12207, agrupa as atividades e tarefas que podem ser executadas durante o ciclo de vida do software em processos fundamentais, processos de apoio e processos organizacionais. (BARBARESCO, 2000).

Segundo (BARBARESCO, 2000). Dentre os processos de apoio existentes, a [ABN1997] descreve o processo de gerência da configuração como um processo de aplicação de procedimentos administrativos e técnicos, por todo o ciclo de vida de software, destinado a:

(1) Identificar e definir os itens de software em um sistema, e estabelecer suas linhas básicas (baseline);

(2) Controlar as modificações e liberações dos itens;

(3) Registrar e apresentar a situação dos itens e dos pedidos de modificação;

(4) Garantir a consistência e a correção dos itens;

(5) Controlar o armazenamento, a manipulação e a distribuição dos itens. Sendo assim neste processo haverá então a execução das seguintes atividades:

a) Implementação do Processo – Nesta atividade um plano de gerência da configuração deve ser desenvolvido. O plano deverá descrever: as atividades da gerência da configuração; procedimentos e cronograma para executar estas atividades; as organizações responsáveis pela execução destas atividades; e seu relacionamento com outras organizações, como por exemplo a de desenvolvimento ou manutenção de software. O plano deve ser documentado e implementado. Neste aspecto o plano poderá fazer parte do plano de gerência da configuração do sistema.

b) Identificação da Configuração – Neste item, uma sistemática para o projeto deve ser estabelecida para a identificação dos itens de software e suas versões a serem controladas. Para cada item de software e suas versões deve ser identificado o seguinte: a documentação que estabelece a linha básica (baseline); as referências de versão; e outros detalhes de identificação.

c) Controle da Configuração – Aqui devem ser executadas as seguintes diretivas: identificação e registro dos pedidos de alteração; análise e avaliação das alterações; aprovação ou rejeição do pedido; e implementação, verificação e liberação do item de software modificado. Devem existir registros de

auditoria, de tal forma que, para cada modificação, a sua razão e a sua autorização possam ser rastreadas. Devem ser realizados controle e auditoria de todos os acessos aos itens de software controlados que tratam de funções críticas de proteção ou segurança.

d) Relato da situação da configuração – Nesta etapa devem ser preparados registros de gerenciamento e relatórios de situação que mostrem a situação e o histórico dos itens de software controlados, incluindo a linha básica (baseline). Os relatórios de situação deveriam incluir o número de alterações em um projeto, as últimas versões do item de software, identificadores de liberação, a quantidade de liberações e as comparações entre elas.

e) Avaliação da configuração – Deve ser determinado e garantido o seguinte: a completeza funcional dos itens de software em relação aos seus requisitos e a completeza física dos itens de software (ou seja, se seu projeto e código refletem uma descrição técnica atualizada).

f) Gerência de liberação e distribuição – Nesta última etapa a liberação e distribuição de produtos de software e documentação devem ser formalmente controladas. Cópias matrizes do código e da documentação devem ser mantidas durante a vida do produto de software. O código e a documentação que contenham funções críticas de proteção ou segurança devem ser manipulados, armazenados, empacotados e distribuídos de acordo com as políticas das organizações envolvidas.

2.2.2 CMMI-DEV V1.2

O CMMI- Capability Maturity Model Integration (SEI, 2006) é um modelo para a melhoria de processos de desenvolvimento de produtos e serviços, composto pelas melhores práticas, que cobre todo o ciclo de vida de um produto, desde a sua concepção até a sua entrega e a posterior manutenção.

O objetivo do CMMI é auxiliar organizações a melhorar seus processos de desenvolvimento e manutenção de produtos e serviço. (HAUKE, 2007).

O CMMI surgiu da necessidade de integrar os diversos CMM existentes em uma única estrutura, com os mesmos processos de avaliação e terminologia. O CMMI é aderente a norma ISO 15504, igualando o modelo a outros modelos que seguem esta norma e possibilitando a equivalência nas avaliações.

O SEI desenvolveu um abrangente metamodelo de processo baseado em um conjunto de capacidades de engenharia de software que devem estar presentes à medida que as empresas alcançam diferentes níveis de capacidade e maturidade de processo. Para conseguir essas capacidades, o SEI alega que uma organização deve desenvolver um modelo de processo que siga as diretrizes do CMMI (PRESSMAN, 2006).

Segundo (PRESSMAN, 2006) O CMMI representa um metamodelo de processo de dois modos diferentes: (1) como um modelo contínuo e (2) com um modelo em estágios. O metamodelo CMMI contínuo descreve um processo em duas dimensões como a Figura 1. Cada área de processo (por exemplo, planejamento de projeto ou gestão de requisitos) é avaliada formalmente com base em metas e práticas específicas, e é classificada de acordo com os seguintes níveis de capacitação:

Nível 0: Incompleto. A área de processo não é realizada ou não atinge todas as metas e objetivos definidos pelo CMMI para o nível 1 de capacitação.

Nível 1: Realizado. Todas as metas específicas da área de processo definidas pelo CMMI foram satisfeitas. As tarefas de trabalho necessárias para produzir os produtos de trabalho definidos estão sendo conduzidas.

Nível 2: Gerido. Todos os critérios do nível 1 foram satisfeitos. Além disso, todo trabalho associado à área de processo está de acordo com a

política definida pela organização; todo o pessoal que está fazendo o trabalho tem acesso aos recursos adequados para fazer o serviço; os interessados estão ativamente envolvidos nas áreas de processo, conforme necessário; todas as tarefas e produtos de trabalho são “monitorados, controlados e revisados e são avaliados quanto à aderência à descrição do processo”.

Nível 3: Definido. Todos os critérios do nível 2 foram alcançados. Além disso, o processo é “feito sob medida para o conjunto-padrão de processos da organização, de acordo com as suas diretrizes quanto a fazer coisas sob medida e contribui com produtos de trabalho, medições e outras informações de aperfeiçoamento de processo para o patrimônio de processos da organização”.

Nível 4: Quantitativamente gerido. Todos os critérios do nível 3 foram alcançados. Além disso, a área de processo é controlada e aperfeiçoada usando medições e avaliação quantitativa. “Objetivos quantitativos para qualidade e desempenho de processo são estabelecidos e usados como critério na gestão do processo”.

Nível 5: Otimizado. Todas as capacidades do nível 4 foram alcançadas. Além disso, a área de processo é adaptada e otimizada usando meios quantitativos (estatísticos) para satisfazer às alterações de necessidades do cliente e continuamente aperfeiçoar a eficácia da área de processo em consideração.

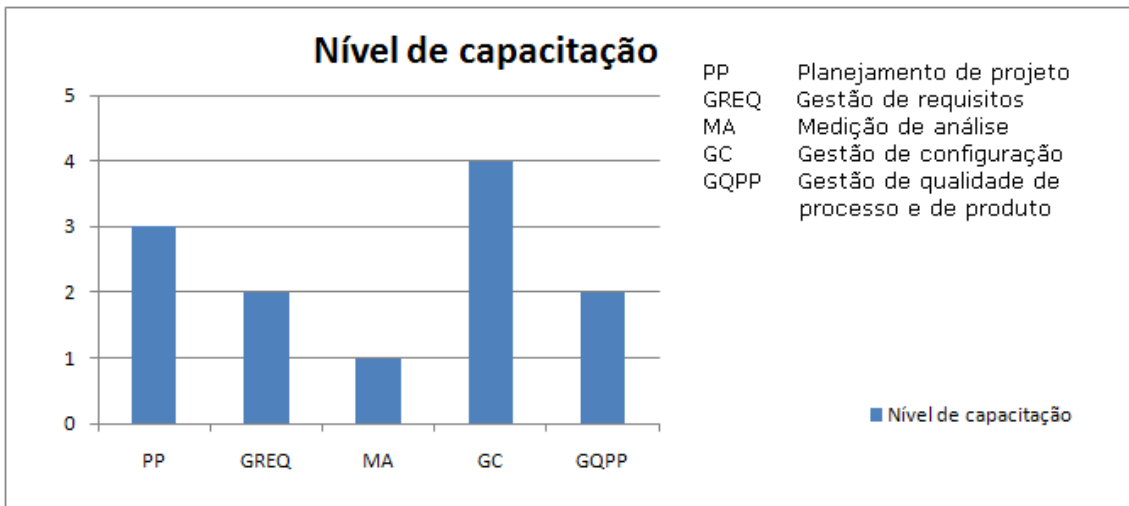


Figura 1 - Perfil da capacitação de área de processo CMMI (PRESSMAN, 2007)

2.2.2.1 Gerência de configuração no CMMI

Segundo (SEI, 2006) o propósito da GCS é estabelecer e manter a integridade dos produtos de trabalho, usando identificação da configuração, controle de configuração, status da configuração contábil e auditoria de configuração.

As metas específicas de gestão de configuração definidas no CMMI segundo (MOLINARIA, 2007) são:

1. Estabelecer baselines, que são utilizadas para o trabalho nos produtos bem como sua manutenção.
2. Rastreamento e controle de mudanças, para permitir a rastreabilidade e controle das mudanças dos produtos.
3. Estabelecer integridade, para permitir que a integridade das baselines seja estabelecida e mantida.

2.2.3 MPS.BR

O MPS.BR é um programa para melhoria de Processo do Software Brasileiro. Iniciou seu desenvolvimento em dezembro de 2003 e é coordenado pela Associação para Promoção da Excelência do Software Brasileiro (MOLINARI, 2007).

O Modelo surgiu pelo fato de muitas empresas estarem buscando certificação CMMI, mas, quando se deparavam com os custos, somente as grandes empresas conseguiam bancar uma real avaliação. O MPS surgiu com intuito de adaptar a realidade do mercado brasileiro para suprir a lacuna nas pequenas e médias empresas. (MOLINARI, 2007).

Segundo (SOFTEX, 2009), o programa conta com duas metas a alcançar a médio e longo prazo:

a) meta técnica, visando à criação e aprimoramento do modelo MPS, com resultados esperados tais como: guias do modelo MPS, instituições implementadoras, instituições avaliadoras (IA), consultores de aquisição.

b) meta de mercado, visando a disseminação e adoção do modelo MPS, em todas as regiões do país, em um intervalo de tempo justo, a um custo razoável.

O Modelo MPS possui 4 guias que norteiam a sua execução:

a) Guia Geral, responsável por fornecer uma visão geral sobre os demais guias que apóiam os diversos níveis do MPS. (SOFTEX, 2009).

b) Guia de Avaliação, descreve o processo e o método de avaliação MA-MPS. (SOFTEX, 2009).

c) Guia de Aquisição, descreve um processo de aquisição de software e serviços correlatos. (SOFTEX, 2009).

d) Guia de Implementação, série de dez documentos que fornecem orientações para implementar nas organizações os níveis de maturidade. (SOFTEX, 2009).

O Modelo MPS estabelece um modelo de processos de software e um processo e um método de avaliação de processos. A base técnica para a construção deste modelo de melhoria é composta pelas normas ISO/IEC 12207:2008 e ISO/IEC 15504 e o modelo é aderente ou compatível com o modelo do CMM/CMMI. O Modelo está dividido em três componentes: Modelo de Referência (MR-MPS), Método de Avaliação (MA-MPS) e Modelo de Negócio (MN-MPS). (SOFTEX, 2009), como ilustrado na Figura 2.

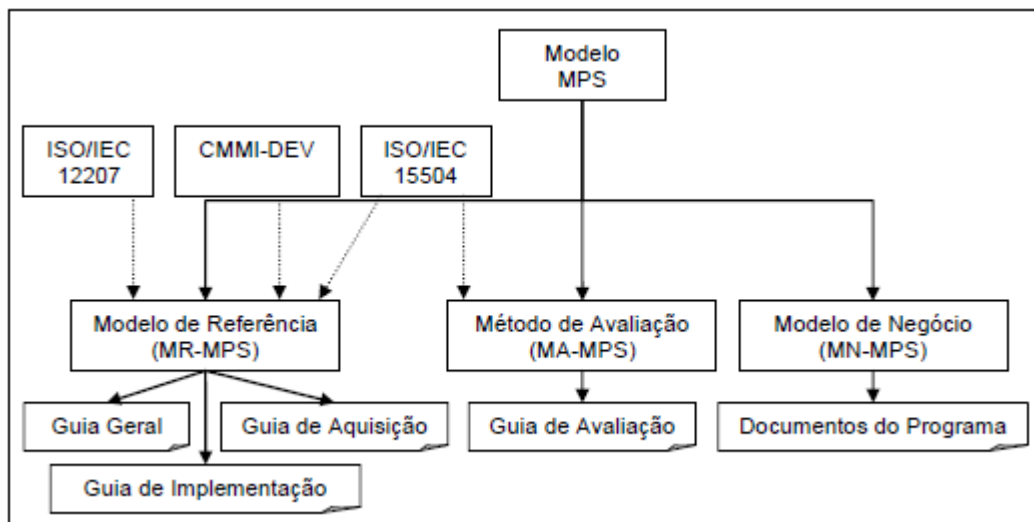


Figura 2 - Componentes do Modelo MPS (SOFTEX, 2009)

No MPS.BR a avaliação de processos segue o modelo por estágios, onde toda organização é avaliada segundo uma série de processos que estão agrupados em níveis de maturidade. Para isso, o MPS.BR define sete níveis de maturidade, ordenado de “A” a “G”, sendo o nível mais alto, conforme demonstra a figura abaixo. A divisão por estágios é inspirada no CMMI, mas possui mais níveis, com objetivo de possibilitar à micro e pequenas empresas de software uma implementação facilitada e com obtenção de resultados em prazos mais curtos (HAUCK, 2007). A Figura 3 ilustra a divisão por níveis.

Nível	Processos	Atributos de Processo
A		AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2, AP 5.1 e AP 5.2
B	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2, AP 4.1 e AP 4.2
C	Gerência de Riscos – GRI Desenvolvimento para Reutilização – DRU Gerência de Decisões – GDE	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
D	Verificação – VER	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Validação – VAL	
	Projeto e Construção do Produto – PCP	
	Integração do Produto – ITP	
	Desenvolvimento de Requisitos – DRE	
E	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Gerência de Reutilização – GRU	
	Gerência de Recursos Humanos – GRH	
	Definição do Processo Organizacional – DFP	
	Avaliação e Melhoria do Processo Organizacional – AMP	
F	Medição – MED	AP 1.1, AP 2.1 e AP 2.2
	Garantia da Qualidade – GQA	
	Gerência de Portfólio de Projetos – GPP	
	Gerência de Configuração – GCO	
	Aquisição – AQU	
G	Gerência de Requisitos – GRE	AP 1.1 e AP 2.1
	Gerência de Projetos – GPR	

Figura 3 - Processos e níveis de maturidade (SOFTEX, 2009)

2.2.3.1 Gerência de Configuração no MPS.BR

O propósito do processo Gerência de Configuração é estabelecer e manter a integridade de todos os produtos de trabalho de um processo ou projeto e disponibilizá-los a todos os envolvidos (SOFTEX, 2009).

Como a gerência de configuração é um processo para o MPS.BR, ela espera que esse processo tenha resultados esperados, são eles: (SOFTEX, 2009).

GC01 – Um Sistema de Gerência de Configuração é estabelecido e mantido.

GC02 – Os itens de configuração são identificados com base em critérios estabelecidos.

GC03 - Os itens de configuração sujeitos a um controle formal são colocados sob baseline.

GC04 – A situação dos itens de configuração e das baselines é registrada ao longo do tempo e disponibilizada.

GC05 – Modificações em itens de configuração são controladas

GC06 – O armazenamento, o manuseio e a liberação de itens de configuração e baselines são controlados.

GC07 – Auditorias de configuração são realizadas objetivamente para assegurar que as baselines e os itens de configuração estejam íntegros, completos e consistentes.

2.3 Gerência de Configuração

A Gerência de Configuração é a disciplina responsável por controlar a evolução de sistemas de software. Apesar de existir um forte apelo para o uso da Gerência de Configuração durante a etapa de manutenção, a sua aplicação não se restringe somente a essa etapa do ciclo de vida do software. Durante o desenvolvimento, o sistema de Gerência de Configuração é fundamental para prover controle sobre os produtos de trabalho produzidos e modificados por diferentes engenheiros de software. Além disso, esse sistema possibilita um

acompanhamento minucioso do andamento das tarefas de desenvolvimento. (SOFTEX, 2009).

Segundo (PRESSMAN, 2007) a gerência de configuração é uma atividade guarda-chuva que é aplicada ao longo de todo o processo de software. Como modificações podem ocorrer em qualquer época, as atividades GCS são desenvolvidas para (1) identificar modificações, (2) controlar modificações, (3) garantir que as modificações sejam adequadamente implementadas e (4) relatar as modificações a outros que possam ter interesse. Pode-se dizer que a gerência de configuração é um conjunto de atividades de acompanhamento e controle que começam quando o projeto de engenharia de software tem início e só terminam quando o software é retirado de operação.

(DAR, 2001) Identificou quatro elementos que devem estar presentes em um sistema GCS, são eles:

(1) Elementos de Processo, uma coleção de procedimentos e tarefas que definem uma abordagem efetiva para a gerência de configuração e atividades relacionadas para todas as partes envolvidas.

(2) Elementos de Componente, um conjunto de ferramentas acoplado a um sistema de gestão de arquivos (exemplo: banco de dados) que possibilita acesso e gestão de cada item de configuração.

(3) Elementos de construção, um conjunto de ferramentas que automatizam a construção de software garantindo que o conjunto adequado de componentes validados foi montado.

(4) Elementos humanos, para implementar GCS efetiva, a equipe de software usa um conjunto de ferramentas e características de processo.

Esses elementos serão discutidos com mais detalhes nas próximas seções.

2.3.1 Itens de configuração

Um item de configuração é o menor item de controle num processo de GCS, ele pode ser qualquer coisa: um arquivo executável, uma aplicação corporativa, documento, cronograma etc. É ele que será controlado, e todas as informações a seu respeito. (MOLINARI, 2007).

Os itens de configuração podem ser identificados e posteriormente catalogados no banco de dados do projeto (PRESSMAN, 2007).

A Identificação é simplesmente conhecer os atributos básicos do item que são: data de entrada em produção, data de criação, responsável pela criação e o relacionamento com outros itens de configuração, esses atributos também são conhecidos com metadados do item de configuração. (MOLINARI, 2007).

2.3.2 Controle de mudança

No desenvolvimento e manutenção de um produto, a mudança é inevitável. Mudar é uma constante, pessoas cometem erros, clientes necessitam de mudanças no produto para seu uso e o próprio ambiente no qual está inserido o produto precisa de algum ajuste. É comum dizer que uma nova solução trará no futuro novos problemas, que, por sua vez, trarão novas mudanças. (MOLINARI, 2007).

Para um projeto grande de engenharia de software, modificações sem controle levam rapidamente ao caos. Para tais projetos, o controle de modificação combina procedimentos humanos e ferramentas automatizadas. Um pedido de modificação é apresentado e avaliado para determinar o grau de impacto ou efeitos colaterais em potencial em outros objetos da configuração e funções do sistema, e o custo projetado da modificação. Os resultados da avaliação são apresentados em um relatório de modificação. Uma ordem de modificação é gerada para cada modificação aprovada, nela

constam as restrições que precisam ser respeitadas e os critérios de revisão e auditoria. (PRESSMAN, 2007).

Os objetos a serem modificados serão controlados pelo desenvolvedor que está fazendo a modificação. Um sistema de controle de versão atualiza o arquivo original quando a modificação estiver feita. Esses novos objetos através de mecanismos adequados de controle de versão são usados para criar a versão seguinte do software. (PRESSMAN, 2007).

Esses mecanismos de controle de versão, integrados com o processo de controle de modificação, implementam dois elementos importantes do controle de modificação: controle de acesso e controle de sincronização. Controle de acesso determina quais desenvolvedores têm autoridade para ter acesso e modificar um determinado objeto de configuração. Controle de sincronização ajuda a garantir que modificações paralelas, realizadas por duas pessoas diferentes, não se superpõem. (PRESSMAN, 2007).

2.3.3 Controle de Versão

Segundo (MOLINARI, 2007). Controle de Versão pode ser definido como responsável por identificar e diferenciar itens armazenados e gerir o item de configuração ao longo do seu ciclo de vida.

Controle de versão combina procedimentos e ferramentas para gerir diferentes versões de objetos de configuração, que são criados durante o processo de software. Um sistema de controle de versão implementa ou está diretamente integrado com três capacidades principais: (1) um banco de dados de projeto (repositório) que guarda todos os objetos de configuração relevantes; (2) uma capacidade de gestão de versão que guarda todas as versões de um objeto de configuração; (3) facilidade de construir que permite ao desenvolvedor coletar todos os objetos de configuração relevantes e construir uma versão específica do software. Também estabelecem um

conjunto de modificações ou uma coleção de todas as modificações que são necessárias para criar uma versão específica do software. (PRESSMAN 2007).

Um exemplo de sistema de controle de versão é o VSS – Visual Source Safe de fabricação da Microsoft e que será utilizado na integração com a ferramenta FAC.

2.3.4 Baseline

Segundo (BARBARESCO, 2000) considera-se uma baseline um marco de referência no desenvolvimento de software. A entrega de um item de configuração ou vários, depois de realizada uma revisão formal por uma equipe de projeto, geram uma baseline, ou seja a partir do momento que todas as partes da especificação forem revisadas, corrigidas e depois aprovadas, qualquer modificação que se faça necessária deverá de ser avaliada e aprovada.

(MOLINARI, 2007) define baseline como; conjunto de um ou mais itens de configuração identificados e liberados para uso independente de suas versões. A baseline pode ser gerada a qualquer momento e ela é útil para sincronizar um produto ou algo a ser entregue ou desenvolvido ou ainda recuperar uma determinada entrega ao cliente. Muitas vezes, itens individuais de sistema são alterados de forma isolada, gerando versões específicas. Uma vez criada uma baseline, ela não deve ser alterada. Em outras palavras, a baseline é uma “foto” do estado da última versão dos itens de configuração. A Figura 4 ilustra um exemplo de composição de baseline.

Item de Configuração	V 1.1	V 1.2	V 1.3	V 1.4	V 1.5	Baseline A1	Última Versão de Cada Item de Configuração
IC01	?	?					V 1.2
IC02	?	?	?				V 1.3
IC03	?						V 1.1
IC04	?	?	?	?	?		V 1.5
IC05	?	?	?	?			V 1.4
IC06	?	?					V 1.2
IC07	?						V 1.1
IC08	?	?	?	?			V 1.4
IC09	?	?					V 1.2

Figura 4 - Entendendo uma Baseline (MOLINARI, 2007)

2.3.5 Auditoria de Configuração

Segundo (Pressman, 2007), Auditoria de configuração está ligada a verificação de que uma baseline ou solicitação de mudança foi efetivamente implementada com sucesso. Para isso a auditoria de configuração envolve dois aspectos (1) revisões técnicas formais e (2) auditoria da configuração de software.

A revisão técnica formal enfoca a correção técnica do objeto de configuração que foi modificado. Os revisores avaliam os itens de configuração para determinar sua consistência com outros itens de configuração, omissões ou efeitos colaterais em potencial. Uma revisão técnica formal deve ser conduzida para todas as modificações, exceto as mais triviais.

Uma auditoria de configuração de software complementa a revisão técnica formal, tratando das seguintes questões:

1. A modificação especificada na Baseline foi executada? Foi adicionada alguma modificação adicional?

2. Uma revisão técnica formal foi conduzida para avaliar a correção técnica?

3. O Processo de software foi seguido e as normas de engenharia de software foram adequadamente aplicadas?

4. A modificação foi “destacada” no Item de Configuração? A data de modificação e o autor foram especificados? Os atributos do objeto de configuração refletem a modificação?

5. Os procedimentos GCS para anotar a modificação, registrá-la e relatá-la foram seguidos?

6. Todos os itens de configuração relacionados foram adequadamente atualizados?

Em alguns casos, as questões de auditoria são formuladas como parte de uma revisão técnica formal. No entanto, quando a GCS é uma atividade formal, a auditoria GCS é conduzida separadamente pelo grupo de garantia de qualidade. Tais auditorias formais também garantem que os itens de configuração corretos (por versão) foram incorporados em uma construção específica em que toda a documentação está atualizada e consistente com a versão que foi construída.

2.4 VSS – Visual Source Safe

Visual Source Safe (VSS) é um produto de sistema de controle de versão que oferece pontos de restauração e colaboração de uma forma paralela, permitindo que empresas de desenvolvimento de software possam trabalhar em diferentes versões do software em simultâneo. O Sistema de controle de versão introduz o conceito de check-in e check-out. O VSS trabalha

com o conceito de liberação de arquivo pessimista, ou seja, quando um desenvolvedor realiza o check-out de um arquivo, outro desenvolvedor não pode efetuar o check-in ou check-out do mesmo artefato. (MSDN, 2004).

O Programa Visual Source Safe (VSS) serve para manter todas as versões de código fonte e qualquer outro tipo de documento, qualquer informação pode ser colocada dentro do programa. O VSS utiliza um sistema de arquivo para armazenar o seu conteúdo, utilizando pastas para armazenar artefatos e versões desses artefatos. (MSDN, 2004).

O VSS disponibiliza um conjunto de interfaces ou API para comunicação com outros processos.

Maiores informações podem ser obtidas na página oficial do produto em (VSS, 2010).

3 A Empresa H2L Solutions

A Empresa H2L Solutions está sendo utilizada como laboratório para a ferramenta FAC.

O Software principal que a empresa comercializa é o chamado “*Business.NET*”, software de gerenciamento e automação comercial. O Software atualmente é comercializado na região da Grande Florianópolis e perspectiva de crescimento para os demais locais do estado de Santa Catarina e, conseqüentemente, um provável aumento da equipe de desenvolvedores acontecerá nos próximos anos.

Não se utiliza metodologia formal do processo de gerência de configuração, as atividades de controle de mudanças, criação de baselines e auditoria de configuração são utilizadas de forma precária e sem documentação. O cenário atual não é escalável do ponto de vista de desenvolvimento, pois o incremento de desenvolvedores acarretaria no caos, pois não há processos formais documentados e executados.

A empresa hoje possui dois desenvolvedores, que trabalham em horários alternados. O repositório utilizado pela empresa para armazenar código fontes e documentos de projetos é o Visual Source Safe (VSS) da Microsoft. A proposta é implantar a ferramenta FAC na empresa sem que haja significativo impacto no seu processo de desenvolvimento.

4 A ferramenta FAC

A Ferramenta foi desenvolvida em paralelo com a sua documentação. Em um primeiro momento, foram definidos os requisitos (funcionais e não funcionais), elaborados os casos de uso e seus refinamentos e, por fim, a arquitetura da ferramenta, elucidada no diagrama de implantação.

A Figura 5 ilustra a tela inicial da ferramenta. A tela inicial apresenta alertas ao usuário logado, por exemplo: Solicitações pendentes a serem avaliadas pelo usuário e também um histórico de todas as baselines (criadas recentemente, desenvolvimento e arquivadas).

The screenshot displays the FAC interface with the following data:

Situação	Data Prevista/Finalizada
Em Desenvolvimento (5)	
Baseline. v.1.8.0.0	19/10/2008
Baseline. v.1.7.0.3	18/10/2008
Baseline. v.1.7.0.2	17/10/2008
Baseline. v.1.7.0.1	16/10/2008
Baseline. v.1.7.0.0	15/10/2008
Criadas Recentemente (5)	
Baseline. v.1.6.0.0	14/10/2008
Baseline. v.1.5.0.0	13/10/2008
Baseline. v.1.4.0.0	12/10/2008
Baseline. v.1.3.0.1	11/10/2008
Baseline. v.1.3.0.0	10/10/2008
Arquivadas (8)	
Baseline. v.1.1.0.2	09/10/2008
Baseline. v.1.1.0.1	08/10/2008
Baseline. v.1.1.0.0	07/10/2008
Baseline. v.1.0.3.0	06/10/2008
Baseline. v.1.0.2.0	05/10/2008
Baseline. v.1.0.0.2	04/10/2008
Baseline. v.1.0.0.1	03/10/2008
Baseline. v.1.0.0.0	02/10/2008

Solicitação	1	2	3	4	5	6
Solicitação 001/10/2008						
Solicitação 002/10/2008	28	1	2	3	4	5
Solicitação 003/10/2008	7	8	9	10	11	12
Solicitação 004/10/2008	14	15	16	17	18	19
Solicitação 005/10/2008	21	22	23	24	25	26
Solicitação 006/10/2008	28	29	30	31	1	2
Solicitação 007/10/2008	4	5	6	7	8	9

Figura 5 - Tela inicial da ferramenta FAC

4.1 Requisitos

A implantação da Ferramenta de Automatização de Gerência de Configuração (FAC) envolve a adoção de procedimentos eficientes que assegurem o sucesso do programa. O levantamento de requisitos faz parte

desse programa e foram divididos em dois módulos: funcionais e não funcionais, como segue abaixo:

4.1.1 Requisitos funcionais

RF01: Deve permitir cadastrar itens de configuração.

RF02: Deve permitir cadastrar baselines, sendo que para cada baseline deve-se identificar o seu nome e adicionar todos os itens de configuração pertencentes a ela.

RF03: Gerar baselines a qualquer momento e produto final desse processo de geração, deve ser um relatório no formato PDF contendo o nome da baseline, todos os itens de configuração pertencentes à baseline, versão em que o item se encontra, e a sua localização “caminho físico” na ferramenta Visual Source Safe.

RF04: Deve permitir a recuperação de uma baseline gerada, a partir do nome da baseline: a ferramenta deve recuperar utilizando a API do Visual Source Safe todos os itens de configuração na versão correspondente.

RF05: Deve permitir cadastrar uma solicitação de mudança.

RF06: Deve permitir aprovar ou reprovar uma solicitação de mudança, e deve-se registrar o motivo e impacto da decisão.

RF07: Deve permitir liberar (*check out*) os itens de configuração para o desenvolvedor que fará a modificação no item utilizando a API do Visual Source Safe.

RF08: Deve permitir que o desenvolvedor através da ferramenta possa registrar o (*check in*) efetuado do Visual Source Safe, e as alterações efetuadas nos itens de configuração.

4.1.2 Requisitos não funcionais

Para garantir a robustez da aplicação alguns requisitos não funcionais são necessários:

RNF 01: Escalabilidade, a arquitetura escolhida deve prover flexibilidade para adicionar novos nós de forma transparente para o usuário.

RNF 02: Segurança, o sistema de autenticação e autorização deve ser implementado para garantir acesso a aplicação de usuários autorizados e também como a aplicação estará sendo utilizada em ambiente Web, é necessário a utilização do protocolo HTTPs para autenticação.

RNF 03: A aplicação deve rodar em ambiente 100% Web, utilizando conceito de RIA (Rich Internet Application).

RNF 04: A aplicação deve-se integrar com a API da aplicação Visual Source Safe.

4.2 *Workflow da FAQ*

O melhor entendimento do processo de trabalho da FAC pode ser visualizado na Figura 6.

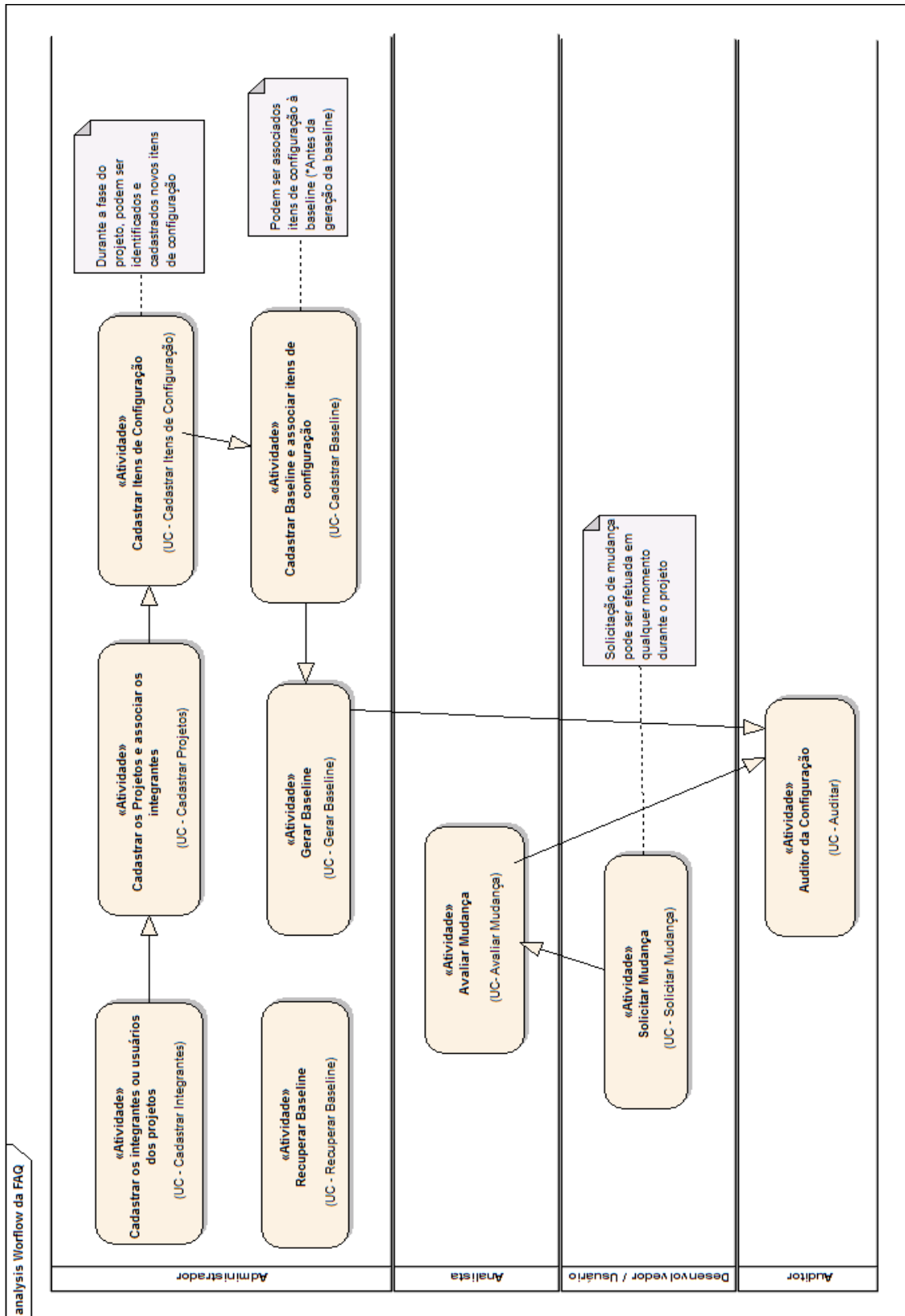


Figura 6 - Workflow da FAC

4.3 Casos de Uso

A Figura 7 ilustra os casos de usos que o sistema deve contemplar para que o objetivo final seja alcançado. O detalhamento de cada caso de uso segue nos próximos capítulos.

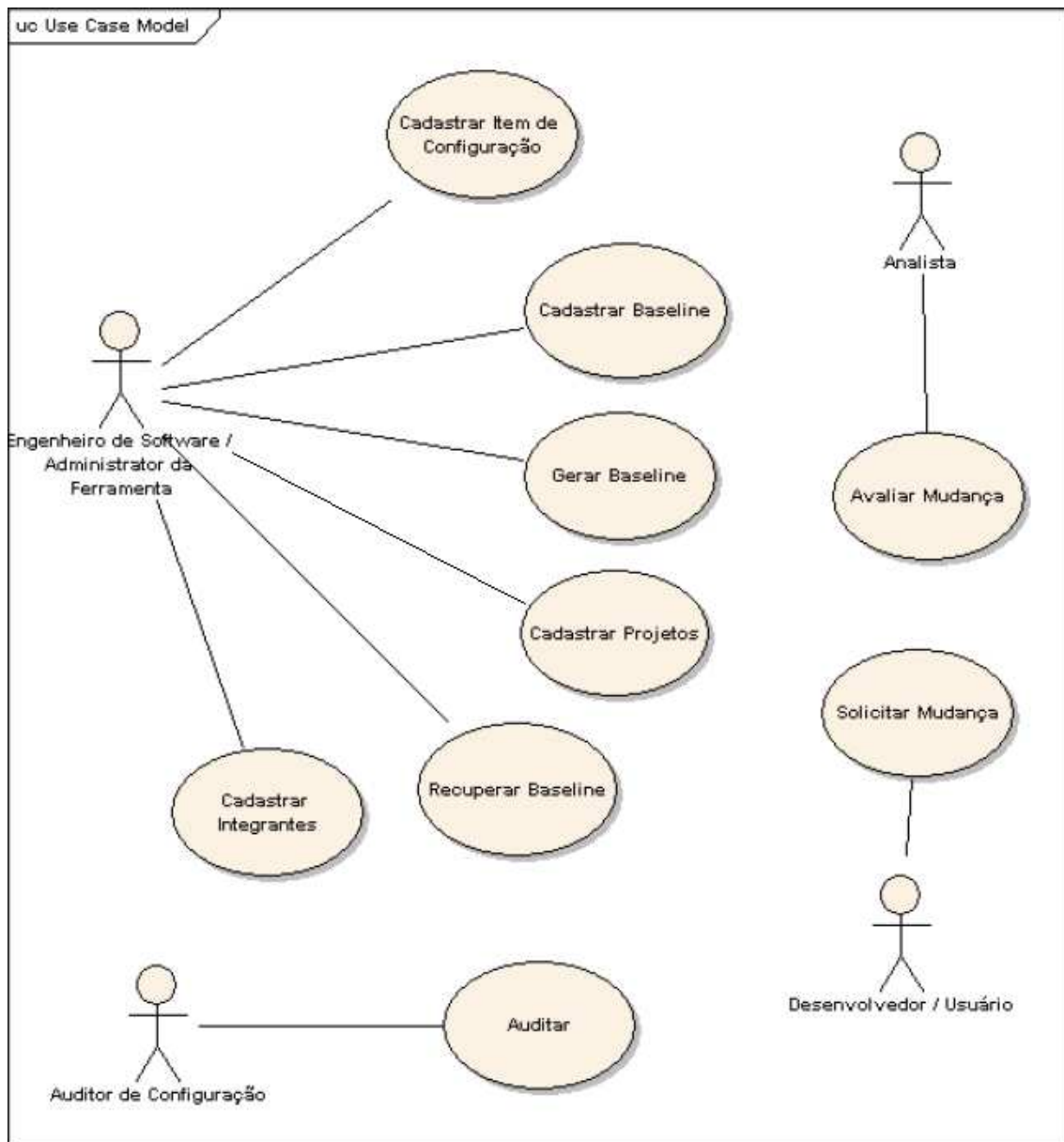


Figura 7 - Diagrama geral de casos de uso

4.3.1 Caso de uso – Cadastrar Item de Configuração

Esse caso de uso define a estrutura básica do sistema, e servirá de insumo para outros casos de uso, uma vez que o item de configuração é o grão da baseline.

O usuário ao cadastrar o item de configuração tem que identificar o seu nome corretamente assim como consta no repositório de dados (VSS) e depois indicar onde se encontra o artefato fisicamente no repositório. A Figura 8 ilustra a tela do cadastro de item de configuração.



Cadastro de Itens de Configuração

Dados Gerais

Nome: Struts.config

Caminho VSS: \$\projetos\faq\web\

Salvar Artefato...

Figura 8 - Tela de cadastro de item de configuração

4.3.2 Caso de uso – Cadastrar Baseline

Nesse caso de uso define-se uma baseline, quando cadastrada a baseline encontra-se no estado de planejada.

Define-se o nome da baseline e posteriormente associa-se ela a um projeto previamente cadastrado (ver: Caso de uso – Cadastro de Projeto). Adicionam-se os itens de configuração que pertencem a essa baseline. A Figura 9 ilustra a tela de cadastro de baseline.

Figura 9 - Tela de cadastro de baseline

4.3.3 Caso de uso – Gerar Baseline

A função desse caso de uso é gerar um relatório em formato PDF com todos os itens de configuração pertencentes à baseline escolhida.

Primeiramente escolhe-se qual projeto deseja-se criar uma baseline, depois de escolhido, apareceram somente as baselines relacionada ao projeto escolhido. Uma vez gerada a baseline ela não poderá ser mais gerada. O relatório PDF é mostrado ao usuário e também armazenado no banco de dados da aplicação e o estado da baseline é mudado para criado. A Figura 10 ilustra a tela de criação de baseline.

Figura 10 - Tela de criação de baseline

O relatório possui informações como nome do projeto e da baseline e também uma lista de itens de configuração, sendo que os itens possuem o nome, o caminho no repositório de dados e a sua versão correspondente no repositório de dados (VSS). A Figura 11 ilustra o relatório gerado pela ferramenta.

FAC – Ferramenta de Automatização de Baseline

Projeto: Ferramenta de Automação de Configuração **Baseline:** B. v. 2.0.0.0

Itens de Configuração.

Nome	Caminho VSS	Versão
Struts.config	\$/fac/struts/struts.config	3
Faces.config	\$/fac/struts/faces.config	4
Cronograma.mpp	\$/fac/struts/cronograma.mpp	2
Uml.eap	\$/fac/struts/uml.eap	1

Figura 11 - Relatório da geração de baseline

O Refinamento do caso de uso – Gerar Baseline, pode ser melhor visualizado na Figura 12 que ilustra o diagrama de seqüência.

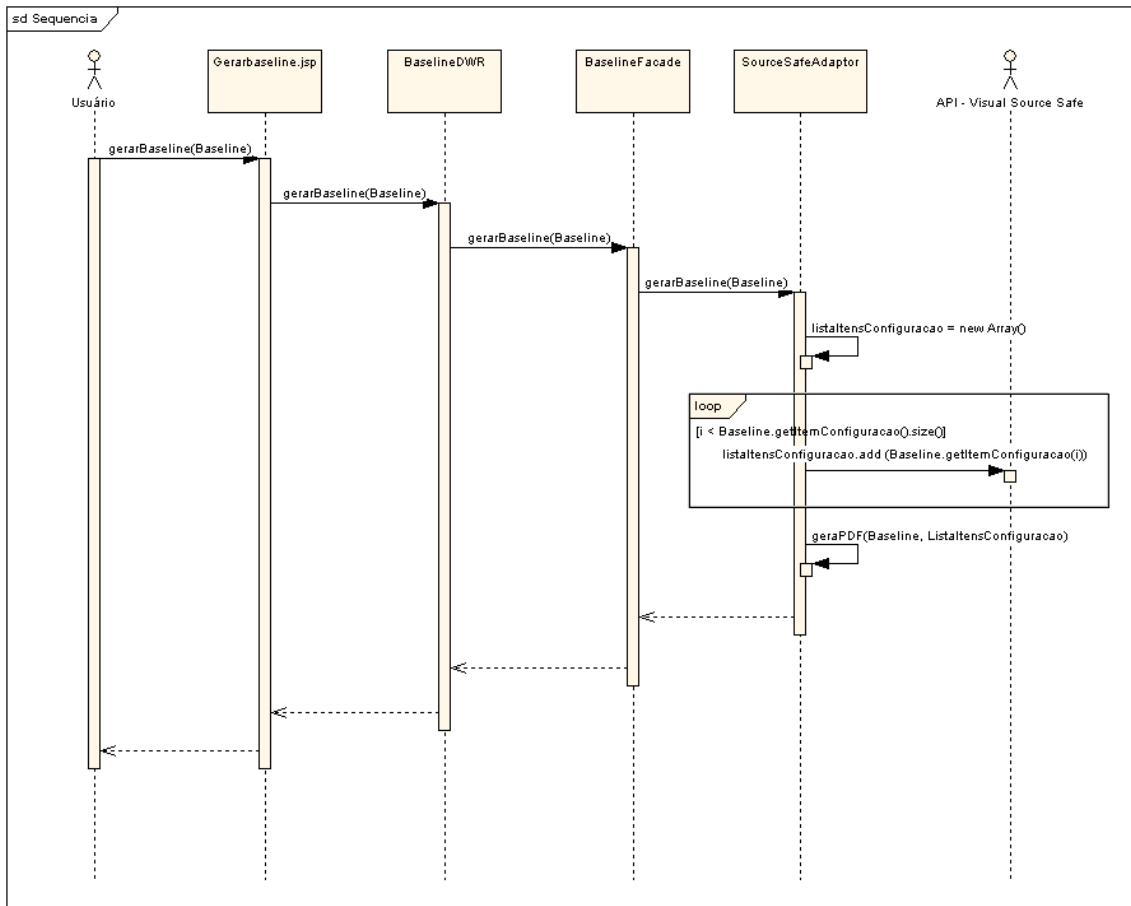


Figura 12 - Diagrama de seqüência da geração de baseline

4.3.4 Caso de uso – Cadastrar Projetos

Esse caso de uso é responsável pelo cadastro de projetos que a corporação tem interesse que fique sobre o processo de gerência de configuração.

Tem-se que identificar o nome, responsável ou gerente de projetos e o caminho físico no VSS.

Podem-se associar integrantes a esse projeto, por exemplo: (Desenvolvedores, Testadores, Implantadores e etc.). A Figura 13 ilustra a tela de Cadastro de projeto.

Cadastro de Projetos

Dados Gerais

Nome:

Responsável:

Caminho VSS:

Integrantes

Integrante:

Integrante(s)	
	Integrante
1	Rodolpo Bur
2	Felipe Henrique
3	João Alberto

Figura 13 - Tela de cadastro de projetos

4.3.5 Caso de uso – Recuperar Baseline

Esse caso de uso tem a função de recuperar uma baseline que se encontra no estado de criada. De acordo com o projeto escolhido, aparecem somente baselines pertencentes aquele projeto. Escolhe-se um caminho local onde será armazenado o conteúdo da baseline. A Figura 14 ilustra a tela de recuperação de baseline.

Recuperar Baseline

Dados Gerais

Projeto:

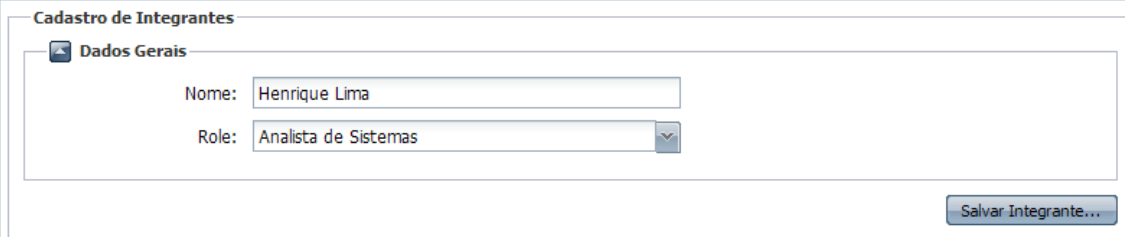
Baseline:

Caminho:

Figura 14 - Tela de recuperação de baseline

4.3.6 Caso de uso – Cadastrar Integrantes

Esse caso de uso é responsável pela criação de usuários ou integrantes da equipe do projeto, a ele é associado uma role ou papel no projeto, por exemplo (Analista de Sistemas, Desenvolvedor, Gerente de Projeto e etc.). A Figura 15 ilustra a tela de cadastro de integrantes.



Cadastro de Integrantes

Dados Gerais

Nome: Henrique Lima

Role: Analista de Sistemas

Salvar Integrante...

Figura 15 - Tela de cadastro de integrantes

4.3.7 Caso de uso – Solicitar Mudança

Nesse caso de uso o usuário ou desenvolvedor pode solicitar uma mudança em projeto específico, descrevendo o motivo de sua solicitação e identificando os itens de configuração que sofreram alteração em função dessa mudança. A Figura 16 ilustra a tela de Solicitação de mudanças.

Solicitação de Mudança

Dados Gerais

Projeto: Ferramenta de Automação de Configuração

Motivo: Tahoma

Mudar a forma como é visualizada o painel de alertas na tela principal.

Itens de Configuração

Artefato:

Versão:

Adicionar Artefato...

Artefato(s)	
Artefato	Versão
1 Faces.config	2
2 Struts.config	1

Página 1 de 1

Salvar Mudança

Figura 16 - Tela de solicitação de mudança

4.3.8 Caso de uso – Avaliar Mudança

Nesse caso de uso o responsável pela avaliação de mudanças tem a opção em visualizar na forma de lista todas as avaliações pendentes. Ele tem duas opções: aceitar ou recusar a mudança. Nessa tela ele tem acesso a informações, como por exemplo: nome do solicitante, data da solicitação, itens de configuração afetados pela mudança e motivo da mudança. A Figura 17 ilustra a tela de lista de solicitações de mudanças.

Solicitações

Lista de Solicitações

	Solicitante	Data	Item de Configuração	Motivo		
1	Henrique Leite	01/10/2008	Struts.config	Alteracao por erro encontrado em teste	☺	☒
2	Felipe Alves	02/10/2008	faces.config	Alteracao por erro encontrado em teste	☺	☒
3	Raimundo Nonato	03/10/2008	cronograma.mpp	Alteracao por erro encontrado em teste	☺	☒
4	Elias João	04/10/2008	menu.jsp	Alteracao por erro encontrado em teste	☺	☒
5	Henrique Leite	05/10/2008	cadCliente.jsp	Alteracao por erro encontrado em teste	☺	☒
6	Raimundo Nonato	06/10/2008	cadFornecedor.jsp	Alteracao por erro encontrado em teste	☺	☒
7	Felipe Alves	07/10/2008	pesquisa.jsp	Alteracao por erro encontrado em teste	☺	☒
8	Henrique Leite	08/10/2008	pesquisaBase.jsp	Alteracao por erro encontrado em teste	☺	☒
9	Elias João	09/10/2008	cliente.java	Alteracao por erro encontrado em teste	☺	☒
10	Henrique Leite	10/10/2008	fornecedor.java	Alteracao por erro encontrado em teste	☺	☒
11	Raimundo Nonato	11/10/2008	planoDeTeste.doc	Alteracao por erro encontrado em teste	☺	☒
12	Henrique Leite	12/10/2008	uml.eap	Alteracao por erro encontrado em teste	☺	☒

Página 1 de 1

Figura 17 - Tela de lista de solicitação de mudança

Caso o avaliador aceite a mudança, é necessário o preenchimento do impacto que a mudança terá no projeto (técnica ou cronograma do projeto). Após a aceitação, a ferramenta FAC libera o check out dos itens de configuração ao desenvolvedor incumbido da tarefa de alteração do projeto. A Figura 18 ilustra a tela de aceitação de solicitação de mudança.

Solicitações

Lista de Solicitações

Ressalva

Impacto: Tahoma

É previsto um impacto de relevância média no sistema.

Salvar Cancelar

Página 1 de 1

Figura 18 - Tela de aceitação de solicitação de mudança

A Figura 19 apresenta o diagrama de máquina de estados do processo de solicitação e avaliação de uma mudança.

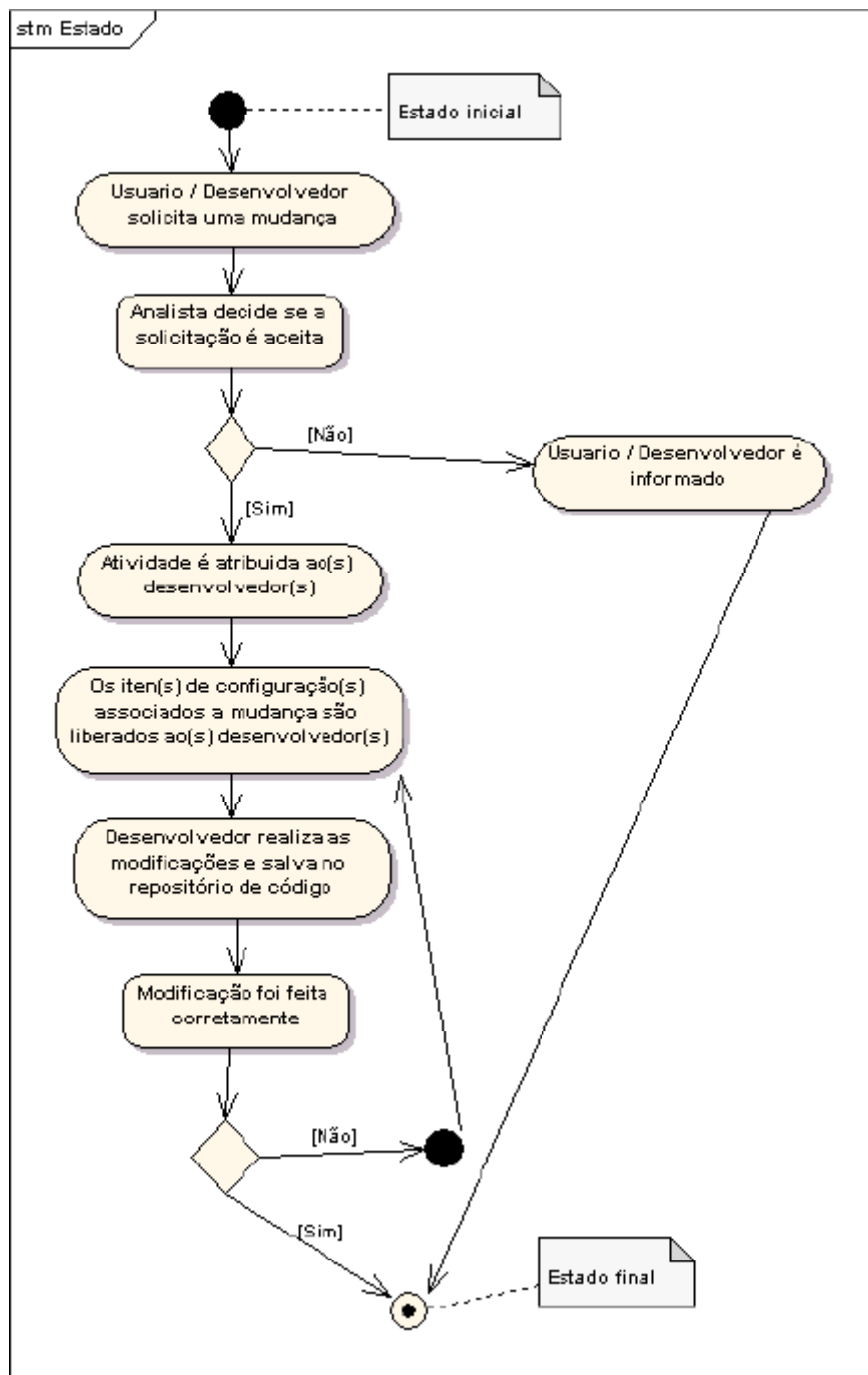


Figura 19 - Diagrama de estados da solicitação e aprovação de mudança

4.3.9 Caso de uso – Auditar

Nesse caso de uso o auditor de configuração tem a opção de escolher um projeto e depois escolher uma baseline do projeto associado, automaticamente todos os itens de configuração e as versões que compõem a baseline são mostrados na forma de lista. O Auditor tem a função de verificar se os itens de configuração que compõem a baseline estão efetivamente no repositório de código, e assim dar o seu parecer sobre aquela baseline. A Figura 20 ilustra a tela de auditoria de configuração.

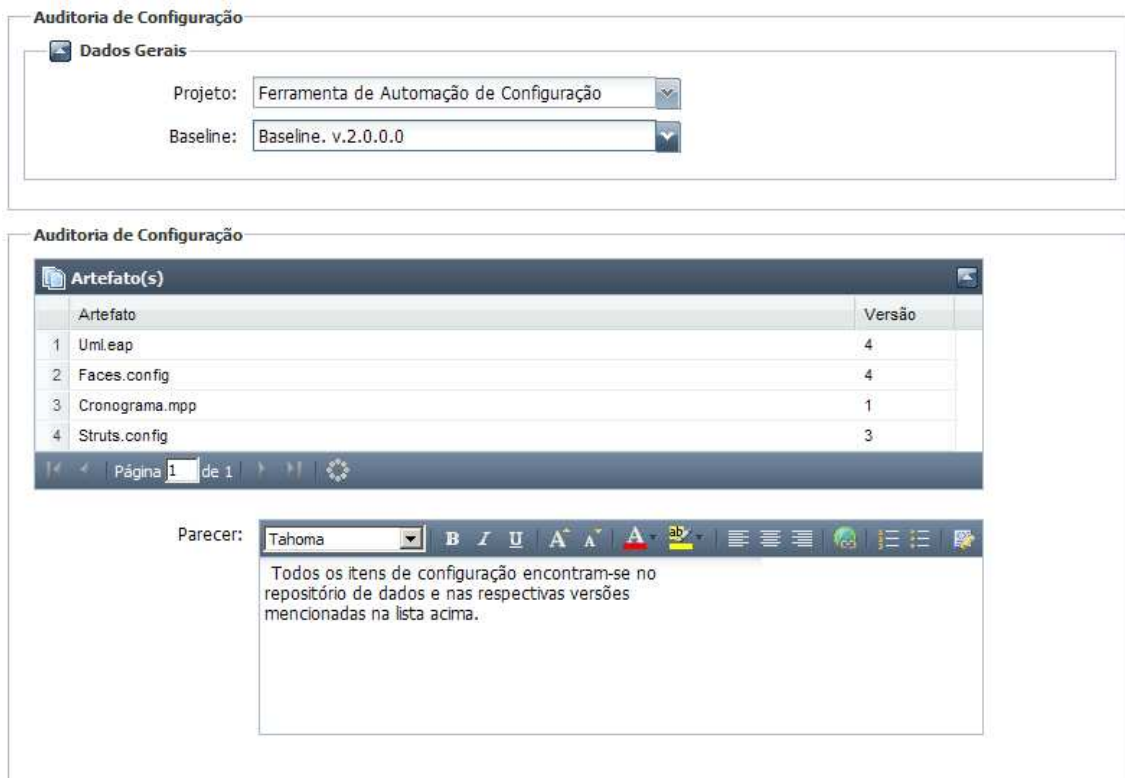


Figura 20 - Tela de auditoria de configuração

4.4 Diagrama de Implantação (Deployment)

Foi proposta uma arquitetura Cliente/Servidor de 3 camadas, utilizando um servidor de aplicação (JBOSS v.5.0) e um servidor Web (Apache Web Server v.2.2) para redirecionamento e balanceador de carga. Para conversação

entre o cliente (Browser) e o servidor recomenda-se utilizar um protocolo seguro, por exemplo HTTPS.

A comunicação com a API do Visual Source Safe (VSS) será feita utilizando o conceito de objetos distribuídos (COM), as interfaces de comunicação do VSS são distribuídas através de sua instalação em ambiente do sistema operacional Windows 98 ou superior.

A Figura 21 apresenta o diagrama de implantação e exemplifica os componentes envolvidos e suas iterações.

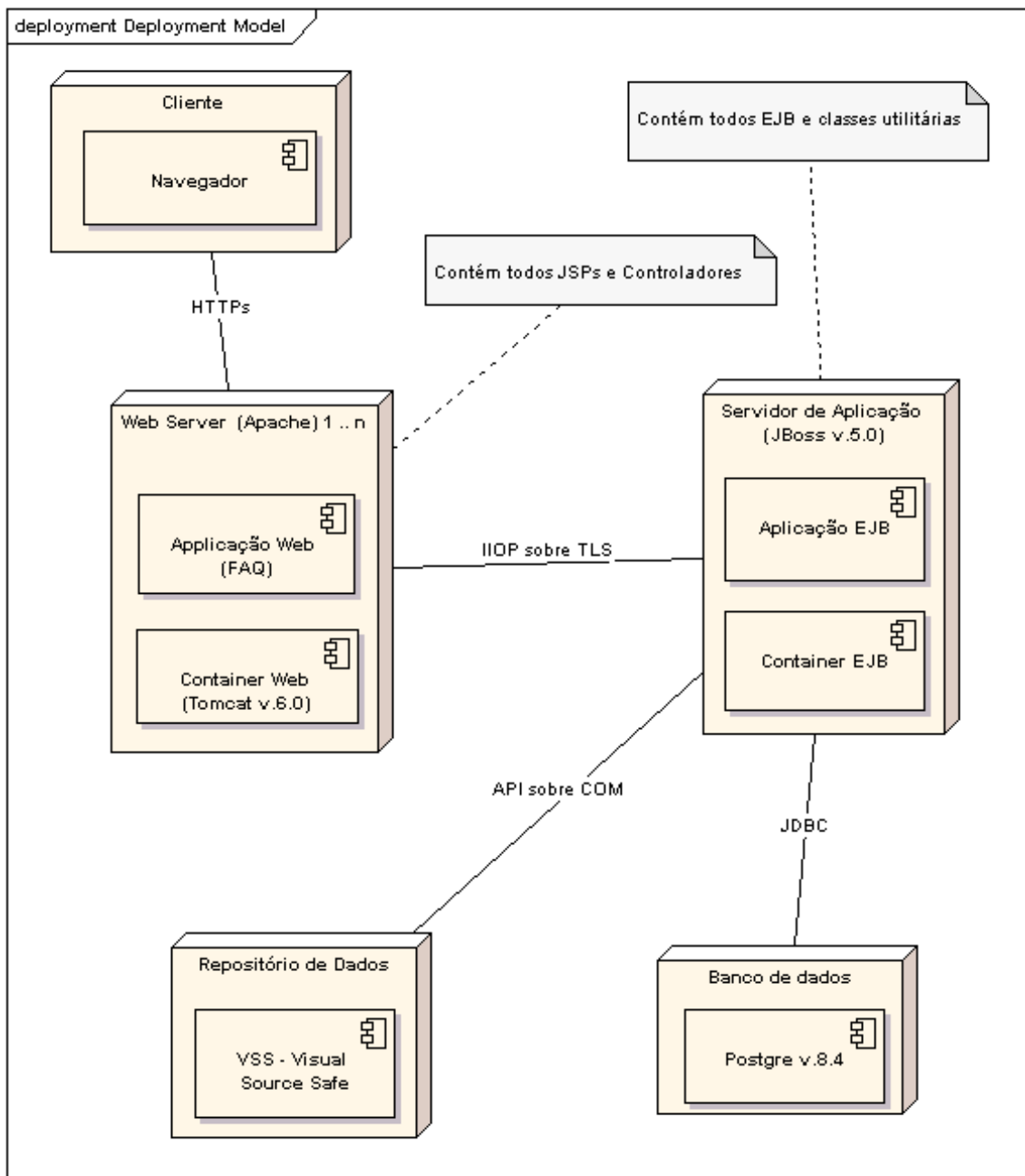


Figura 21 - Diagrama de implantação

4.5 Comunicação com a API do Visual Source Safe

A comunicação com o VSS é feita utilizando uma API nativa chamada JIntegra for COM (JINTEGRA, 2010), ela é responsável pelo encapsulamento das chamadas ao VSS, disponibilizando de forma transparente ao

desenvolvedor o acesso ao VSS, a Figura 22 ilustra a classe de conexão utilizada no projeto FAC na linguagem Java.

```
import SSAPI.*;

public class VssClient {

    /*
     * Método para conectar a base de dados do Visual Source Safe
     */
    public VSSDatabase Conecta() throws Exception {
        VSSDatabase database = null;
        try {
            database = new VSSDatabase();
            database.open("c://vss/vss.ini", "admin", "senha");
        } catch (Exception e) {
            e.printStackTrace();
        }
        return database;
    }

    /*
     * Recupera todos itens de configuração de um projeto
     */
    public void recuperaItens(VSSDatabase database, String spec, String fonte){

        IVSSItem item = database.getVSSItem(spec, false);
        if (item.isDirectory()){
            //chama função de recursão
            recuperaItens(database, item.getDirectory(), fonte);
        } else {
            item.checkout("Comentário de ckeck-out", fonte, 0);
        }
    }
}
}
```

Figura 22 - Classe de Conexão com o VSS

5 Conclusão

O desenvolvimento da ferramenta FAC envolveu, atividades comuns de desenvolvimento de software e também conceitos da Engenharia de Software tais como Gerência de Configuração que foram fundamentais para a construção da ferramenta. O modelo de maturidade MPS.BR e o processo de GCS constituíram os alicerces deste trabalho, do ponto de vista conceitual.

Do ponto de vista prático, a integração com a API do Visual Source Safe (VSS) foi de grande importância, pois ela subsidiou a possibilidade de liberar ou restringir itens de configuração aos usuários e também a leitura das versões dos itens de configuração, função essa que é melhor visualizada no relatório de geração de baseline.

5.1 Resultados obtidos

O principal resultado obtido foi o desenvolvimento da ferramenta FAC. A FAC permite que gere relatórios de baselines, solicite mudanças, avaliação de mudanças e relatórios de auditoria de configuração.

A ferramenta FAC no momento da elaboração desse documento está em fase de implantação na empresa H2L Solutions, atendendo as necessidades e anseios da empresa.

Foram atendidos os sete resultados esperados do processo de Gerência de Configuração do processo MPS.BR.

Como resultado secundário, pode ser citado a participação, por parte do autor deste trabalho, dos seguintes cursos: *Introdução ao MPS.BR* e *Curso para Implementador Oficial MPS.br*, sendo que no ultimo também pode citar que o autor obteve a certificação: *Implementador Oficial MPS.BR*. O aperfeiçoamento dos conhecimentos sobre os conceitos de Engenharia de Software envolvidos nesse trabalho também foi significativo.

5.2 Limitações

A ferramenta FAC está fortemente acoplada a API do Visual Source Safe (VSS), dificultando com que a ferramenta FAC trabalhe com outro repositório de código, por exemplo (Subversion ou CVS). Esta limitação foi reportada por um desenvolvedor da empresa H2L Solutions, pois a empresa planeja em breve a migração de seu repositório.

5.3 *Trabalhos futuros*

Em função da limitação reportada no capítulo anterior, um possível trabalho futuro é a implementação de uma camada intermediária entre a comunicação da ferramenta FAC e API do Visual Source Safe. Assim fica a opção da escolha de outro repositório de código, por exemplo (Subversion ou CVS).

5.4 *Considerações finais*

A implantação do processo de Gerência de Configuração aderente ao modelo MPS.BR em uma organização envolve o estabelecimento de uma série de detalhes e um alto grau de formalismo do processo. A ferramenta FAC, que implementa as recomendações do processo de Gerência de Configuração, é uma resposta a esses detalhes e ao formalismo.

A ferramenta FAC oferece o suporte necessário à implantação de um processo de Gerência de Configuração aderente ao modelo MPS.BR eficiente em um ambiente organizacional.

6 Referencias Bibliográficas

ABNT, Associação Brasileira de Normas Técnicas. **NBR ISO/IEC 12207 – Tecnologia de informação – Processos de ciclo de vida de software**. Rio de Janeiro: ABNT, 2001.

BARBARESCO, Eduardo Alexandre. **Software de apoio ao processo de gerência da configuração segundo normas e modelos da qualidade**. Blumenau: 2000.

DAR, Dart, S. “**Concepts in Configuration Management Systems, Software Engineering Institute**”, disponível em <http://www.sei.cmu.edu/legacy/scm>

FREDERICO, César, S. “**Ferramenta de suporte à área de processo Medidas e Análises do CMMI**”, Florianópolis, 2007

HAUCK, Jean Carlo Rossa. **Uma abordagem de modelagem de processos suportada por um guia de referência alinhado ao CMMI-DEV, MPS.BR e ISSO/IEC 15504**. Florianópolis, 2007

ISO, International Organization for Standardization. **ISO/IEC 15504: Information technology – Software process assessment**, ISO/IEC International Standard, 2005.

MOLINARI, Leonardo. **Gerência de Configuração Técnicas e Práticas no Desenvolvimento do Software**. Florianópolis: Visual Books, 2007.

PRESSMAN, Roger S. **Software engineering: A Practitioner’s Approach, 6th. Ed.**, McGraw-Hill, 2006.

SEI, Software Engineering Institute. **CMMI for Development: Version 1.2: CMMI-DEV**. USA: SEI, 2006.

SOFTEX, **MPS.BR - Melhoria de Processo do Software Brasileiro, Guia de Implementação**. Brasília: Softex, 2009.

MSDN, **Microsoft Visual SourceSafe Roadmap**. Disponível em:
<http://msdn.microsoft.com/en-us/library/aa302175.aspx>, Acesso em 10/03/2010

VSS, **Visual SourceSafe**, Disponível em: <http://msdn.microsoft.com/en-us/vstudio/aa700907.aspx>, 2010.

JINTEGRA, **JIntegra for COM**, Disponível em <http://j-integra.intrinsyc.com/>,
2010