

**UNIVERSIDADE FEDERAL DO PARÁ  
CENTRO TECNOLÓGICO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**Carla Marina Costa Paxiúba**

**Acompanhamento e Avaliação de Projetos através da  
Monitoração de Eventos em um Ambiente de Gestão de  
Processos de Software**

**UFPA / CT / PPGEE  
Campus Universitário do Guamá  
Belém-Pará-Brasil  
2007**



**UNIVERSIDADE FEDERAL DO PARÁ  
CENTRO TECNOLÓGICO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**Carla Marina Costa Paxiúba**

**Acompanhamento e Avaliação de Projetos através da  
Monitoração de Eventos em um Ambiente de Gestão de  
Processos de Software**

**UFPA / CT / PPGEE  
Campus Universitário do Guamá  
Belém-Pará-Brasil  
2007**

**UNIVERSIDADE FEDERAL DO PARÁ  
CENTRO TECNOLÓGICO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**Carla Marina Costa Paxiúba**

**Acompanhamento e Avaliação de Projetos através da  
Monitoração de Eventos em um Ambiente de Gestão de  
Processos de Software**

**Dissertação submetida à Banca  
Examinadora do Programa de  
Pós-Graduação em Engenharia  
Elétrica da UFPA para a obtenção  
do Grau de Mestre em  
Engenharia Elétrica**

**UFPA / CT / PPGEE  
Campus Universitário do Guamá  
Belém-Pará-Brasil  
2007**

**P341a**

**Paxiúba, Carla Marina Costa**

**Acompanhamento e avaliação de projetos através da monitoração de eventos em um ambiente de gestão de processos de software / Carla Marina Costa Paxiúba; orientador, Rodrigo Quites Reis.-2007**

**Dissertação (Mestrado) – Universidade Federal do Pará, Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica,**

**1. Software – projeto e construção. 2. Projeto de sistemas – gerência. I. Título.**

**CDD – 20. ed. 005.1**

**UNIVERSIDADE FEDERAL DO PARÁ  
CENTRO TECNOLÓGICO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**ACOMPANHAMENTO E AVALIAÇÃO DE PROJETOS ATRAVÉS DA  
MONITORAÇÃO DE EVENTOS EM UM AMBIENTE DE GESTÃO DE  
PROCESSOS DE SOFTWARE**

**Autora: Carla Marina Costa Paxiúba**

DISSERTAÇÃO DE MESTRADO SUBMETIDA À AVALIAÇÃO DA BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA UNIVERSIDADE FEDERAL DO PARÁ E JULGADA ADEQUADA PARA OBTENÇÃO DO GRAU DE MESTRE (OU DOUTOR) EM ENGENHARIA ELÉTRICA NA ÁREA DE COMPUTAÇÃO APLICADA.

APROVADA EM: 04/07/2007

BANCA EXAMINADORA:

---

Prof. Dr. Rodrigo Quites Reis  
(ORIENTADOR – UFPA)

---

Prof. Dr. Roberto Célio Limão de Oliveira  
(MEMBRO – UFPA)

---

Prof. Dr. Eloi Luiz Favero  
(MEMBRO – UFPA)

---

Prof. Dr. Manoel Ribeiro Filho  
(MEMBRO – UFPA)

VISTO:

---

Prof. Dr. Evaldo Gonçalves Pelaes  
(COORDENADOR DO PPGEE/CT/UFPA)  
UFPA / CT / PPGEE

*Aos meus pais*

## Agradecimentos

A Deus.

Aos meus pais pelo incentivo e dedicação, mesmo à distância.

Ao meu orientador Rodrigo Quites Reis, pelo conhecimento compartilhado, pelo incentivo, ajuda e paciência em todos os momentos e, principalmente, pelo grande e essencial apoio na realização deste trabalho. Muito Obrigada.

A professora Carla Reis pelas muitas contribuições a minha formação.

Aos professores do mestrado pelos ensinamentos compartilhados.

Ao Marcelo Pereira, pela ajuda primordial na implementação deste trabalho.

A Luciana, minha grande amiga com quem eu pude contar sempre, por estar sempre presente, apoiando, incentivando e ajudando em todos os momentos deste mestrado e da vida.

A Valéria e Andréa pelos muitos e bons anos de amizade.

As amigas Jane, Keilla e Zilda pela convivência, repleta de companheirismo, solidariedade e apoio nos momentos de saudade de casa.

Ao Ticiano, pelos ensinamentos profissionais, pela disponibilidade em me ajudar, e, sobretudo, pelas essenciais palavras de incentivo e motivação. Muito Obrigada.

Aos colegas do Labes.

Aos colegas do Serpro.

Ao Labes.

Ao Serpro.

Ao CNPq.

A Universidade Federal do Pará.

# Sumário

---

<b>CAPÍTULO 1 - INTRODUÇÃO .....</b>	<b>18</b>
1.1 Motivação.....	19
1.2 Objetivo .....	20
1.3 Metodologia de Trabalho .....	21
1.4 Organização.....	21
<b>CAPÍTULO 2 - QUALIDADE E GERENCIAMENTO DE PROJETOS DE SOFTWARE.....</b>	<b>23</b>
2.1 CMMI – Capability Maturity Model Integration .....	23
2.1.1. Nível de Maturidade 1: Inicial.....	24
2.1.2. Nível de Maturidade 2: Gerenciado .....	25
2.1.3. Nível de Maturidade 3: Definido .....	25
2.1.4. Nível de Maturidade 4: Gerenciado Quantitativamente.....	26
2.1.5. Nível de Maturidade 5: Otimizado.....	27
2.1.6. Comparando os níveis de maturidade do CMMI .....	28
2.2 MPS.Br – Melhoria do Processo de Software Brasileiro .....	31
2.3 PMBOK – Project Management Body of Knowledge .....	32
2.4 Acompanhamento de Processos no CMMI, MPS-Br e PMBOK.....	34
2.4.1 Acompanhamento de Projetos no CMMI .....	34
2.4.2 Acompanhamento de Projetos no MPS.Br .....	35
2.4.3 Acompanhamento de Projetos no PMBOK .....	37
2.5 Considerações Finais.....	37
<b>CAPÍTULO 3 - MODELO PARA ACOMPANHAMENTO DE PROJETOS DE SOFTWARE.....</b>	<b>38</b>
3.1 Introdução.....	38
3.2 Acompanhamento de Projetos .....	39
3.3 Requisitos da Ferramenta .....	42
3.4 Proposta de Atendimento dos Requisitos da Ferramenta.....	46
3.5 Considerações Finais.....	48
<b>CAPÍTULO 4 - A FERRAMENTA MONITORING PROCESS.....</b>	<b>49</b>

<b>4.1 Introdução.....</b>	<b>49</b>
<b>4.2 Ambiente WebAPSEE.....</b>	<b>49</b>
<b>4.3 Gerência de Eventos no Ambiente WebAPSEE.....</b>	<b>51</b>
4.3.1 O modelo de dados do Pacote <i>Log</i> .....	52
4.3.2 Mecanismo de Gerência de Eventos .....	55
4.3.3 Arquitetura Monitoring Process .....	62
<b>4.3 A Ferramenta Monitoring Process .....</b>	<b>66</b>
4.3.1 Relatórios Monitoring Process .....	66
4.3 Considerações Finais.....	74
<b>CAPÍTULO 5 - AVALIAÇÃO DA PROPOSTA.....</b>	<b>75</b>
<b>5.1 Simulação .....</b>	<b>75</b>
5.1.1 - O processo .....	75
<b>5.2 Análise de Características.....</b>	<b>79</b>
5.2.1 Requisitos para Acompanhamento e Monitoração de Projetos.....	79
5.2.2 Ferramentas Analisadas .....	80
5.2.3 Quadro Comparativo.....	82
5.2.4 Requisitos de Qualidade.....	83
<b>5.3 Considerações Finais.....</b>	<b>84</b>
<b>CAPÍTULO 6 - CONSIDERAÇÕES FINAIS .....</b>	<b>86</b>
<b>6.1 Principais Contribuições.....</b>	<b>86</b>
<b>6.2 Trabalhos Futuros.....</b>	<b>87</b>
<b>6.3 Considerações Finais.....</b>	<b>88</b>
<b>REFERÊNCIAS .....</b>	<b>89</b>
<b>APÊNDICE A - JXLS.....</b>	<b>93</b>
<b>APÊNDICE B – NOTAÇÃO APSEE-PML.....</b>	<b>95</b>
<b>B.1 Atividades.....</b>	<b>95</b>
<b>B.2 Agentes, Grupos, Recursos e Artefatos .....</b>	<b>95</b>
<b>B.3 Conexões.....</b>	<b>97</b>
<b>APÊNDICE C – EXTENSÃO WEBAPSEE .....</b>	<b>99</b>

# Lista de Figuras

---

Figura 1. Os 5 Níveis de Maturidade do CMMI .....	24
Figura 2. Componentes do Modelo CMMI .....	28
Figura 3. Gerências do PMBOK .....	33
Figura 4. Modelo de Acompanhamento de Projetos.....	39
Figura 5. Atividade Acompanhamento de Processo.....	40
Figura 6. Gerar Relatório.....	41
Figura 7. Layout Relatórios de Desvios .....	47
Figura 8. Layout Relatório de Monitoramento .....	47
Figura 9. Layout de Relatório de Acompanhamento de Projetos.....	48
Figura 10. Representação gráfica para as principais construções da Linguagem de Modelagem de Processo do webAPSEE [webAPSEE, 2006]. 50	50
Figura 11. Telas WebAPSEE.....	51
Figura 12. Pacote Log .....	53
Figura 13. Rule 1.1 – Início da execução do Processo .....	56
Figura 14. Rule 1.2 Sincronização de Agenda.....	57
Figura 15. Rule 4.4 Início de Atividade Decomposta .....	57
Figura 16. Alocação de Recursos.....	58
Figura 17. Ativação de uma conexão.....	58
Figura 18. Processo Exemplo Log .....	59
Figura 19. Execução Processo Exemplo Log .....	60
Figura 20. Exemplo com registro de eventos coletados na execução de um processo exemplo .....	60
Figura 21. Início Atividade Processo Exemplo Log .....	61
Figura 22. Exemplo Log de Eventos – Início de Atividade .....	61
Figura 23. Exemplo Log de Eventos – Término do Processo.....	62
Figura 24. Exemplo Log de Eventos – Término de Execução .....	62
Figura 25. Fluxo de Geração de Relatórios.....	63
Figura 26. Diagrama de Atividades de Geração de Relatórios .....	64
Figura 27. Camadas que compõem a Arquitetura do webAPSEE [Lima et al. 2006].....	65
Figura 28. Relatório Effort Monitoring .....	68
Figura 29. Relatório Effort Deviation .....	69
Figura 30. Relatório Schedule Monitoring .....	70
Figura 31. Relatório Schedule Deviation .....	70
Figura 32. Relatório Cost Monitoring .....	71
Figura 33. Relatório Cost Deviation.....	72
Figura 34. Relatório Acompanhar Atividade.....	73
Figura 35. Relatório Activity Monitoring .....	73
Figura 36. Processo Exemplo Sincon .....	76

Figura 37.	Atividade Decomposta Elicitação de Requisitos.....	77
Figura 38.	Relatório Schedule Deviation do Processo Exemplo .....	78
Figura 39.	Relatório Cost Deviation do Processo Exemplo.....	78
Figura 40.	RelatórioEffort Deviation do Processo Exemplo.....	79
Figura 41.	Notação para atividades na APSEE-PML .....	95
Figura 42.	Notação para Agente, Grupo, Recursos e Artefatos na APSEE- PML	96
Figura 43.	Notação para Conexões na APSEE-PML.....	97
Figura 44.	Tipos e Dependência de Conexões .....	97

# Lista de Tabelas

---

Tabela 1. Áreas de Processo do CMMI.....	30
Tabela 2. Áreas de Processo do MPS.Br .....	32
Tabela 3. Requisitos Não Funcionais.....	46
Tabela 4. Seqüência de Atividades Processo Exemplo .....	76
Tabela 5. Características da Ferramenta.....	80
Tabela 6. Comparação de Ferramentas .....	82
Tabela 7. Requisitos de Qualidade [RUP].....	83
Tabela 8. Requisitos de Qualidade Monitoring Process.....	84
Tabela 9. Principais Requisitos Atendidos.....	86

## Lista de Quadros

---

Quadro 1. Requisito "Necessidade Exemplo" .....	42
Quadro 2. Requisito "Monitorar as Atividades do Projeto" .....	42
Quadro 3. Requisito "Monitorar os Custos do Projeto" .....	43
Quadro 4. Requisito "Monitorar os Prazos do Projeto" .....	43
Quadro 5. Requisito "Monitorar os Esforços no Projeto" .....	44
Quadro 6. Requisito "Identificar Possíveis Desvios do Projeto" .....	44
Quadro 7. Requisito "Indicar Atividades com Desvios de Estimativas" .....	45
Quadro 8. Requisito "Permitir o Acompanhamento dos projetos da Organização" .....	45
Quadro 9. Requisito "Permitir Fácil Manipulação dos Relatórios de Acompanhamento" .....	45

# Lista de Abreviaturas e Siglas

CMMI - Capability Maturity Model Integration

MPS.Br – Melhoria de Processos do Software Brasileiro

PMBOK - Project Management Body of Knowledge

PSEE - Process-centered Software Engineering Environment

SERPRO – Serviço Federal de Processamento de Dados

SEI - Software Engineering Institute

UFPA – Universidade Federal do Pará

## Resumo

Uma das maiores dificuldades encontradas no gerenciamento de projetos de software é saber a dimensão do que está sendo gerenciado. Inúmeras dúvidas são pertinentes aos gerentes de projeto quando se fala em dimensionamento, prazo e custo dos projetos. Neste contexto pesquisas mostram que a minoria dos projetos são finalizados no tempo e orçamento estimados e com todas as funcionalidades acordadas implementadas. Os demais projetos, ou são finalizados com prazos e custos ultrapassados ou não chegam a serem concluídos. A gestão de projetos e produtos de software somente atinge o nível desejado de eficácia e exatidão se houver medidas que possibilitem gerenciar através de fatos. E, mais importante do que estabelecer estas medidas é acompanhá-las durante toda a execução dos projetos.

O trabalho apresentado nesta dissertação propõe um acompanhamento de projetos eficiente, guiada pelas normas das principais abordagens de melhoria de processo existente e fazendo uso do acompanhamento das métricas como ferramenta fundamental para uma efetiva gerência de projetos.

Este acompanhamento será obtido através da extensão do mecanismo de registro de eventos do ambiente de gerenciamento de processos WebAPSEE e emissão extração de relatórios gerenciais de acompanhamento de projetos que estão alinhados com os requisitos definidos pelas abordagens de melhoria de processo propostos por modelos como o CMMI e MPS-BR. A proposta foi avaliada através de uma análise crítica envolvendo a simulação da execução de um projeto e realização de seu acompanhamento através da ferramenta proposta.

**Palavras-chave:** Acompanhamento de Projetos, WebAPSEE, Monitoring Process, CMMI e MPS.Br

## Abstract

One of the greatest challenges found in software project management nowadays is to know the true dimension of what is being managed. There are several concerns that are relevant to project managers with respect to the estimation of size, duration, and cost of a project. In this context researchs show that the minor part of the projects are completed on time, on budget, and with all the features and functions originally specified. The other projects are completed overbudget, late or fail. Software project and product management can reach a given level of efficacy and exactness only if certain measurements are made in order to make it possible to manage based on facts.

This dissertation focuses on efficient project management process which must be guided by the international process improvement standards and through the use of metrics as a fundamental tool for an effective project management. The basic assumption is that, without the use of adequate metrics, the planning and monitoring of projects become empirical activities, based solely on the feeling and experience of the project manager.

This work presents a proposal for projects monitoring through the extension of the event recording mechanism of the WebAPSEE process-centered software engineering environment. The mechanism is the base to provide software support to generate monitoring reports based on process improvement approaches like CMMI and the Brazilian MPS.BR. The proposal was evaluated through a critical analysis involving the monitoring of simulated project with the proposed tool.

**Key Words:** Monitoring Projects, WebAPSEE, Monitoring Process, CMMI e MPS.Br

# Capítulo 1 - Introdução

---

A Gerência de Projetos é uma disciplina relativamente nova, mas que vem despertando interesse em profissionais de várias áreas [PMBOK, 2004]. Na Engenharia de Software, a percepção da importância do gerenciamento de projetos de software, e conseqüentemente seu desenvolvimento como uma disciplina desta área, vem crescendo continuamente. O aumento no interesse sobre as técnicas de gerenciamento de projeto é resultado de um crescimento das incertezas dentro do ambiente de negócio. Mudanças políticas, o avanço na tecnologia da informação, a globalização, entre outros aspectos, têm contribuído para a instabilidade no universo dos negócios. E nesse ambiente de mudanças, as organizações procuram utilizar técnicas de gerenciamento para melhor controlar, acompanhar e garantir o sucesso de seus projetos [Meneses, 2001].

Atualmente, com a dependência cada vez maior das organizações em relação à tecnologia da informação, a geração de produtos de software com qualidade e a um custo compensador, torna-se um fator crítico de sucesso [Fernandes, 1995]. A gestão de negócios está cada vez mais focada em garantir a satisfação do cliente através de produtos e serviços com elevados e comprovados padrões de qualidade [Maciel, 2000].

No contexto da indústria de software, o mercado tem exigido produtos ainda mais sofisticados e em prazos de desenvolvimento mais curtos, o que tem impulsionado a pesquisa na área de qualidade de software, objetivando encontrar meios para garantir que o software produzido atenda às expectativas do cliente e aos atributos de qualidade definidos pela empresa fornecedora de software.

Um projeto de sucesso é definido pelo PMBOK [2004] como aquele entregue no prazo, dentro do orçamento previsto e atendendo aos requisitos de funcionalidade e de qualidade acordados com o cliente. Neste contexto, é consenso na literatura que a gerência de projeto é um dos aspectos mais críticos para o sucesso dos projetos de software, pois esta é a responsável por planejar o projeto,

definindo seu escopo, estimando seus custos, recursos alocados e prazo, bem como controlar o projeto e acompanhar sua execução segundo o que foi planejado.

O gerenciamento de projetos de software não é uma tarefa trivial. Controlar grandes equipes de desenvolvimento de modo que o aproveitamento das mesmas seja satisfatório exige um acompanhamento intenso, além da utilização de técnicas e modelos de métricas que quantifiquem e qualifiquem o andamento do projeto [Henderson, 1997]. Dessa forma, as métricas de software têm se tornado uma ferramenta essencial para apoiar o gerente de projetos na captura das informações relevantes para o gerenciamento da qualidade do produto e do processo de desenvolvimento. Isto porque, as métricas fornecem um conjunto de informações tangíveis para planejar, realizar estimativas, gerenciar e controlar os projetos com maior precisão, identificando os desvios em relação ao que foi planejado.

O trabalho apresentado nesta dissertação busca a qualidade do desenvolvimento de software, através de um acompanhamento de projetos eficiente, guiada pelas normas do setor fazendo uso do acompanhamento de projetos como atividade fundamental para uma efetiva gerência de projetos.

Em seguida são apresentadas a motivação, objetivos, metodologia e a organização deste trabalho.

## **1.1 Motivação**

A gerência de projetos abrange todo o processo de desenvolvimento de software. Ela inicia-se com a definição do escopo e o planejamento do projeto a ser desenvolvido, e segue com o acompanhamento do projeto, coleta de dados, avaliação das métricas e revisões dos planos do projeto, visando à entrega do produto dentro do prazo e custo esperados e com a qualidade adequada, sendo de fundamental importância para o sucesso do projeto.

Dentro do contexto de busca pela qualidade do processo de desenvolvimento de software, várias normas e padrões específicos para software têm sido propostos. Entre eles, podemos destacar o modelo de origem norte-americana CMMI (*Capability Maturity Model Integration*) [Chrissis et al, 2003] e MPS.Br (Melhoria do Processo de Software Brasileiro) [Softex, 2005]. Ambos os modelos propõem um caminho gradual, através de níveis de maturidade da capacitação, que leva as organizações a se aprimorarem continuamente na busca da suas próprias soluções

para os problemas inerentes ao desenvolvimento sistemático de software. Particularmente o nível 2 de maturidade proposto pelo CMMI, chamado Gerenciado, e Nível G do MPS.Br, chamado Parcialmente Gerenciado, sugerem que sejam estabelecidos processos básicos de gerência de software para controlar e acompanhar custos, cronograma e funcionalidades.

Mesmo com o grande número de propostas visando a melhoria nos processos de gerenciamento de projetos, uma pesquisa realizada pelo *Standish Group*, no ano de 2000 [Standish Group], mostra que apenas 28% dos projetos são bem sucedidos, ou seja, são finalizados no tempo e orçamento estimados e com todas as funcionalidades acordadas implementadas. Dos demais projetos, 49% são finalizados, porém com prazos e custos ultrapassados, e 23% dos projetos nem chegam a serem concluídos. Outra pesquisa também realizada pelo *Standish Group* [Standish Group] em 1995 revelou que apenas 9% dos projetos de software em grandes empresas são finalizados dentro do prazo e custo esperados. Esse número aumenta para 16% nas empresas de porte médio e para 28% nas empresas consideradas de pequeno porte. Esses dados mostram que à medida que os projetos aumentam sua complexidade e o tamanho da sua equipe, aumenta a dificuldade de gerenciá-los e de acompanhar seus prazos e custos.

Mediante as afirmativas apresentadas pode-se concluir que a gestão de projetos e produtos de software somente alcança um nível adequado de eficácia e exatidão se houver medidas que permitam acompanhar a situação do projeto. Estas medidas devem ser estimadas e acompanhadas durante todo o curso do projeto.

## **1.2 Objetivo**

Neste trabalho é proposto um modelo de gerência de eventos para um ambiente de desenvolvimento de software centrado em processo e a utilização deste modelo para um efetivo acompanhamento de projetos.

O objetivo desta abordagem é automatizar algumas funções gerenciais e disponibilizar ao gerente de projetos relatórios de acompanhamento baseados na literatura e em abordagens como CMMI, MPS.Br e PMBOK.

A principal contribuição esperada é a definição de um método de acompanhamento de projetos que faz uso de um ambiente de gestão de processos

de software, o webAPSEE<sup>1</sup>, utilizando uma ferramenta acoplada a este ambiente possibilitando ao gerente realizar esta atividade atendendo as principais abordagens de melhoria e gestão de processo.

Ao final deste trabalho será realizada uma análise do atendimento dos requisitos e uma comparação com ferramentas semelhantes à abordagem apresentada neste trabalho.

### 1.3 Metodologia de Trabalho

Para atender os objetivos deste trabalho as etapas abaixo foram realizadas:

- **Estudo do Ambiente e Proposta do Gerenciador de Eventos.** Primeiramente foi realizado estudo do ambiente de gestão de processo webAPSEE, com o objetivo de verificar possíveis pontos de melhoria no gerenciador de eventos do ambiente e o resultado deste trabalho foi uma proposta de extensão do mecanismo de gerência de eventos do ambiente.
- **Estudo das Atividades de Acompanhamento de Projeto.** Levantamento dos Requisitos de Acompanhamento de Projetos propostos por abordagens como CMMI, MPS.Br e PMBOK.
- **Seleção dos Requisitos e Proposta da Ferramenta.** Após a identificação dos Requisitos, foi verificado como o gerenciador de eventos proposto anteriormente poderia atendê-los e como resultado foi proposta uma ferramenta de acompanhamento de projetos que visa atender os requisitos identificados.
- **Avaliação do Atendimento dos Requisitos.** Para avaliar o atendimento dos requisitos foi realizada uma análise da ferramenta, dos seus requisitos e foi realizada uma comparação com ferramentas que possuem o mesmo propósito.

### 1.4 Organização

Além deste capítulo introdutório, a dissertação está organizada da forma descrita a seguir.

---

<sup>1</sup> webAPSEE é um ambiente de desenvolvimento de Software Centrado em Processo (PSEE - Process-Centered Software Engineering Environment) desenvolvido como Software Livre pelo LABES-UFPA. O ambiente tem como finalidade principal atender a requisitos organizacionais para auxiliar na coordenação das atividades relacionadas com o desenvolvimento de software.

O Capítulo 2 apresenta os principais modelos de qualidade de software e gerenciamento de projetos de software atualmente existente, as práticas e atividades preconizadas pelos modelos e o que deve ser feito para atendê-los.

O Capítulo 3 apresenta uma proposta para atendimento dos requisitos identificados na seção 2, como atividades essenciais ao acompanhamento de projetos de software.

O capítulo 4 apresenta o protótipo desenvolvido de acordo com a proposta do capítulo 3 e apresentado com intuito de auxiliar na avaliação deste trabalho. A aplicação foi desenvolvida e incorporada a um ambiente de engenharia software centrada em processos. A aplicação desenvolvida e o ambiente webAPSEE possibilitam que o suporte efetivo as atividades de monitoração e controle de projetos.

O capítulo 5 apresenta uma análise da proposta desta dissertação, através da simulação da execução de um projeto real e utilização da ferramenta proposta. Uma avaliação das características da ferramenta e atendimento aos requisitos descritos na seção 2, através também é apresentada neste capítulo.

Por fim, no Capítulo 6 são apresentadas às conclusões do trabalho, descrevendo suas contribuições, extensões, e melhorias vislumbradas para trabalhos futuros.

# Capítulo 2 – Qualidade e Gerenciamento de Projetos de Software

---

Este capítulo tem o objetivo de apresentar os principais modelos de qualidade de software e gerenciamento de projetos de software existente atualmente, as práticas e atividades preconizadas pelos modelos e o que deve ser feito para atendê-los.

## 2.1 CMMI – Capability Maturity Model Integration

Visando a melhoria do desenvolvimento de software, vários modelos para avaliação do processo de produção de software têm sido propostos por instituições no mundo inteiro. Dentre os mais utilizados, pode-se citar o CMMI do *Software Engineering Institute* (SEI) o qual tem sido bastante utilizado pelas empresas de software, tais como EDS Wireless Resource Center, Motorola, Boeing, IBM, Bellcore, entre outras [SEI, 2007] [Waina, 2002].

O CMMI é um modelo norte-americano que fornece às organizações diretrizes para controlar seus processos de desenvolvimento de software, de modo a desenvolver e manter software de melhor qualidade, bem como instituir uma cultura de excelência em engenharia e gerenciamento de projetos de software.

O modelo SW-CMM (*Software Capability Maturity Model*) foi definido no SEI a pedido do Departamento de Defesa dos Estados Unidos. A partir de 1991, foram desenvolvidos CMMs para várias disciplinas (Engenharia de Sistemas, Engenharia de Software, Aquisição de Software, Gerência e Desenvolvimento da Força de Trabalho, Desenvolvimento Integrado do Processo e do Produto). Embora estes modelos tenham mostrado sua utilidade, o uso de múltiplos modelos se mostrou problemático. O CMMI surgiu para resolver o problema da utilização de vários modelos e é o resultado da evolução do SW-CMM, SECM (*System Engineering Capability Model*) e IPD-CMM (*Integrated Product Development Capability Maturity Model*). Cabe ressaltar que o framework CMMI foi desenvolvido para ser consistente e compatível com a ISO/IEC 15504 [SEI, 2002].

Existem dois tipos de representação no *framework* CMMI: em estágios e contínua. Tem-se, assim, um único modelo que pode ser visto de duas perspectivas distintas. A representação em estágios é a representação usada no SW-CMM. Esta representação define um conjunto de áreas de processo para definir um caminho de melhoria para a unidade organizacional, descrito em termos de níveis de maturidade.

O CMMI propõe um caminho gradual, através dos níveis de maturidade da capacitação, que leva as organizações a se aprimorarem continuamente na busca das suas próprias soluções para os problemas inerentes ao desenvolvimento sistemático de software. A capacitação aqui mencionada refere-se à habilitação que a organização tem em sistematicamente produzir software com a qualidade esperada, dentro dos prazos acordados e com os recursos estimados [Fiorini, 1998].

A estrutura do CMMI consiste de cinco (1 a 5) níveis de maturidade, conforme ilustrado na Figura 1. Os níveis estabelecem práticas e processos que devem ser seguidos pela organização.

As sub-seções a seguir detalham os níveis identificados na figura 1. As informações apresentadas foram obtidas a partir de CMMI (2006).



Figura 1. Os 5 Níveis de Maturidade do CMMI

### 2.1.1. Nível de Maturidade 1: Inicial

No nível de maturidade 1 do CMMI, os processos são informais e caóticos. A organização normalmente não possui um ambiente estável. O sucesso destas organizações depende da competência e heroísmo das pessoas e não do uso de

processos comprovados. Apesar deste ambiente informal e caótico, organizações de nível 1 de maturidade muitas vezes produzem produtos e serviços que funcionam; entretanto, elas freqüentemente excedem o orçamento e o cronograma de seus projetos.

As organizações de maturidade nível 1 são caracterizadas por uma tendência a não cumprir compromissos, abandonar processos em momentos de crises e não ser capazes de repetir sucessos do passado.

### **2.1.2. Nível de Maturidade 2: Gerenciado**

No nível de maturidade 2, uma organização atingiu todas as metas específicas e genéricas das áreas de processos do nível 2 de maturidade. Em outras palavras, os projetos da organização asseguraram que os requisitos são gerenciados e que os processos são planejados, executados, medidos e controlados.

A disciplina dos processos refletida pelo nível 2 de maturidade ajuda a assegurar que as práticas existentes são mantidas em momentos de stress. Quando estas práticas existem, os projetos são executados e gerenciados de acordo com seus planos documentados.

No nível 2 de maturidade, os requisitos, processos, produtos de trabalho e serviços são gerenciados. A situação dos produtos de trabalho e a entrega dos serviços são visíveis para o gerenciamento em pontos definidos (por exemplo, nos *milestones*<sup>2</sup> principais e no momento em que as principais tarefas são completadas).

Compromissos são estabelecidos entre os *stakeholders*<sup>3</sup> „relevantes e são revistos conforme necessário. Os produtos de trabalho são revisados com os *stakeholders* e controlados. Os produtos de trabalho e serviços satisfazem seus requisitos, padrões e objetivos definidos.

### **2.1.3. Nível de Maturidade 3: Definido**

No nível de maturidade 3, uma organização atingiu todas as metas específicas e genéricas das áreas de processos definidas para os níveis de

---

<sup>2</sup> Milestone pode ser definido como pontos de checagem ou marcos de desenvolvimento.

<sup>3</sup> Stakeholder pode ser definido como parte interessada ou interveniente, refere-se a todos os envolvidos em um processo, por exemplo, clientes, colaboradores, investidores, fornecedores, comunidade, etc.

maturidade 2 e 3. No nível de maturidade 3, os processos são bem caracterizados e entendidos e estão descritos em padrões, procedimentos, ferramentas e métodos.

O conjunto de processos padrão da organização, que é a base para o nível de maturidade 3, é estabelecido e melhorado ao longo do tempo. Estes processos padrão são usados para estabelecer a consistência em toda a organização. Os projetos estabelecem seus processos definidos adaptando o conjunto de processos padrão da organização de acordo com as instruções de adaptação.

O gerenciamento da organização estabelece os objetivos dos processos com base no conjunto de processos padrão da organização e assegura que estes objetivos estão sendo tratados de forma adequada.

Uma diferença crucial entre os níveis de maturidade 2 e 3 é o escopo de padrões, descrições de processos e procedimentos. No nível de maturidade 2, os padrões, descrições de processos e procedimentos podem ser bastante diferentes em cada instância do processo (por exemplo, em um projeto específico). No nível de maturidade 3, os padrões, descrições de processos e procedimentos para um projeto são adaptados do conjunto de processos padrão da organização para se adequar a um projeto ou unidade organizacional específicos. O conjunto de processos padrão da organização inclui os processos tratados nos níveis de maturidade 2 e 3. Conseqüentemente, os processos que são executados em toda a organização são consistentes, exceto pelas diferenças permitidas pelas instruções de adaptação.

Outra diferença crítica é que no nível de maturidade 3, os processos são geralmente descritos mais detalhadamente e com maior rigor que no nível de maturidade 2. No nível de maturidade 3, os processos são gerenciados de forma mais pró-ativa, utilizando um entendimento dos inter-relacionamentos das atividades de processos e medidas detalhadas do processo, seus produtos de trabalho e seus serviços.

#### **2.1.4. Nível de Maturidade 4: Gerenciado Quantitativamente**

No nível de maturidade 4, uma organização atingiu todas as metas específicas das áreas de processos atribuídas aos níveis de maturidade 2, 3 e 4 e as metas genéricas atribuídas aos níveis de maturidade 2 e 3. São selecionados os subprocessos que contribuem significativamente para o desempenho geral do

processo. Estes subprocessos selecionados são controlados utilizando estatísticas e outras técnicas quantitativas.

Os objetivos quantitativos para a qualidade e o desempenho dos processos são estabelecidos e utilizados como critérios para o gerenciamento de processos. Os objetivos quantitativos são baseados nas necessidades dos clientes, usuários finais, da organização e dos responsáveis pela implantação dos processos. A qualidade e o desempenho do processo são entendidos em termos estatísticos e são gerenciados durante toda a vida dos processos.

Para estes processos, são coletadas e analisadas de forma estatística, medidas detalhadas de desempenho de processos. Causas especiais de variações de processos são identificadas e, quando apropriado, as fontes das causas especiais são corrigidas para evitar ocorrências futuras.

Medidas de qualidade e desempenho de processos são incorporadas ao repositório de medições da organização para dar suporte a futuras decisões baseadas em fatos ocorridos.

Uma importante diferença entre os níveis de maturidade 3 e 4 é a previsibilidade do desempenho do processo. No nível de maturidade 4, o desempenho dos processos é controlado utilizando estatística e outras técnicas quantitativas e este é previsível de forma quantitativa. No nível de maturidade 3, os processos são previsíveis apenas de forma qualitativa.

#### **2.1.5. Nível de Maturidade 5: Otimizado**

No nível de maturidade 5, uma organização atingiu todas as metas específicas das áreas de processos atribuídas aos níveis de maturidade 2, 3, 4 e 5 e as metas genéricas atribuídas aos níveis de maturidade 2 e 3. Os processos são continuamente melhorados com base em um entendimento quantitativo das causas comuns de variações inerentes aos processos.

O nível de maturidade 5 se concentra no melhoramento contínuo do desempenho de processos através de melhorias tecnológicas incrementais e inovadoras. Os objetivos quantitativos de melhoria de processos para a organização são estabelecidos, continuamente revisados para refletir alterações nos objetivos do negócio e utilizados como critérios para o gerenciamento da melhoria de processos. Os efeitos das melhorias de processos aplicadas são medidos e avaliados contra os

objetivos quantitativos de melhoria de processos. Tanto os processos definidos como o conjunto de processos padrão da organização são alvos de atividades de melhoria mensuráveis.

As melhorias de processos que tratam as causas comuns de variações de processos e melhoram de forma mensurável os processos da organização são identificadas, avaliadas e aplicadas. As melhorias são selecionadas com base em um entendimento quantitativo de sua contribuição esperada para que a organização atinja seus objetivos de melhoria de processos contra o seu custo e impacto na organização. O desempenho dos processos da organização é continuamente melhorado.

### 2.1.6. Comparando os níveis de maturidade do CMMI

Com exceção do Nível 1, cada nível de maturidade é decomposto em áreas de processo, que conduzem ao alcance de metas de melhoria do processo, indicando os pontos que as organizações devem focar para melhorar seu processo de software.

Em cada área de processo, as metas e práticas específicas são listadas em primeiro lugar, seguidas pelas metas e práticas genéricas. Os componentes do modelo CMMI são ilustrados na Figura 2 e detalhados em seguida com o objetivo de fornecer maior entendimento dos componentes do modelo CMMI.

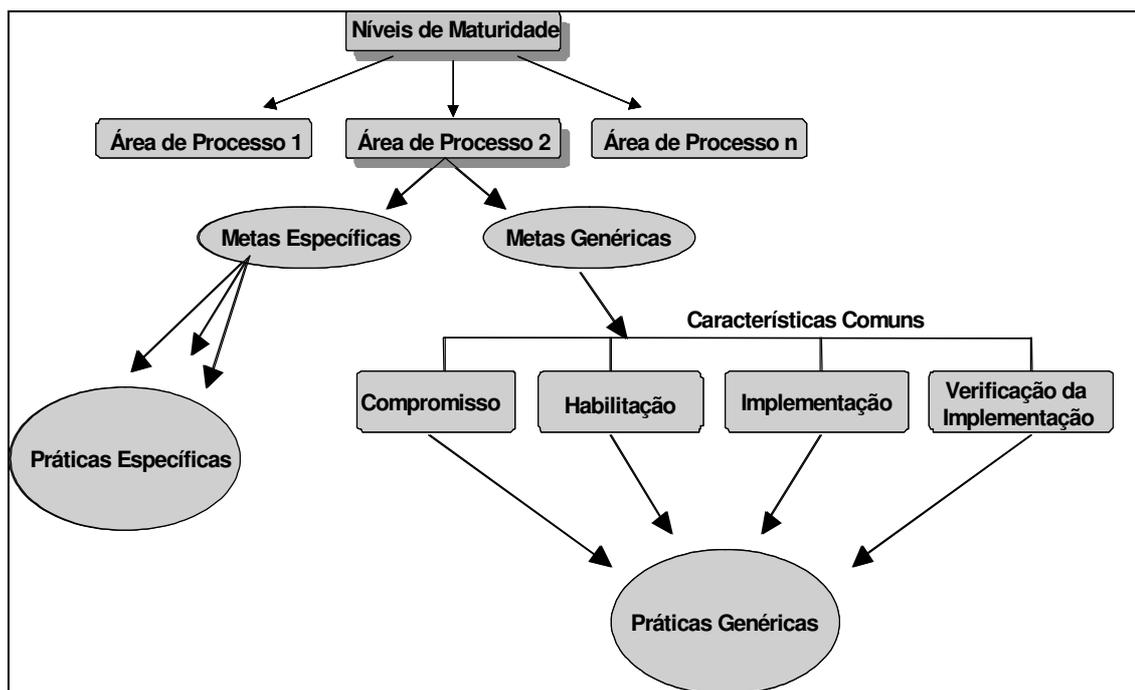


Figura 2. Componentes do Modelo CMMI

- **Nível de Maturidade**

Um nível de maturidade é uma etapa evolucionária definida da melhoria de processos. Cada nível de maturidade estabiliza uma parte importante dos processos da organização.

- **Áreas de Processos**

Uma área de processo é um grupo de práticas relacionadas em uma área que, quando executadas de forma coletiva, satisfazem um conjunto de metas consideradas importantes para trazer uma melhoria significativa naquela área.

- **Metas Específicas**

As metas específicas se aplicam a uma área de processo e tratam de características únicas que descrevem o que deve ser implementado para satisfazer a área de processo. Metas específicas são componentes exigidos do modelo e são utilizadas nas avaliações para auxiliar a determinar se a área de processo está sendo satisfeita.

- **Práticas Específicas**

Uma prática específica é uma atividade que é considerada importante na satisfação de uma meta específica associada. As práticas específicas descrevem as atividades que se espera que resultem no atendimento de metas específicas de uma área de processo. As práticas específicas são componentes esperados do modelo.

- **Características Comuns**

Quatro características comuns organizam as práticas genéricas de cada área de processo. As características comuns são componentes de modelo que não estão classificados. Elas são somente agrupamentos que oferecem uma maneira de apresentar as práticas genéricas.

- Compromisso
- Habilitação
- Implementação
- Verificação da Implementação

A tabela 1 apresenta os níveis de maturidade e as áreas de processo que o compõem.

Tabela 1. Áreas de Processo do CMMI

<b>Nível de Maturidade</b>	<b>Área de Processo</b>
<b>Nível 2 Gerenciado</b>	Gerenciamento de Requisitos Planejamento do Projeto Monitoramento e Controle do Projeto Gerenciamento de Acordos com Fornecedores Medições e Análises Garantia da Qualidade do Processo e do Produto Gerenciamento de Configurações
<b>Nível 3 Definido</b>	Desenvolvimento de Requisitos Soluções Técnicas Integração de Produtos Verificação Validação Foco no Processo Organizacional Definição do Processo Organizacional Treinamento Organizacional Gerenciamento Integrado do Projeto Gerenciamento de Riscos Análises de Decisões e Resoluções
<b>Nível 4 Quantitativamente Gerenciado</b>	Gerenciamento Quantitativo do Processo Desempenho do Processo Organizacional
<b>Nível 5 Otimização</b>	Análise de Causa e Resolução Inovação e Implantação na Organização

Como objetivo principal deste trabalho optou-se por apoiar a área de processo de Monitoramento e Controle do Projeto, pertencente ao nível 2 de maturidade do CMMI.

## **2.2 MPS.Br – Melhoria do Processo de Software Brasileiro**

O MPS.Br é um programa para Melhoria de Processo do Software Brasileiro, está em desenvolvimento desde dezembro de 2003 e é coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX). O MPS.Br baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de produtos de software e serviços correlatos.

O Modelo de Referência denominado MR-MPS define níveis de maturidade que são uma combinação entre processos e sua capacidade, de forma análoga ao estabelecido pelo CMMI. A definição dos processos segue a forma apresentada na Emenda 1 da ISO/IEC 12207, declarando o propósito e os resultados esperados de sua execução. Isso permite avaliar e atribuir graus de efetividade na execução dos processos.

A capacidade do processo é a caracterização da habilidade do processo para alcançar os objetivos de negócio, atuais e futuros; estando relacionada com o atendimento aos atributos de processo associados aos processos de cada nível de maturidade.

Os níveis de maturidade estabelecem patamares de evolução de processos, caracterizando estágios de melhoria da implementação de processos na organização. O nível de maturidade em que se encontra uma organização permite prever o seu desempenho futuro ao executar um ou mais processos. O MR-MPS define sete níveis de maturidade: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). A escala de maturidade se inicia no nível G e progride até o nível A.

A tabela 2 apresenta os níveis de maturidade e os processos que o compõem.

Tabela 2. Áreas de Processo do MPS.Br

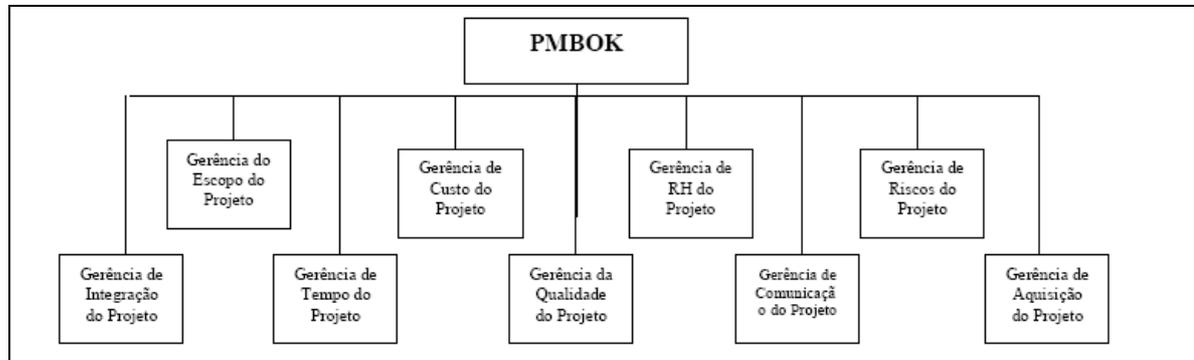
<b>Nível de Maturidade</b>	<b>Processos</b>
<b>A</b>	Análise de Causas e Resolução
<b>B</b>	Gerência de Projetos (Evolução)
<b>C</b>	Gerência de Riscos Análise de Decisão e Resolução Desenvolvimento para Reutilização
<b>D</b>	Desenvolvimento de Requisitos Integração do Produto Projeto e Construção do Produto Validação Verificação
<b>E</b>	Avaliação e Melhoria do Processo Organizacional Definição do Processo Organizacional Gerência de Recursos Humanos Gerência de Reutilização
<b>F</b>	Medição Gerência de Configuração Aquisição Garantia da Qualidade
<b>G</b>	Gerência de Requisitos Gerência do Projeto

O escopo deste trabalho visa apoiar o processo de gerência de projetos, pertencente ao nível G de maturidade do MPS.Br.

### **2.3 PMBOK – Project Management Body of Knowledge**

O PMBOK é um guia que aborda processos, áreas de conhecimento, técnicas, regras e métodos para o gerenciamento de projetos genéricos [PMBOK, 2004]. Este guia prevê que o projeto será dividido em etapas e estas fazem parte do ciclo de vida [Prado, 2000]. Cada etapa é formada por processos das nove áreas de

conhecimento, como mostra a Figura 3. As gerências serão explicadas a seguir, usando definições dispersas em [PMBOK, 2004][Vargas, 2003][Prado 2000].



**Figura 3. Gerências do PMBOK**

Na Gerência de Escopo estão todos os processos necessários para o desenvolvimento do produto conforme as suas características. O que não estiver explícito no escopo não será realizado. Normalmente um projeto é iniciado por uma demanda de mercado, uma necessidade de negócio, um pedido do cliente, um avanço tecnológico ou uma exigência legal.

A Gerência de Tempo é responsável por gerar o cronograma do projeto. A Gerência de Custos do Projeto possui os processos para assegurar que o projeto será concluído dentro do orçamento aprovado. A Gerência da Qualidade tem como principal meta garantir que os objetivos serão alcançados com a qualidade necessária.

A Gerência de Recursos Humanos tem como principal objetivo utilizar, da melhor forma, possível a potencialidade dos componentes da equipe do projeto. Atualmente, os aspectos humanos são o foco de estudos e trabalhos na área, de modo a compensar o foco nos aspectos técnicos do passado.

A Gerência das Comunicações do Projeto inclui os processos necessários para que as informações sejam geradas, distribuídas e armazenadas de uma forma adequada. Todos os envolvidos do projeto devem estar preparados para enviar e receber informações e ainda compreender como as suas informações afetam o projeto como um todo.

A Gerência de Riscos tem a função de identificar, analisar e responder aos riscos do projeto. A Gerência de Aquisições do Projeto tem como objetivo garantir que elementos externos sejam fornecidos para o projeto. Para isso deve-se determinar o que contratar e quando, documentar os requisitos do produto e identificar os fornecedores, obter propostas de fornecimento, fazer a escolha do fornecedor, gerenciar o relacionamento e encerrar o contrato quando necessário.

O escopo deste trabalho visa apoiar a gerência de custos e tempo definidos no PMBOK.

## **2.4 Acompanhamento de Processos no CMMI, MPS-Br e PMBOK.**

Nesta seção serão apresentadas as práticas e processos que devem ser seguidos com o objetivo de monitorar e acompanhar projetos de software e que são definidas nas principais abordagens de melhoria de processo e gerenciamento de projetos apresentadas nas seções anteriores.

Deve-se ressaltar que os modelos apresentados são guias de referência que estabelecem requisitos a serem atendidos, porém não indicam como as práticas indicadas devem ser realizadas.

### **2.4.1 Acompanhamento de Projetos no CMMI**

Para o CMMI o objetivo do Monitoramento e Controle do Projeto é fornecer um entendimento do progresso do projeto de forma que as ações corretivas apropriadas possam ser tomadas quando o desempenho do projeto se desviar significativamente do plano

Para a área de processo de Acompanhamento de projetos o CMMI preconiza algumas práticas que devem ser seguidas. A seguir são apresentadas as principais práticas e sub-práticas na área de processo de acompanhamento de projetos.

#### **Prática 1 - Monitorar os Parâmetros de Planejamento do Projeto**

Os parâmetros de planejamento do projeto constituem os indicadores típicos do desempenho e progresso do projeto e incluem os atributos dos produtos de trabalho e tarefas, custo, esforço e cronograma. Atributos dos produtos de trabalho e tarefas incluem itens como tamanho, complexidade, peso, formato, encaixe ou funções.

O monitoramento normalmente envolve medir os valores reais dos parâmetros de planejamento do projeto, comparar os valores reais com os estimados no plano e identificar os desvios significativos. Registrar os valores reais dos parâmetros de planejamento do projeto inclui registrar as informações contextuais associadas para auxiliar no entendimento das medidas. São Produtos de Trabalho Típicos

- Registros de desempenho do projeto
- Registros de desvios significativos

Sub-práticas

1. Monitorar o progresso contra o cronograma.

O monitoramento do progresso normalmente inclui:

- Medir periodicamente o nível real de atividades.
- Comparar o nível real de atividades e milestones completados contra o cronograma documentado no plano do projeto
- Identificar os desvios significativos das estimativas de cronograma no plano do projeto.

**Prática 2. Monitorar os custos e esforços gasto no projeto.**

Atender a prática de monitorar o esforço e o custo normalmente inclui as seguintes atividades:

- Medir periodicamente o esforço e custo real gasto e o pessoal designado
- Comparar o esforço, custo, pessoal e treinamento reais com as estimativas e orçamentos documentados no plano do projeto
- Identificar desvios significativos dos orçamentos no plano do projeto

#### **2.4.2 Acompanhamento de Projetos no MPS.Br**

Assim como o CMMI, o MPS.Br também detalha alguns propósitos que devem ser seguidos no acompanhamento de projetos. A seguir estes são apresentados.

**Propósito 1 - O cronograma e o orçamento do projeto são estabelecidos e mantidos.**

As dependências entre tarefas são estabelecidas e potenciais gargalos identificados através de métodos apropriados (por exemplo, análise de caminho crítico). Os gargalos são resolvidos quando possível, e o cronograma das atividades com início, duração e término é estabelecido. Recursos requeridos são refletidos nos custos estimados. Com base na EAP e nas estimativas de esforço e custo, deve ser definido o cronograma, considerando as dependências entre as tarefas. É importante ter-se o cuidado de manter a coerência entre ciclo de vida, EAP, estimativas e cronogramas.

Com base no cronograma e na estimativa de custos, é estabelecido o orçamento do projeto. Este resultado é importante porque o cronograma e o orçamento são instrumentos fundamentais para o acompanhamento do dia-a-dia do projeto. Desta forma, sempre que necessário devem ser revistos e atualizados.

**Propósito 2 - O Planejamento do projeto é monitorado no que se refere ao cronograma, custos, recursos, riscos, envolvimento dos interessados e dados.**

A aderência aos diversos planos deve ser avaliada continuamente durante todo o ciclo de vida do projeto. Os resultados e os critérios de conclusão de cada tarefa são analisados, as entregas são avaliadas nos termos de suas características (através de revisões e auditorias, por exemplo), a dispensa de esforços e a aderência ao cronograma são investigadas, e o uso dos recursos é examinado. Análises devem ser realizadas e decisões serem tomadas considerando-se as variações dos dados e desvios entre resultados e valores atuais e esperados. O acompanhamento pode ser feito utilizando ferramentas de planejamento, em que se pode examinar o previsto contra o realizado, usando indicadores de progresso, cumprimento de marcos, entre outros. O acompanhamento também pode ser feito através de reuniões e comunicação pessoal. Contudo, é importante ressaltar que devem existir registros desses acompanhamentos. Esta é uma atividade essencial de gerenciamento: acompanhar o que foi planejado, detectar problemas e corrigi-los.

### **2.4.3 Acompanhamento de Projetos no PMBOK**

Para o PMBOK monitorar e controlar um processo constitui-se na atividade de observar a execução do projeto, de forma que possíveis problemas possam ser identificados no momento adequado e que possam ser tomadas ações corretivas, quando necessário, para controlar a execução do projeto. O principal benefício deste grupo de processo é que o desempenho do projeto é observado e medido regularmente para identificar variações em relação ao plano de gerenciamento do projeto.

O monitoramento contínuo permite que a equipe do projeto tenha uma visão clara da saúde do projeto e destaca as áreas que exigem atenção adicional. O grupo de processos de monitoramento e controle, além de monitorar e controlar o trabalho que está sendo realizado dentro de um grupo de processos, também monitora e controla todo o esforço do projeto. Quando as variações comprometem os objetivos do projeto, o planejamento deve ser reexaminado e ações corretivas devem ser tomadas. Por exemplo, uma data de término de atividade não cumprida pode exigir um aumento na equipe atual, dependência de horas extras ou compensações entre os objetivos de orçamento e de cronograma.

## **2.5 Considerações Finais**

Neste capítulo foi apresentado o CMMI, modelo para avaliação de processo de desenvolvimento de software proposto pelo Software Engineering Institute (SEI) e o MPS.Br. Estes modelos fornecem às organizações diretrizes para controlar seus processos de desenvolvimento de software, desenvolvendo uma cultura de excelência em engenharia e gerenciamento de projetos de software.

Também foi apresentado o PMBOK que é um guia que aborda processos, áreas de conhecimento, técnicas, regras e métodos para o gerenciamento de projetos genéricos.

Finalizando este capítulo abordou a atividade de acompanhamento de projetos e as práticas indicadas para esta atividade. O monitoramento e controle de projetos é o foco deste trabalho e um modelo proposto para esta atividade será apresentado na próxima seção.

# Capítulo 3 – Modelo Para Acompanhamento de Projetos de Software

---

Este capítulo apresenta uma proposta para atendimento dos requisitos identificados na seção 2, através da associação das necessidades apresentadas, com as funcionalidades fornecidas pela abordagem utilizada neste trabalho.

## 3.1 Introdução

As principais abordagens de melhoria de processo e gerenciamento de projetos indicam que o gerente de um projeto deve acompanhá-lo, realizando constantes medições dos custos, prazos e esforço, comparando com os valores estimados, analisando os desvios e tomando as ações corretivas quando necessárias.

O processo de acompanhamento de projetos descrito no capítulo 2 é resumido na Figura 4. Para modelagem deste processo utilizou-se a notação do ambiente WebAPSEE cujo detalhamento é apresentado no Apêndice B.

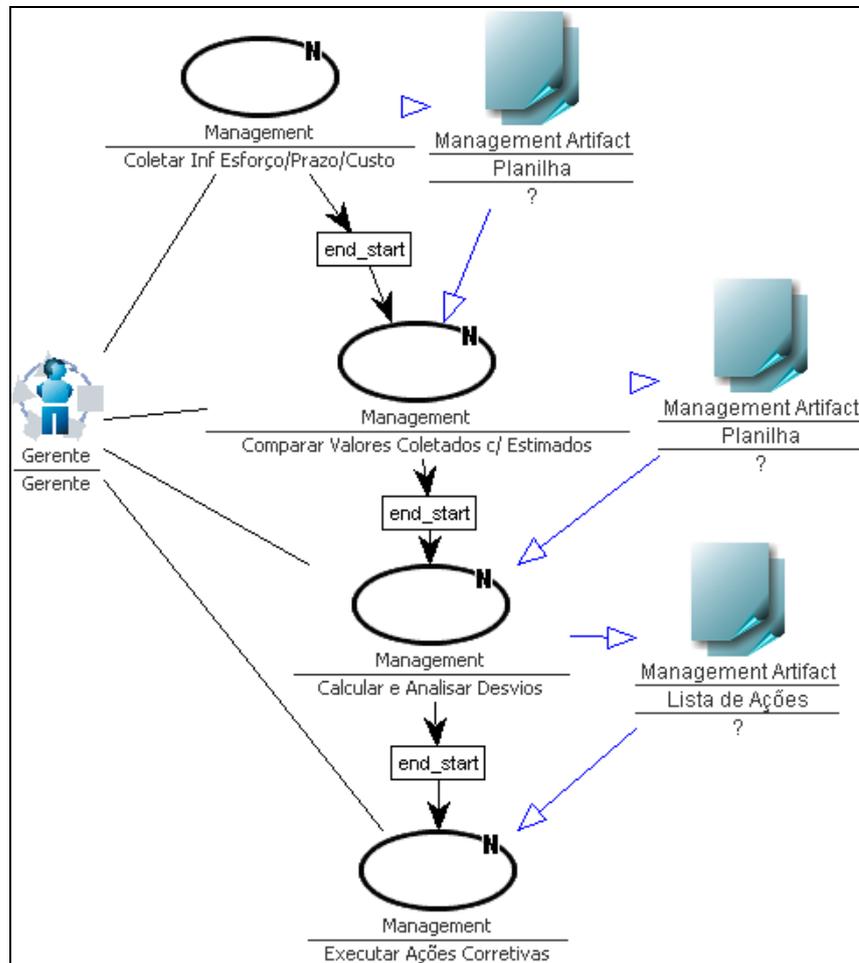
O fluxo seguido é detalhado a seguir:

1. O Gerente de Projetos deve coletar as informações de Prazo, Custo e Esforço gasto no projeto. Esta atividade possui como saída um artefato do tipo planilha que deve conter os valores obtidos.

2. O Gerente de Projetos compara os valores coletados, resultantes da execução do processo, com os valores estimados. Esta atividade possui como saída uma planilha atualizada com os valores estimados e realizados.

3. O Gerente de Projetos Calcula e Analisa os desvios de estimativas. Esta atividade possui como saída lista de ações elaborada a partir da detecção de atividades com problemas. São ditas atividades com problemas aquelas que apresentarem desvios maiores que os aceitáveis pela organização;

4. O gerente executa as ações previstas na lista de ações, resultante da atividade de Cálculo e Análise dos Desvios.



**Figura 4. Modelo de Acompanhamento de Projetos**

Tomando por base estas indicações, este trabalho apresenta uma abordagem para monitoração e controle de projetos. A seção 3.2 apresenta a proposta de acompanhamento fornecida neste trabalho. A seção 3.3 apresenta as necessidades identificadas, as funcionalidades propostas para atendimento e os requisitos não funcionais. A seção 3.4 apresenta um modelo para atendimento destas necessidades e a seção 3.5 apresenta as considerações finais.

### 3.2 Acompanhamento de Projetos

Conforme exposto anteriormente, este trabalho propõe uma extensão ao ambiente de gestão de processos webAPSEE para atender os requisitos de acompanhamento de projetos de software descritos no capítulo 2, baseados em modelos de maturidade de processo como CMMI e MPS.Br. Como o objetivo destes modelos não é somente a automação de processos, e sim indicar as boas práticas que as

organizações devem seguir, alguns dos requisitos definidos para a atividade de acompanhamento de projetos não estavam contemplados no ambiente.

O módulo de acompanhamento de processo proposto neste trabalho, **Monitoring Process**, visa atender as práticas identificadas na seção 2 incorporando algumas melhorias no ambiente **WebAPSEE**. A Figura 5 apresenta o fluxo do processo proposto para acompanhamento de processos.

O gerente de processos deve modelar o processo e realizar estimativas. As atividades podem ser realizadas concorrentemente. Após completar estas atividades deve executar o processo e a partir deste modelo pode já começar a acompanhar o processo até a finalização do mesmo, através da utilização do módulo de monitoramento de processos. Após o término da execução dos processos os relatórios continuam sendo emitidos, com o objetivo de apoiar o gerente na avaliação do processo e na realização de estimativas para futuros projetos.

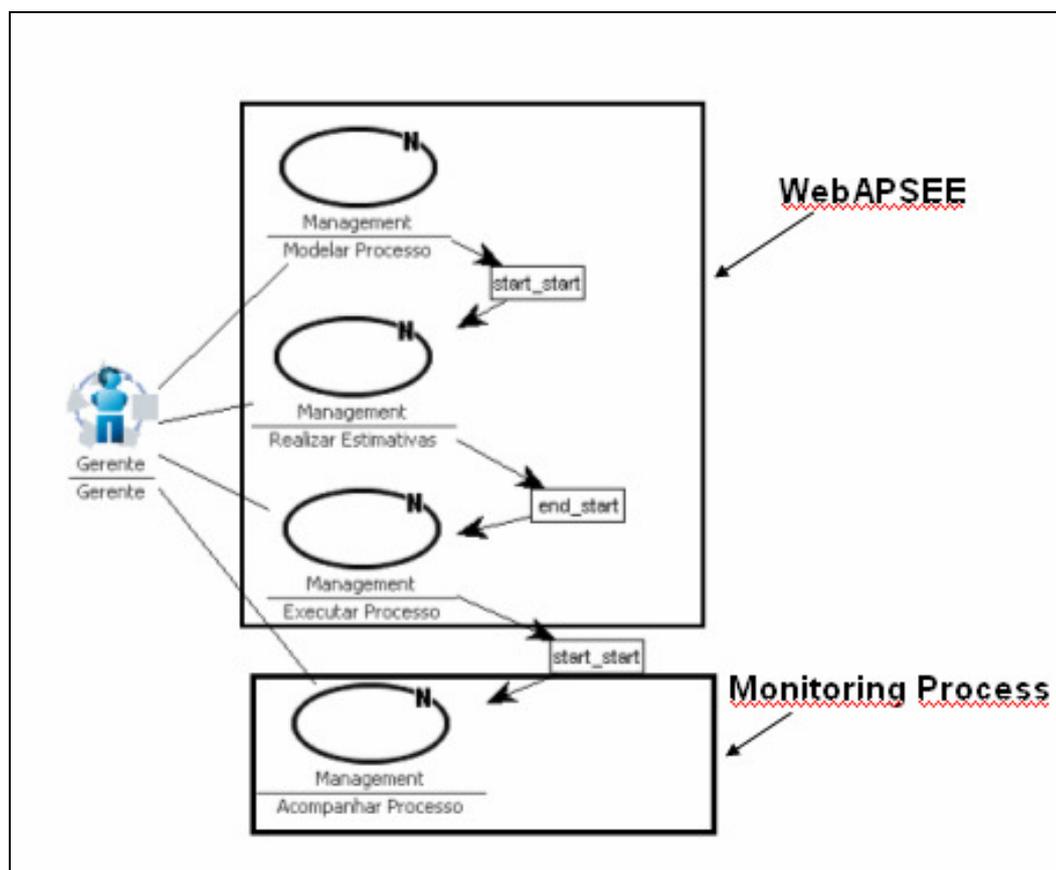
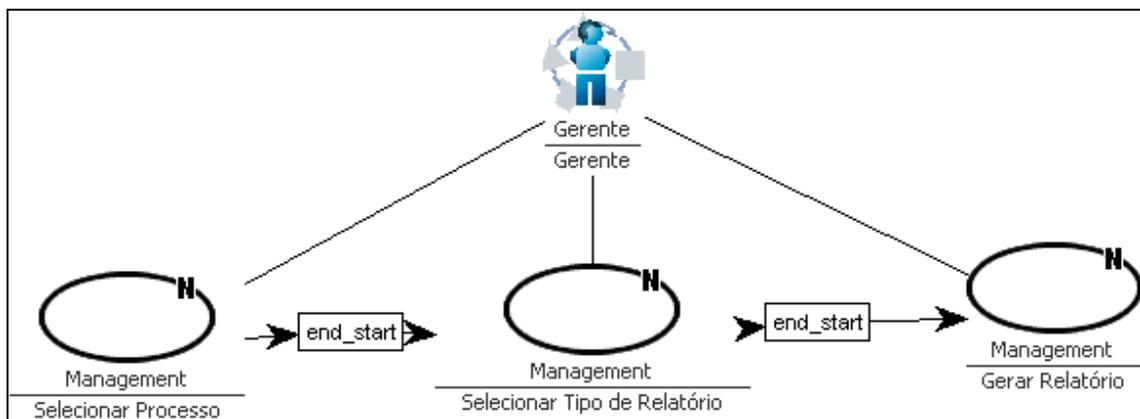


Figura 5. Atividade Acompanhamento de Processo

A seguir é feita uma descrição detalhada das atividades que devem ser seguidas segundo a abordagem de Monitoração de Processo.

1. O gerente acessa o ambiente webAPSEE, modela o processo de software, cadastrando as atividades, as dependências entre elas, alocando os recursos e agentes necessários para a realização de cada atividade.
2. Para cada agente alocado informa o custo por hora dos agentes. Este passo é necessário para a geração dos relatórios de acompanhamento de custo.
3. Para cada atividade do processo informa suas expectativas de esforço e prazo.
4. Para cada atividade do processo informa a data prevista de início e término das atividades.
5. Após a modelagem do processo e a execução do mesmo, o gerente pode realizar o acompanhamento do processo através do acesso ao módulo **Monitoring Process**, proposto neste trabalho, e disponível no Manager Console.

Para o acompanhamento de projetos o gerente deve selecionar o projeto sobre o qual deseja obter informações, selecionar o tipo de relatório e gerar o acompanhamento, conforme indicado na Figura 6.



**Figura 6. Gerar Relatório**

Os relatórios são gerados em formato de planilhas eletrônicas com o objetivo de facilitar ao gerente à manipulação das informações apresentadas.

### 3.3 Requisitos da Ferramenta

Nesta seção são apresentadas as necessidades identificadas a partir dos requisitos apresentados na seção 2 e alguns propostos pela autora.

O item apresentado no quadro 1 exibe o modelo adotado para descrição das necessidades e funcionalidades identificadas. O modelo é adaptado a partir da proposta de identificação de necessidades apresentado no *template* do artefato Documento de Visão, especificado pelo *Rational Unified Process* – RUP [RUP].

Quadro 1. Requisito “Necessidade Exemplo”

<b>Necessidade Exemplo</b>	
<< Descrição da Necessidade – O que o sistema deve oferecer >>	
<b>Id Func.</b>	<b>Descrição das Funcionalidades/atores envolvidos</b>
<b>F E.1</b>	<< Descrição das Funcionalidades – Como o sistema irá atender a necessidade identificada >>
	<< <b>Lista dos atores que utilizarão a funcionalidade</b> >>
<< Fonte das necessidades identificadas >>	

A partir do exemplo segue listagem dos requisitos identificados e que a abordagem proposta neste trabalho tem como objetivo atender.

Quadro 2. Requisito “Monitorar as Atividades do Projeto”

<b>Necessidade 1</b>	
Monitorar as Atividades do Projeto	
<b>Id Func.</b>	<b>Descrição das Funcionalidades/atores envolvidos</b>
<b>F1.1</b>	Emissão de Relatórios de acompanhamento de atividade, exibindo status atual de todas as atividades do projeto.
	<b>Gerente</b>
<b>F1.2</b>	Emissão de Relatórios de acompanhamento de atividade, exibindo os desvios de início e fim das atividades
	<b>Gerente</b>
<b>Fonte:</b> 1 - MPS. BR - Processo: Gerência de Projetos GPR 13. O progresso do projeto é monitorado com relação ao estabelecido no Plano do Projeto e os resultados são documentados; 2 – CMMI - Area de Processo: Monitoramento e Controle do Projeto SG 1 : Monitorar o Projeto Contra o Plano	

Quadro 3. Requisito “Monitorar os Custos do Projeto”

<b>Necessidade 2</b>	
Monitorar os Custos do Projeto	
<b>Id Func.</b>	<b>Descrição das Funcionalidades/atores envolvidos</b>
<b>F2.1</b>	Emissão de Relatórios de acompanhamento dos custos do projeto, comparando os valores estimados, com os valores realizados e apresentando os desvios identificados.
	<b>Gerente</b>
<p>Fonte:</p> <p>1 - MPS. BR - Processo: Gerência de Projetos                      GPR 5. O orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos;</p> <p>2 – CMMI - Area de Processo: Monitoramento e Controle do Projeto                      SG 1 : Monitorar o Projeto Contra o Plano                      SP 1.1: Monitorar os Parâmetros de Planejamento do Projeto</p>	

Quadro 4. Requisito “Monitorar os Prazos do Projeto”

<b>Necessidade 3</b>	
Monitorar os Prazos do Projeto	
<b>Id Func.</b>	<b>Descrição das Funcionalidades/atores envolvidos</b>
<b>F3.1</b>	Emissão de Relatórios de acompanhamento dos prazos do projeto, comparando os valores estimados, com os valores realizados e apresentando os desvios identificados.
	<b>Gerente</b>
<p>Fonte:</p> <p>1 - MPS. BR - Processo: Gerência de Projetos                      GPR 5. O orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos;</p> <p>2 – CMMI - Area de Processo: Monitoramento e Controle do Projeto                      SG 1 : Monitorar o Projeto Contra o Plano                      SP 1.1: Monitorar os Parâmetros de Planejamento do Projeto</p>	

Quadro 5. Requisito “Monitorar os Esforços no Projeto”

<b>Necessidade 4</b>	
Monitorar os Esforços no Projeto	
<b>Id Func.</b>	<b>Descrição das Funcionalidades/atores envolvidos</b>
<b>F4.1</b>	Emissão de Relatórios de acompanhamento dos esforços do projeto, comparando os valores estimados, com os valores realizados e apresentando os desvios identificados.
	<b>Gerente</b>
<p>Fonte:</p> <p>1 - MPS. BR - Processo: Gerência de Projetos                      GPR 5. O orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos;</p> <p>2 – CMMI - Area de Processo: Monitoramento e Controle do Projeto                      SG 1 : Monitorar o Projeto Contra o Plano                      SP 1.1: Monitorar os Parâmetros de Planejamento do Projeto</p>	

Quadro 6. Requisito “Identificar Possíveis Desvios do Projeto”

<b>Necessidade 5</b>	
Identificar Possíveis Desvios do Projeto	
<b>Id Func.</b>	<b>Descrição das Funcionalidades/atores envolvidos</b>
<b>F5.1</b>	Emissão de Relatórios que identifiquem desvios de estimativas do projeto quando estes excederem o limite de controle.
	Os relatórios devem ser emitidos em modo gráfico.
<b>Gerente</b>	
<p>Fonte:</p> <p>1 - MPS. BR - Processo: Gerência de Projetos                      GPR 17. Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão;</p> <p>2 – CMMI - Area de Processo: Monitoramento e Controle do Projeto                      SG 1 : Monitorar o Projeto Contra o Plano                      SP 1.1: Monitorar os Parâmetros de Planejamento do Projeto - Documentar os desvios significativos dos parâmetros de planejamento do projeto.</p>	

Quadro 7. Requisito “Indicar Atividades com Desvios de Estimativas”

<b>Necessidade 6</b>	
Indicar Atividades com Desvios de Estimativas.	
<b>Id Func.</b>	<b>Descrição das Funcionalidades/atores envolvidos</b>
<b>F6.1</b>	Exibir para o usuário atividades onde ações corretivas devem ser tomadas.
	<b>Gerente</b>
Fonte : Proposta da Autora	

Quadro 8. Requisito “Permitir o Acompanhamento dos projetos da Organização”

<b>Necessidade 7</b>	
Permitir o Acompanhamento de todos os projetos da Organização	
<b>Id Func.</b>	<b>Descrição das Funcionalidades/atores envolvidos</b>
<b>F7.1</b>	Permitir ao usuário selecionar qualquer processo que esteja em execução ou finalizado no ambiente.
	<b>Gerente</b>
Fonte : Proposta da Autora	

Quadro 9. Requisito “Permitir Fácil Manipulação dos Relatórios de Acompanhamento”

<b>Necessidade 8</b>	
Permitir Fácil Manipulação dos Relatórios de Acompanhamento	
<b>Id Func.</b>	<b>Descrição das Funcionalidades/atores envolvidos</b>
<b>F8.1</b>	Geração de relatórios em formato de planilhas eletrônicas com o objetivo de facilitar ao gerente à manipulação das informações apresentadas.
	<b>Gerente</b>
Fonte : Proposta da Autora	

Além das necessidades identificadas acima, também são apresentados os principais requisitos não funcionais que são necessários à ferramenta proposta neste trabalho.

Tabela 3. Requisitos Não Funcionais

<b>Requisito Não Funcional</b>	<b>Detalhamento</b>
<b>Funcionabilidade</b>	Os relatórios devem atender as funções e propriedades específicas identificadas, além de seguirem o padrão definido.
<b>Usabilidade</b>	A utilização do módulo <b>Monitoring Process</b> deve ser de fácil manuseio
<b>Eficiência</b>	Recurso e tempo envolvidos devem ser compatíveis com o nível de desempenho do software

### **3.4 Proposta de Atendimento dos Requisitos da Ferramenta**

A partir das necessidades apresentadas no item 3.3, identificou-se à necessidade de prover diferentes tipos de relatórios para o gerente. Além dos requisitos definidos na seção anterior, os relatórios propostos neste trabalho também foram baseados nos relatórios de acompanhamento de projetos utilizados pelo Serviço Federal de Processamento de Dados – SERPRO, disponibilizando os mesmos tipos de informações que os relatórios utilizados nesta organização. O SERPRO possui vários pólos de desenvolvimento de software, com certificação no nível 2 do CMM-SW e em vias de atender o nível 3 do CMMI. A autora deste trabalho é funcionária desta organização.

Para prover análise de desvios apresentando diferença entre os valores estimados e realizados, e exibindo estas informações para o gerente atendendo aos requisitos de usabilidade e as necessidades identificadas é proposto o padrão indicado na figura 7 – o qual mostra uma captura de tela da planilha eletrônica utilizada. As colunas da tabela apresentam as seguintes informações:

- A primeira coluna da tabela as atividades.
- A segunda mostra os desvios de estimativas.
- A terceira coluna exibe o valor realizado obtido a partir da execução das atividades no ambiente.

- A quarta coluna deve indicar o valor estimado informado no ambiente pelo gerente durante a modelagem do processo.

Além disso, este relatório também deve disponibilizar visualização gráfica dos desvios de estimativas para cada atividade.

Convém observar que o desvio é obtido a partir da fórmula:

$$Desvio = \left( \frac{Valor Realizado}{Valor Estimado} - 1 \right) \times 100$$

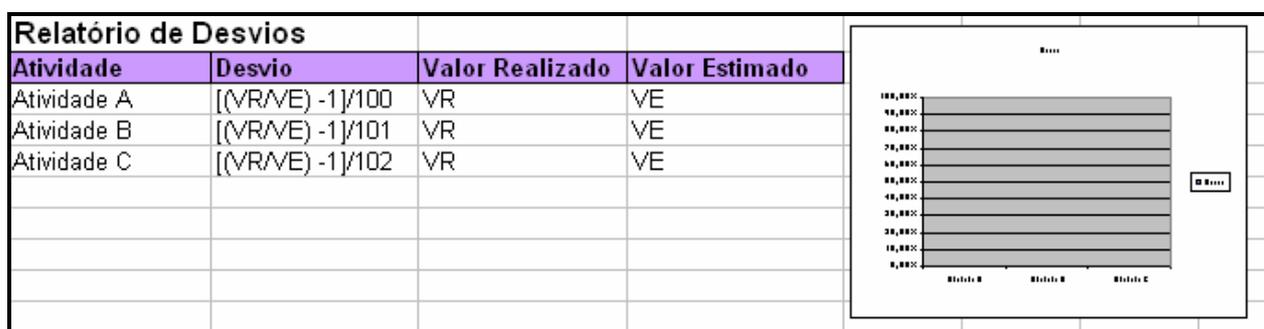


Figura 7. Layout Relatórios de Desvios

Além do Relatório de Desvios o ambiente deve fornecer relatórios mais concisos agrupando as atividades por tipos de atividades definidos no ambiente. Neste caso, os relatórios devem seguir o modelo definido na figura 8. Neste modelo é realizado agrupamento pelo tipo de atividade e por cargo das pessoas responsáveis por executar as atividades. Os valores estimados e realizados são resultantes deste acompanhamento e o desvio é exibido na quarta coluna.

Relatório de Acompanhamento de Projeto			
Tipo de Atividade			
Cargo	Valor Estimado	Valor Realizado	Desvio
Cargo 1	VE	VR	$[(VR/VE)-1/100]$
Cargo 2	VE	VR	$[(VR/VE)-1/100]$
Cargo 3	VE	VR	$[(VR/VE)-1/100]$
	SOMA VE	SOMA VR	$[SOMA(VR/VE)-1/100]$

Figura 8. Layout Relatório de Monitoramento

Por fim, o ambiente também deve emitir relatórios de acompanhamento de processos, monitorando possíveis atividades com problemas. A figura 9 apresenta o layout para estes relatórios.

Devem ser indicados os processos, as atividades com problemas, os agentes responsáveis e a quantidade de vezes que as tarefas foram re-executadas.

<b>Índice de Tarefas Refeitas</b>			
<b>Processo</b>	<b>Atividade</b>	<b>Agente</b>	<b>Nº de Versões</b>
Nome do Processo 1	Nome da Atividade 1	Agente Responsável 1	Qtd de Versões
Nome do Processo 2	Nome da Atividade 2	Agente Responsável 2	Qtd de Versões
Nome do Processo 3	Nome da Atividade 3	Agente Responsável 3	Qtd de Versões
Nome do Processo 4	Nome da Atividade 4	Agente Responsável 4	Qtd de Versões
Nome do Processo 5	Nome da Atividade 5	Agente Responsável 5	Qtd de Versões
Nome do Processo 6	Nome da Atividade 6	Agente Responsável 6	Qtd de Versões
Nome do Processo 7	Nome da Atividade 7	Agente Responsável 7	Qtd de Versões
Nome do Processo 8	Nome da Atividade 8	Agente Responsável 8	Qtd de Versões
Nome do Processo 9	Nome da Atividade 9	Agente Responsável 9	Qtd de Versões

**Figura 9. Layout de Relatório de Acompanhamento de Projetos**

### **3.5 Considerações Finais**

Realizar o Acompanhamento de Projetos sem o apoio de uma ferramenta automatizada freqüentemente é uma tarefa tediosa e passível de erros, além de comprometer grande parte do tempo do gerente de projetos na coleta de informações.

A abordagem proposta neste capítulo propõe a utilização de um ambiente de gerenciamento de processos para modelagem e execução dos processos e a construção de um módulo de acompanhamento de projetos que atenda as necessidades identificadas neste capítulo. O capítulo enumera os requisitos da extensão proposta ao ambiente WebAPSEE, os quais norteiam o desenvolvimento de uma ferramenta de software a qual tem os seus detalhes arquiteturais e de implementação mostrados no próximo capítulo.

# Capítulo 4 – A Ferramenta Monitoring Process

---

Este capítulo apresenta um protótipo para validação da proposta deste trabalho. A aplicação foi desenvolvida e incorporada a um ambiente de engenharia software centrada em processos. A aplicação desenvolvida e o ambiente webAPSEE possibilitam suporte efetivo as atividades de monitoração e controle de projetos.

## 4.1 Introdução

Com o objetivo de apoiar a atividade de monitoramento de projetos, no webAPSEE, foi implementado o módulo **Monitoring Process** que visa atender as necessidades identificadas na seção 3.

**Monitoring Process** é disponibilizado no ambiente de processo de software webAPSEE, e assim utiliza as informações cadastradas no registro de eventos do ambiente. O acesso aos relatórios disponibilizados no ambiente pode ser realizado durante toda a execução de um processo no ambiente e após o término destes.

Este capítulo é organizado como segue. A seção 4.2 descreve o ambiente webAPSEE onde a ferramenta proposta está inserida. A seção 4.3 descreve o modelo de gerência de eventos proposto e implementado neste ambiente. A seção 4.4 descreve a ferramenta **Monitoring Process**. Finalmente, na seção 4.5 são apresentadas as considerações finais.

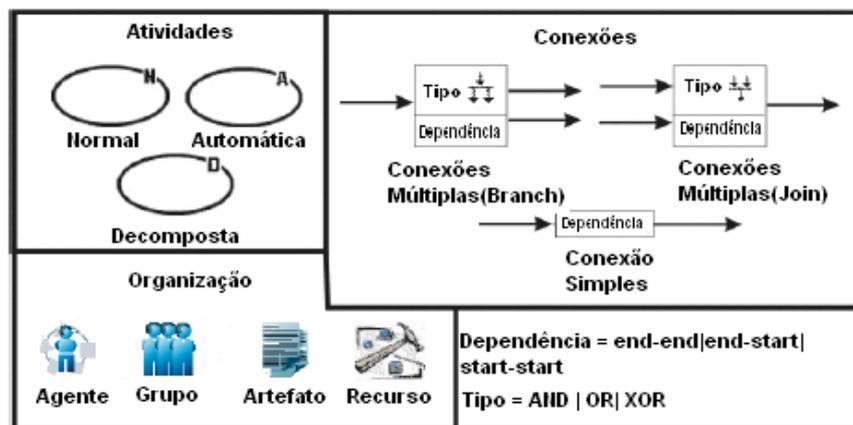
## 4.2 Ambiente webAPSEE

webAPSEE é um ambiente de desenvolvimento de Software Centrado em Processo (PSEE - *Process-Centered Software Engineering Environment*) utilizado para automatizar o gerenciamento de processos de software que evoluiu de um Mecanismo de Processo de Software proposto originalmente para o ambiente PROSOFT [Lima et al 1998]. Atualmente, webAPSEE é um ambiente baseado em Software Livre, servindo como base de integração para um número de meta-modelos para apoiar simulação, instanciação, execução, melhoria e reuso de processos [Lima, 2002][Lima et al., 2002][Reis et al., 2002]. A versão atual da

ferramenta encontra-se gratuitamente disponível em [LABES 2007], assim como sua documentação técnica. Pode-se destacar que o webAPSEE é usado em projetos de desenvolvimento de software hospedados no Centro de Tecnologia da Eletronorte e em projetos conduzidos internamente pelo Laboratório de Engenharia de Software da UFPA.

O meta-modelo do webAPSEE descreve o processo de software como uma coleção parcialmente ordenada de atividades. O ambiente dispõe de uma Linguagem de Modelagem de Processo que fornece representação gráfica para o conjunto de construtores da linguagem proposta. Um conjunto dinâmico de mecanismos de controle é disponível, fornecendo sincronização através de conexões de atividades.

A notação gráfica é resumida pela Figura 10 através das atividades do processo, conexões que interligam as atividades, agentes e grupos de agentes responsáveis pela execução das atividades, artefatos produzidos pelas atividades e recursos necessários para a execução das atividades.



**Figura 10.** Representação gráfica para as principais construções da Linguagem de Modelagem de Processo do webAPSEE [webAPSEE, 2006].

A Figura 11 apresenta duas telas descrevendo as diferentes visões para um gerente e um desenvolvedor de um processo no ambiente webAPSEE. A tela a) exibe a ferramenta usada pelo gerente, mostrando um processo em execução com suas atividades, suas conexões, seus artefatos, e agentes responsáveis. A tela do gerente permite a modelagem do processo e o acompanhamento da sua execução em tempo real. A tela b) exibe para o desenvolvedor sua agenda de tarefas para o processo selecionado, permitindo a realização de ações de controle sobre as tarefas

designadas. Convém observar aqui a terminologia adotada pelo ambiente: quando da descrição do processo como um todo, é usado o conceito de Atividade, enquanto que Tarefa é o termo usado para descrever a situação de uma Atividade para um Desenvolvedor específico.

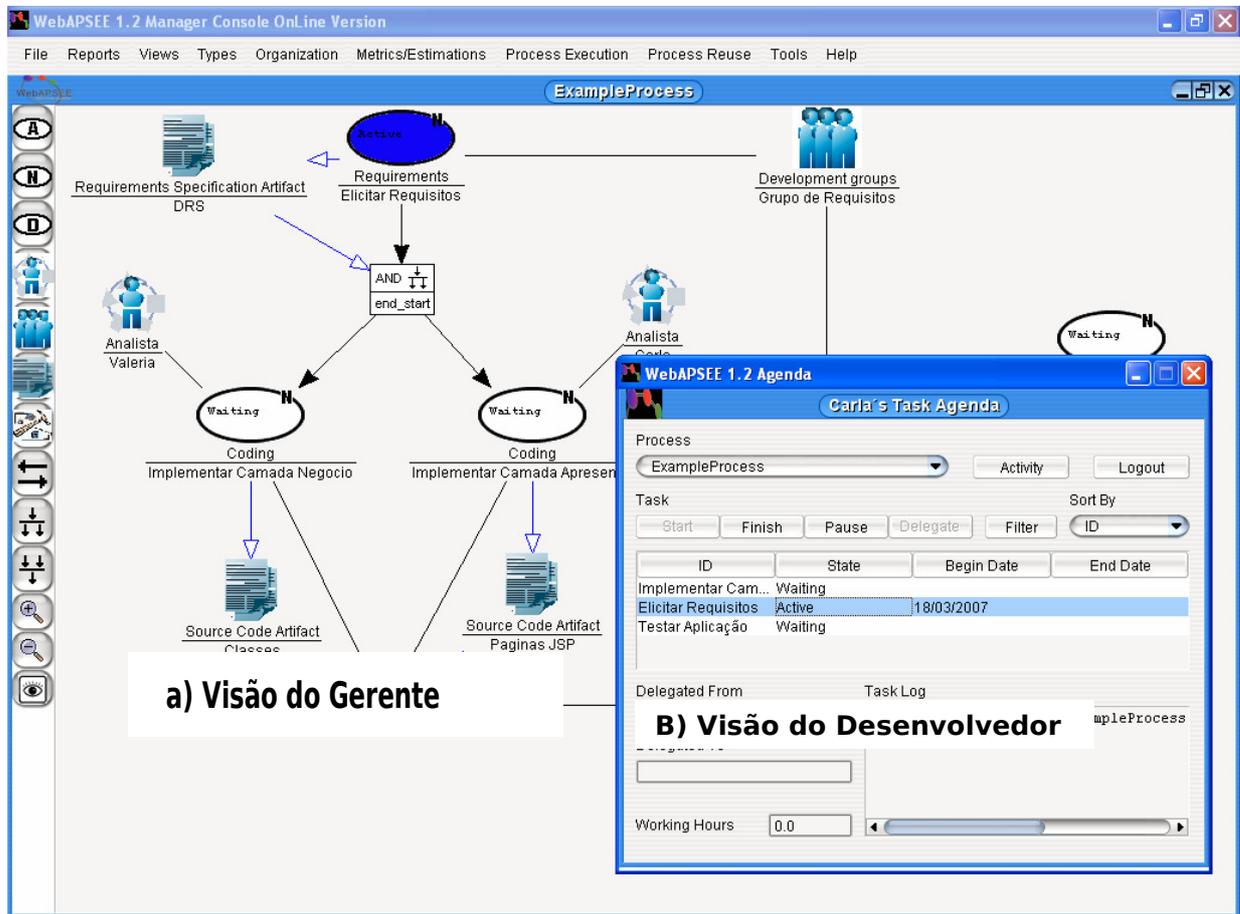


Figura 11. Telas WebAPSEE

### 4.3 Gerência de Eventos no Ambiente webAPSEE

O módulo de gerência de eventos do WebAPSEE é de fundamental importância para a viabilidade da proposta deste trabalho, pois este módulo registra todos os eventos ocorridos nas atividades do processo (início, pausa, término), o que permite realizar um efetivo acompanhamento dos processos. Porém, apesar de haver pouca dúvida acerca dos benefícios de se prover gerenciamento automático de eventos em PSEEs, neste tópico a literatura especializada não apresenta propostas suficientemente detalhadas. Embora algumas soluções na área de gerência de *Workflow* [Van der Aalst, 2004] estejam disponíveis e documentadas na literatura, estas não lidam com as especificidades do processo de software, desconsiderando elementos como as métricas associadas às atividades e aos artefatos produzidos.

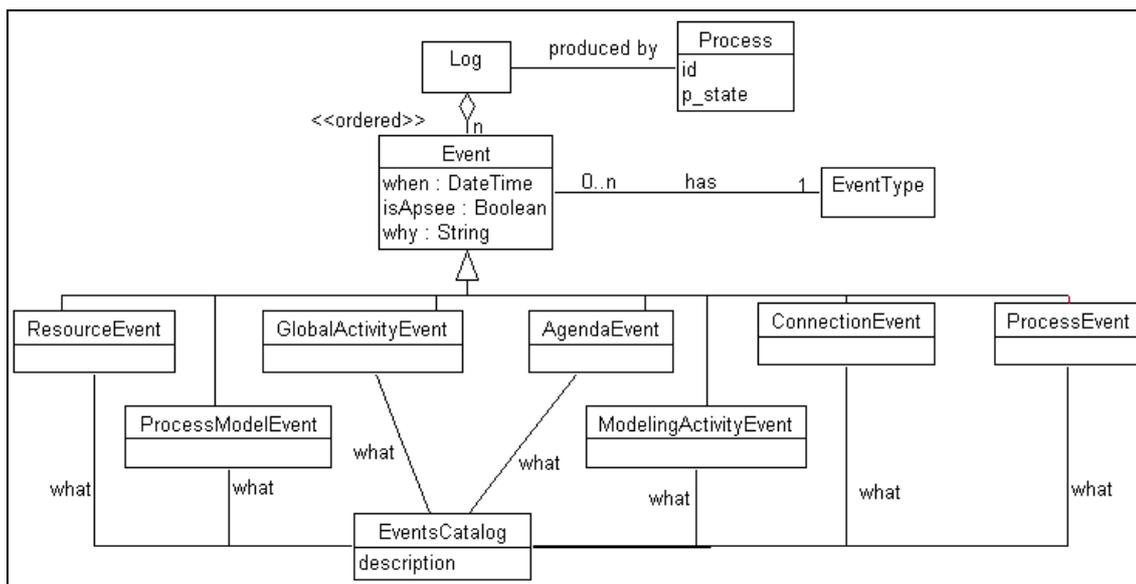
O modelo de gerência de eventos projetado para o ambiente webAPSEE é descrito nesta seção. Tal modelo considera o registro de dois tipos de eventos: 1) os eventos gerados por ações do usuário como, por exemplo, registro do início de execução de uma tarefa ou o *upload* de um artefato, e; 2) o registro de eventos gerados internamente pelo sistema, como por exemplo, as ações resultantes do término de uma tarefa (a qual pode propagar, em função do teste das conexões de atividades envolvidas, automaticamente para o início de outras atividades).

O registro de eventos do webAPSEE foi incorporado posteriormente à disponibilização da primeira versão e é resultado do trabalho descrito por [Paxiúba et al, 2005]. Atenção especial é dada nesta seção para descrever o modelo de dados e o modelo com Gramáticas de Grafos, no qual houve a necessidade de se incorporar as alterações nas funções para que houvesse alimentação automática do registro de eventos.

#### **4.3.1 O modelo de dados do Pacote *Log***

O principal elemento do modelo de gerência de eventos proposto é o pacote *Log*, que consiste de um conjunto de classes e regras para registro de eventos. Esta seção é responsável por apresentar a estrutura de dados principal para armazenar informações acerca de ocorrência de eventos.

Como mostra a Figura 12, o pacote *Log* compreende todos os eventos do webAPSEE. Seu objetivo é armazenar informação sobre o que aconteceu (atributo ***What***), quando aconteceu (atributo ***When***), quem foi o responsável, neste caso pode ser o mecanismo de execução (atributo *isApssee* indica se o responsável é o *Apssee Manager*), ou o agente ou atividade (atributo ***Who*** indica o responsável), e a razão da ocorrência do evento (atributo ***Why***). O valor do atributo ***What*** está relacionado com uma instância específica do catálogo de eventos (classe *EventsCatalog*) ou pode armazenar a identificação da regra que foi responsável pela ocorrência.



**Figura 12. Pacote Log**

Cada processo de software no ambiente WebAPSEE tem um *log* (ou seja, uma instância da classe *Log*), que é composta por eventos (classe *Event*). Eventos estão relacionados aos seguintes componentes do WebAPSEE: recursos, modelos de processos, atividades (visão global, visão do desenvolvedor e eventos de modelagem), conexões entre atividades, e o próprio processo de software.

O campo **Id** contém referências para objetos existentes no meta-modelo do ambiente, e assim é possível navegar através do mesmo para coletar dados adicionais. Por exemplo, é possível coletar informações sobre o tempo planejado para realizar certa atividade e o tempo que realmente foi preciso para realizá-la. Também, é possível identificar que agente e políticas de alocação de recursos foram ativados para um específico modelo de processo ou atividade. Isto é possível pois cada tipo de evento (representados pelas especializações da classe *Event* na figura 5) está associado a outros elementos do ambiente WebAPSEE.

Os tópicos a seguir resumem as funcionalidades para as principais classes do pacote *Log*.

- **Event:** É a classe principal para descrever ocorrências em um *log* de processo. Cada instância de *Event* está relacionada a um tipo de componente do ambiente APSEE (classe *EventType*).
- **ProcessModelEvent:** É a classe responsável por armazenar as mudanças de estado para modelos de processos e atividades decompostas. Cada processo é composto por um modelo de processo,

que pode ser definido de uma maneira recursiva, ou seja, é composto de atividades que podem ser decompostas em novos modelos de processos (chamadas de atividades decompostas).

- **GlobalActivityEvent:** Esta classe registra mudanças do estado global de atividades simples. Ao contrário do tipo *AgendaEvent* (que armazena ocorrências para uma atividade com relação à um determinado agente ou grupo apresentado a seguir), o tipo *GlobalActivityEvent* é usado para armazenar as mudanças de estado relacionadas apenas a ações consolidadas. Este controle é necessário, já que uma atividade pode ser executada por vários agentes ou grupos, e o estado global da execução da atividade pode ser diferente do
- **AgendaEvent:** A maioria dos PSEEs existentes adotam agendas de tarefas como principal tipo interação com desenvolvedores [Bandinelli 1994]. As agendas dos agentes constituem o mecanismo principal de comunicação entre o mecanismo de execução do APSEE e os agentes. No modelo webAPSEE, um agente pode participar de vários processos. Assim, cada agenda representa a visão do agente sobre o processo em execução, ou seja, atividades alocadas a ele e os documentos que pode manipular. Para registro de todas as ocorrências da agenda este modelo propõe a utilização da classe *AgendaEvent*, que pertence ao pacote *Log*. Objetos da classe *AgendaEvent* registram os eventos de atividades simples de acordo com a visão do desenvolvedor (Agente).
- **ConnectionEvent:** Conexões são utilizadas no webAPSEE tanto para estabelecer ligações entre atividades como para descrever controle de processo e fluxo de dados. Os eventos de conexões são registrados na classe *ConnectionEvent* de acordo com as regras do ambiente webAPSEE.
- **ProcessEvent:** O processo é um elemento de primeira ordem no ambiente APSEE, sendo definido como uma composição de todos os tipos mencionados. Todas as mudanças de estado no ambiente webAPSEE são registradas como eventos na classe *ProcessEvent*.

- **ResourceEvent:** Armazena os eventos em relação à gerência de mudança de estados de recursos, possuindo relacionamento com o meta-modelo do webAPSEE através do pacote *Resources* [Paxiúba e Nascimento 2005]. O modelo de recursos do ambiente webAPSEE, descrito no meta-modelo pelo pacote *Resources*, consiste de tipos de recursos (Exclusivos, Compartilhados e Consumíveis), instância de recursos e relacionamentos entre estes. Além de prover um mecanismo para descrever os recursos disponíveis na organização, o modelo de recursos é também utilizado para controlar o acesso concorrente a recursos limitados, aloca-los, verificar suas disponibilidade na organização e verificar a consistência do uso dos mesmos [Lima 2003].
- **EventsCatalog:** É um catálogo com a descrição de todos os eventos que podem ocorrer para componentes de processo no ambiente APSEE. Portanto, esta classe contém todos os possíveis valores do atributo *What*.
- **ModelingActivityEvent:** A permissão para mudança dinâmica de processos é uma importante característica do webAPSEE [Lima et al 2002]. Desse modo, um tipo específico de evento (*ModelingActivityEven*) é associado a mudanças no modelo de processo que influenciem sua execução.

#### 4.3.2 Mecanismo de Gerência de Eventos

A gramática visual da APSEE-PML é responsável por gerar instâncias de modelo de processos e garantir sua consistência quando modificações dinâmicas são necessárias, enquanto regras comportamentais representam a semântica da execução. Mais do que 160 regras foram descritas para permitir a execução de processos; um conjunto separado com mais de 200 regras foi descrito para a gramática de geração (que inclui verificação de consistência) [Lima et al 2002].

A Figura 13 mostra a Regra 1.1 como um exemplo introdutório. Esta regra, que corresponde à chamada da função *ExecuteProcess()* – Função responsável por iniciar um novo processo no ambiente, descreve a reação do sistema ao início de um processo. Neste caso, a pessoa na figura (chamada de *Manager* na figura 6)

requisita a execução do processo (na condição do lado esquerdo da regra). Se todas as condições são satisfeitas (ou seja, o estado do processo é “*not\_started*”), uma nova instância de *ProcessEvent* é gerada (lado direito).

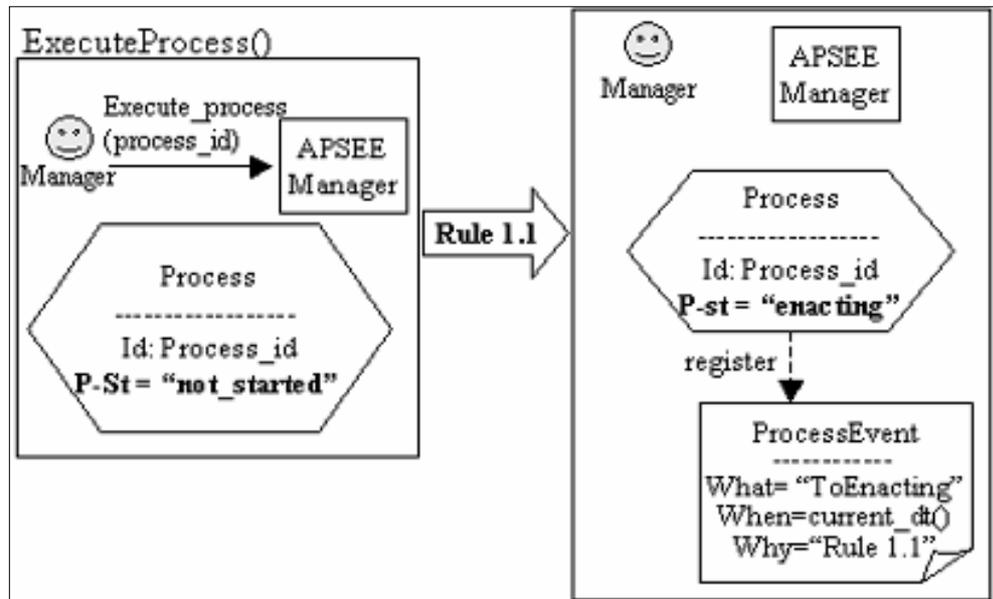
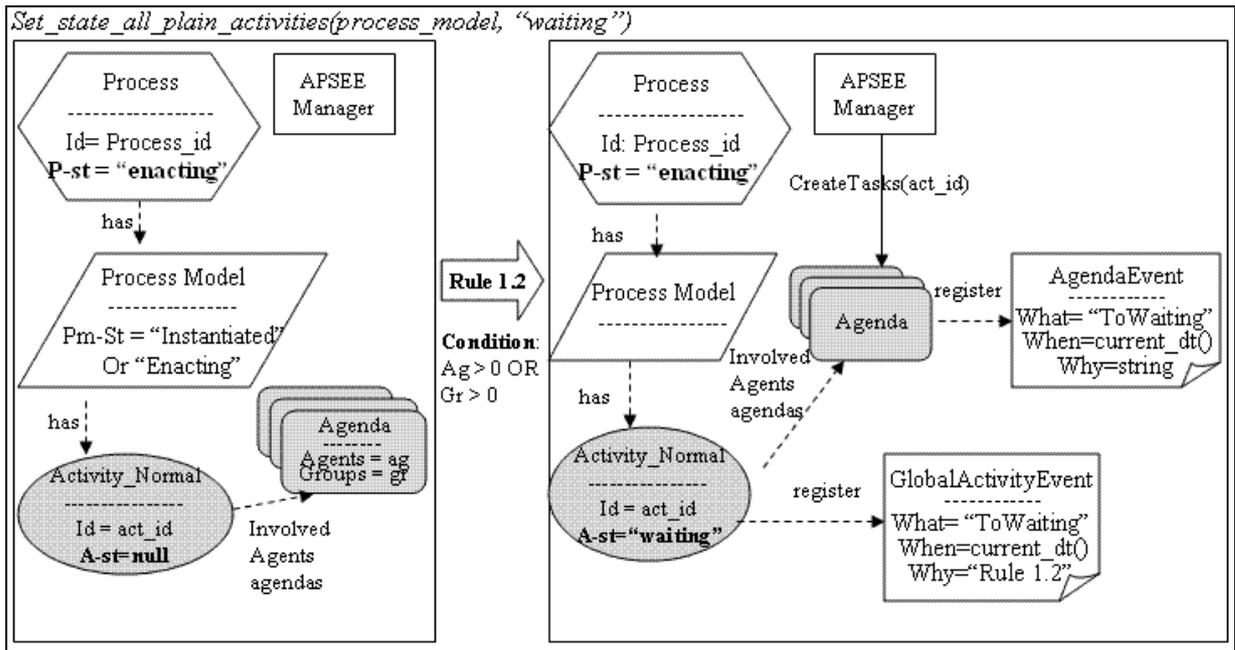


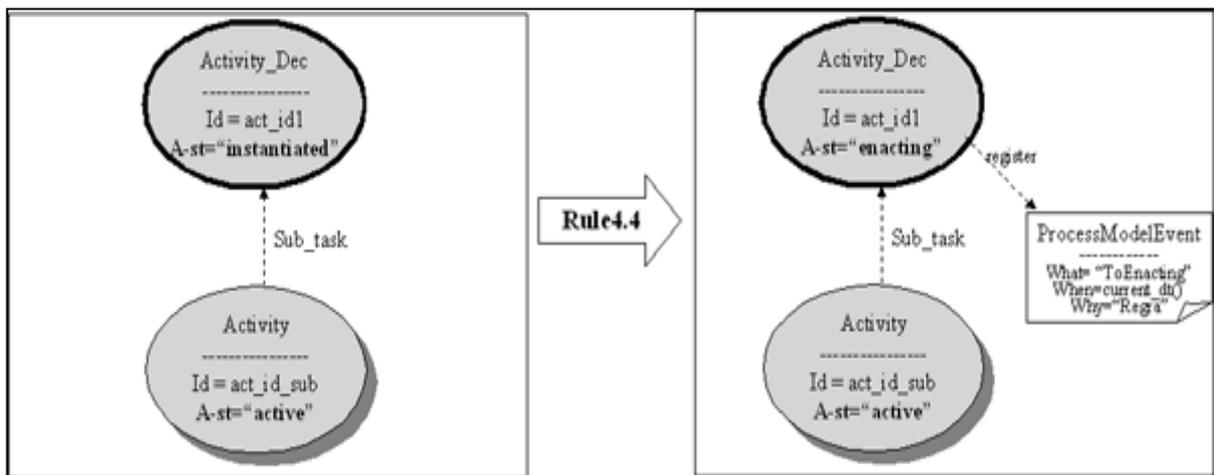
Figura 13. Rule 1.1 – Início da execução do Processo

A Figura 14 apresenta a regra 1.2, que é responsável por sincronizar as agendas dos agentes desenvolvedores com o modelo de processo principal. O modelo de processo instanciado deve possuir, por sua vez, no mínimo uma atividade normal com estado igual a *null* (atividades normais são executadas por humanos, ao contrário das atividades automáticas). O resultado, que é representado no lado direito da regra, altera o estado da atividade para *waiting*, enquanto que o APSEE *Manager* (o *kernel* do sistema) chama a função *CreateTasks* para incluir referências à esta atividade nas agendas correspondentes. A versão atualizada desta regra – apresentada aqui – inclui a criação de instâncias correspondentes de *GlobalActivityEvent* e *AgendaEvent*.



**Figura 14. Rule 1.2 Sincronização de Agenda**

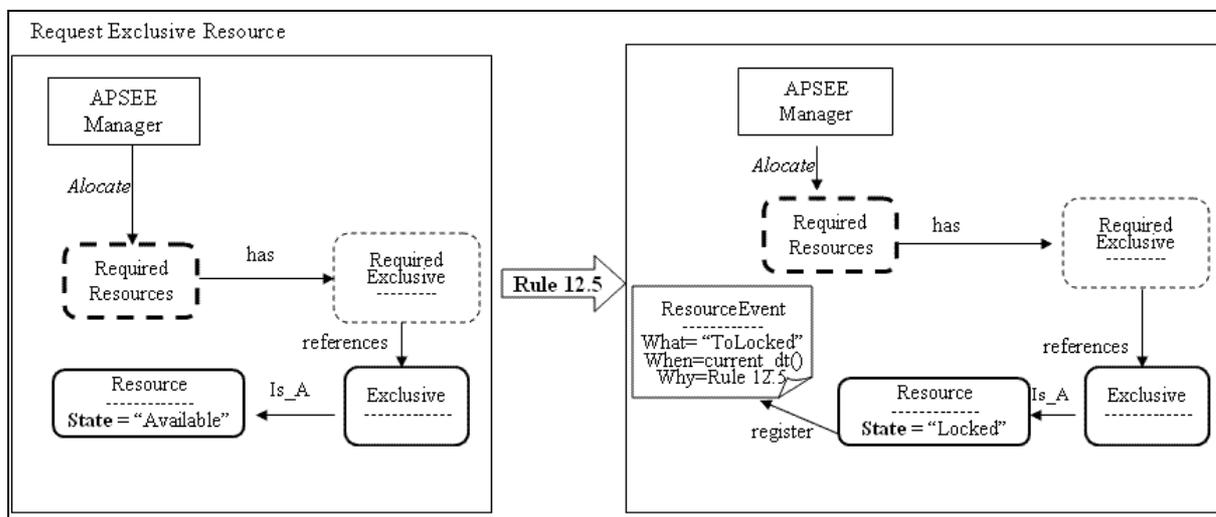
Na Figura 15 tem-se ilustrada a regra 4.4, que é responsável por atualizar o estado do modelo de processos. O lado esquerdo da regra apresenta uma atividade decomposta instanciada que possui uma sub-tarefa ativa. Quando uma atividade decomposta possui pelo menos uma sub-tarefa como ativa seu estado muda de instanciado (*Instantiated*) para em execução (*Enacting*), desta maneira uma nova instância de *ProcessModelEvent* é gerada (lado direito).



**Figura 15. Rule 4.4 Início de Atividade Decomposta**

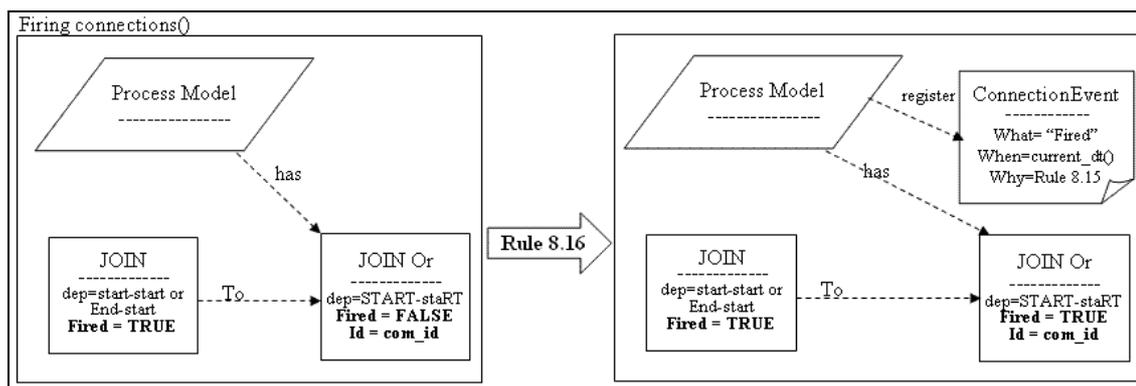
Na Figura 16 vê-se a regra 12.5, que é responsável pela alocação de recursos exclusivos. Neste caso, o *Manager* requisita a alocação de recursos exclusivos requeridos para uma determinada tarefa. Como no conjunto de recursos requeridos tem um recurso exclusivo com estado "*Available*" (lado esquerdo da regra), o estado desse recurso passa a ser "*Locked*" e é feito o registro do evento de

mudança de estado na classe *ResourceEvent* (lado direito). O *Manager* somente encerra a solicitação de alocação dos recursos requeridos se entre eles não houver qualquer recurso disponível, o que é tratado na regra 12.1 do mecanismo de execução demonstrada em [Paxiúba e Nascimento 2005].



**Figura 16. Alocação de Recursos**

A Figura 17 mostra o registro do evento de ativação de uma conexão *Join Or* pertencente a um modelo de processo e que depende de uma conexão *Join* para ser ativada. Como a conexão *Join* está ativada (*Fired* = TRUE no lado esquerdo da regra), a conexão *Join Or* também é ativada, registrando esse evento na classe *ConnectionEvent*.

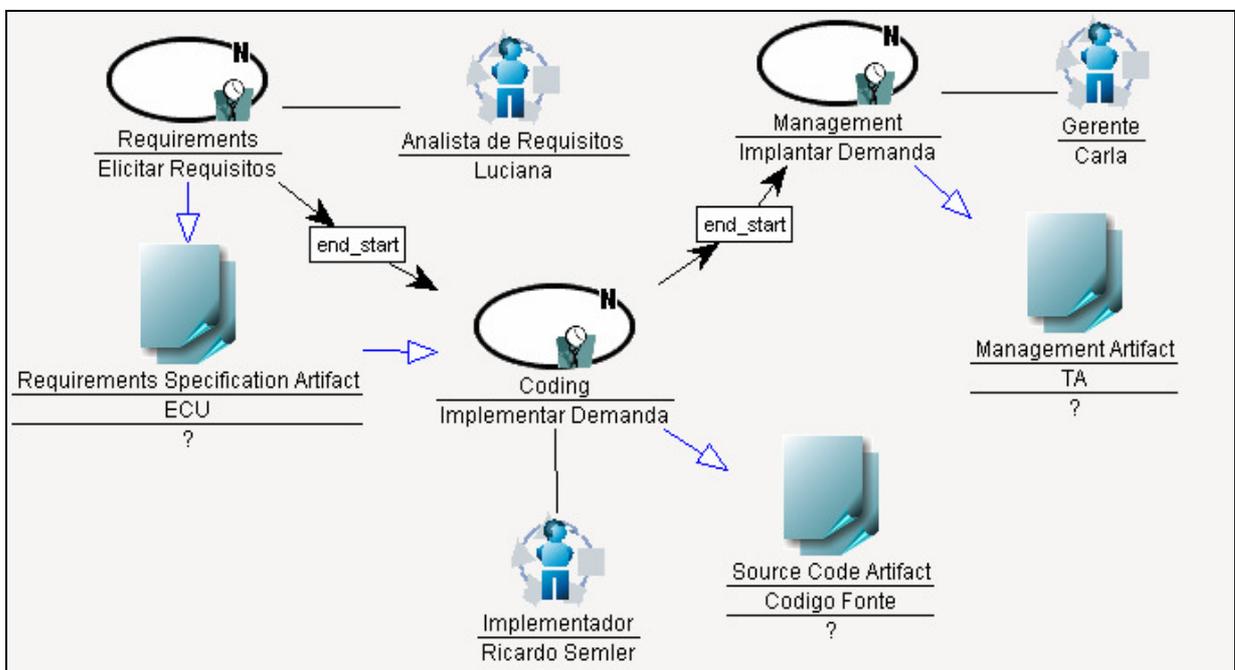


**Figura 17. Ativação de uma conexão**

Para cada regra executada no ambiente um evento é gravado no *log* de execução do ambiente. O *log* armazena, entre outras informações, a identificação do evento (identificado como *oid* na figura 4), o tipo do evento (*EventTypeOid*), o processo que originou o evento (*LogOID*), a data de ocorrência do evento (*\_when*), a regra do ambiente que originou o evento (*Why*) e a sinalização booleana (0 para

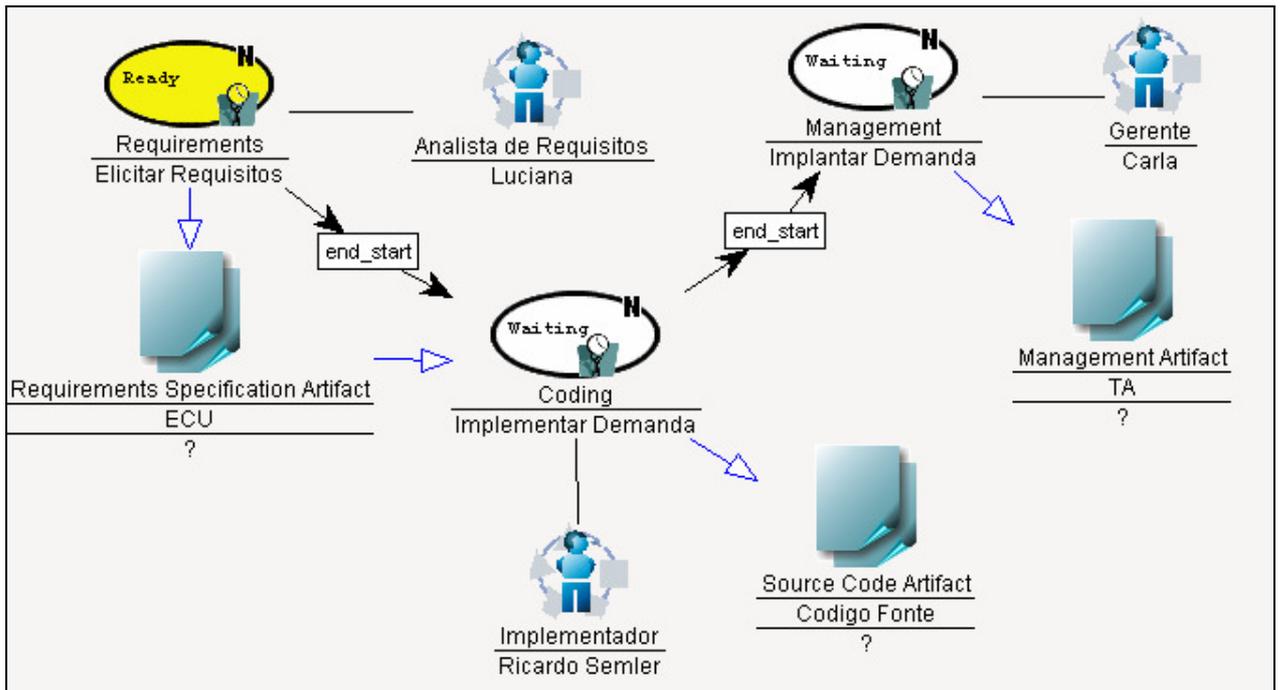
falso e 1 para verdadeiro) se o evento foi originado pelo ambiente (*isCreatedbyApsee*) ou por ação do usuário.

Com o objetivo de detalhar o funcionamento do log será apresentado o acompanhamento da execução de um processo no ambiente e os eventos gerados no log. A Figura 18 apresenta o processo exemplo que contém três atividades principais: Elicitação de Requisitos, Implementação da Demanda e Implantação do Sistema. A segunda atividade de Implementação da Demanda somente pode começar após o término da primeira (conexão *End-Start*), e por sua vez a terceira atividade (Implantar Demanda) somente pode iniciar após o término da segunda (conexão *End-Start*).



**Figura 18. Processo Exemplo Log**

Ao iniciar a execução do processo, as atividades que estiverem prontas para executar são classificadas como *Ready* e as demais como *Waiting*. O início da execução do processo pode ser visualizado na Figura 19. Neste exemplo a atividade Elicitar Requisitos é classificada como *Ready* e as demais assumem o status *Waiting*.



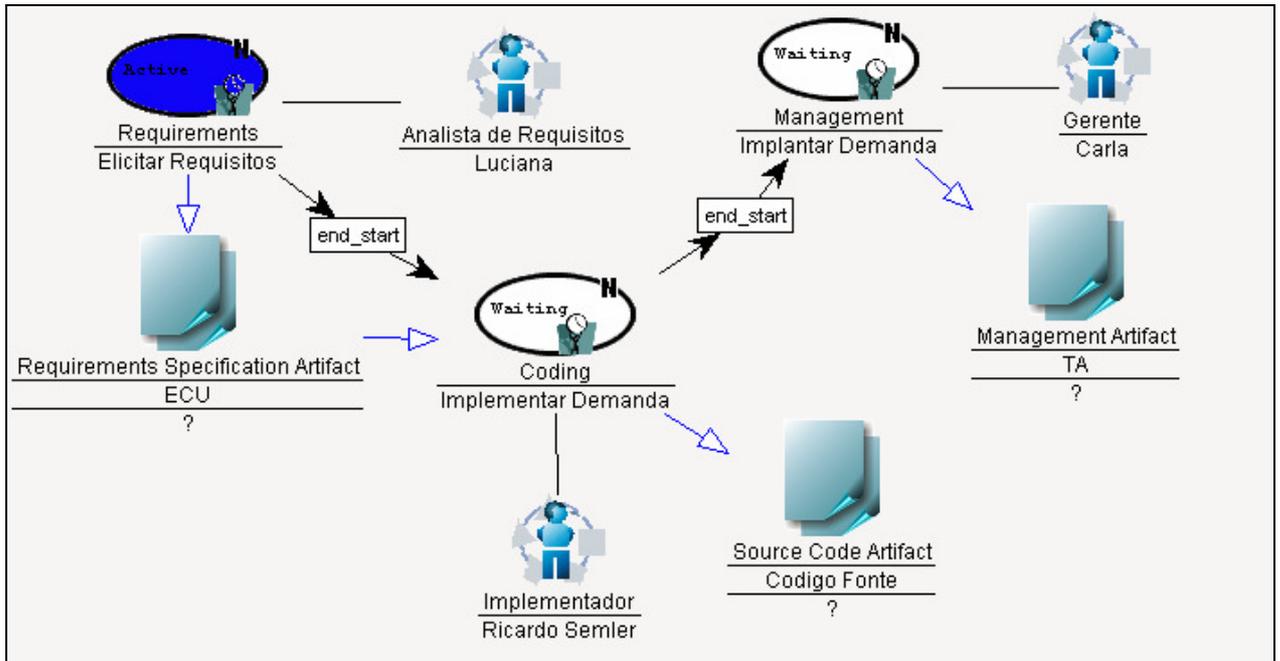
**Figura 19. Execução Processo Exemplo Log**

oid	EventTypeOID	LogOID	_when	isCreate...	why
320	375	10	2007-06-16 00:13:49	1	Rule 1.2
321	376	10	2007-06-16 00:13:49	1	Rule 1.2
322	377	10	2007-06-16 00:13:49	1	Rule 1.1
323	378	10	2007-06-16 00:13:49	1	Rule 1.2
324	379	10	2007-06-16 00:13:49	1	Rule 2.1

**Figura 20. Exemplo com registro de eventos coletados na execução de um processo exemplo**

A Figura 20 apresenta os eventos armazenados no sistema a partir do início da execução do processo Exemplo. Cada Evento está associado a uma regra. A linha 3 da tabela representa o início da execução do processo, pois está associada a regra *Rule 1.1* referente ao início da execução de um processo no ambiente. As linhas 1, 2 e 4 representam a atualização das atividades para estado *Waiting* após início da execução do processo (*Rule 1.2*). A linha 5 representa a atualização da atividade Elicitação de Requisito que está pronta para executar, para o estado *Ready*. (*Rule 2.1*).

A Figura 21 apresenta o início da atividade Elicitar Requisitos.



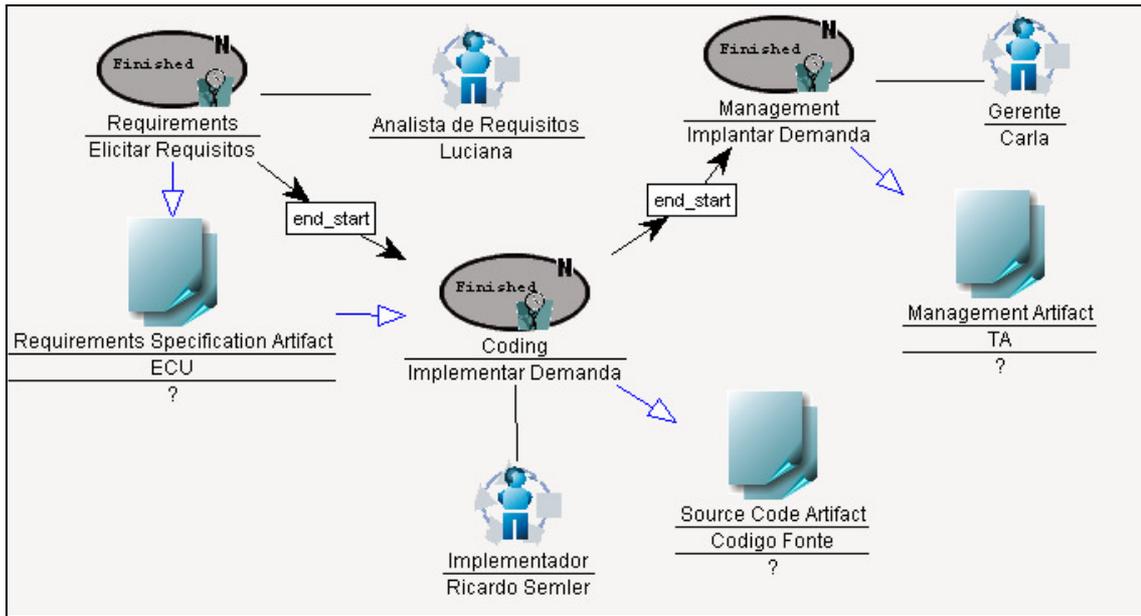
**Figura 21. Início Atividade Processo Exemplo Log**

A Figura 22 apresenta o evento gerado no gerenciador de eventos do ambiente após o início da atividade. Cabe ressaltar que no exemplo o atributo *isCreatedByApsee* possui valor 0, pois o início de uma atividade não é um evento originado pelo ambiente e sim pelo agente responsável pela atividade. A regra associada a este evento é a regra 3.1, referente ao início da tarefa pelo agente e atualização do estado para *Active*.

oid	EventTypeOID	LogOID	_when	isCreatedByAps...	why
327	382	10	2007-06-16 00:32:18	0	Rule 3.1

**Figura 22. Exemplo Log de Eventos – Início de Atividade**

A Figura 23 representa o encerramento do processo. Com a finalização de todas as atividades e término da execução do processo.



**Figura 23. Exemplo Log de Eventos – Término do Processo**

A Figura 24 apresenta os eventos associados ao encerramento da última atividade do processo (*Rule 3.6*) e ao término da execução do processo (*Rule 4.1*)

oid	EventTypeOID	LogOID	_when	isCreatedByAps...	why
338	393	10	2007-06-16 00:47:07	1	Rule 3.6
339	394	10	2007-06-16 00:47:07	1	Rule 4.1

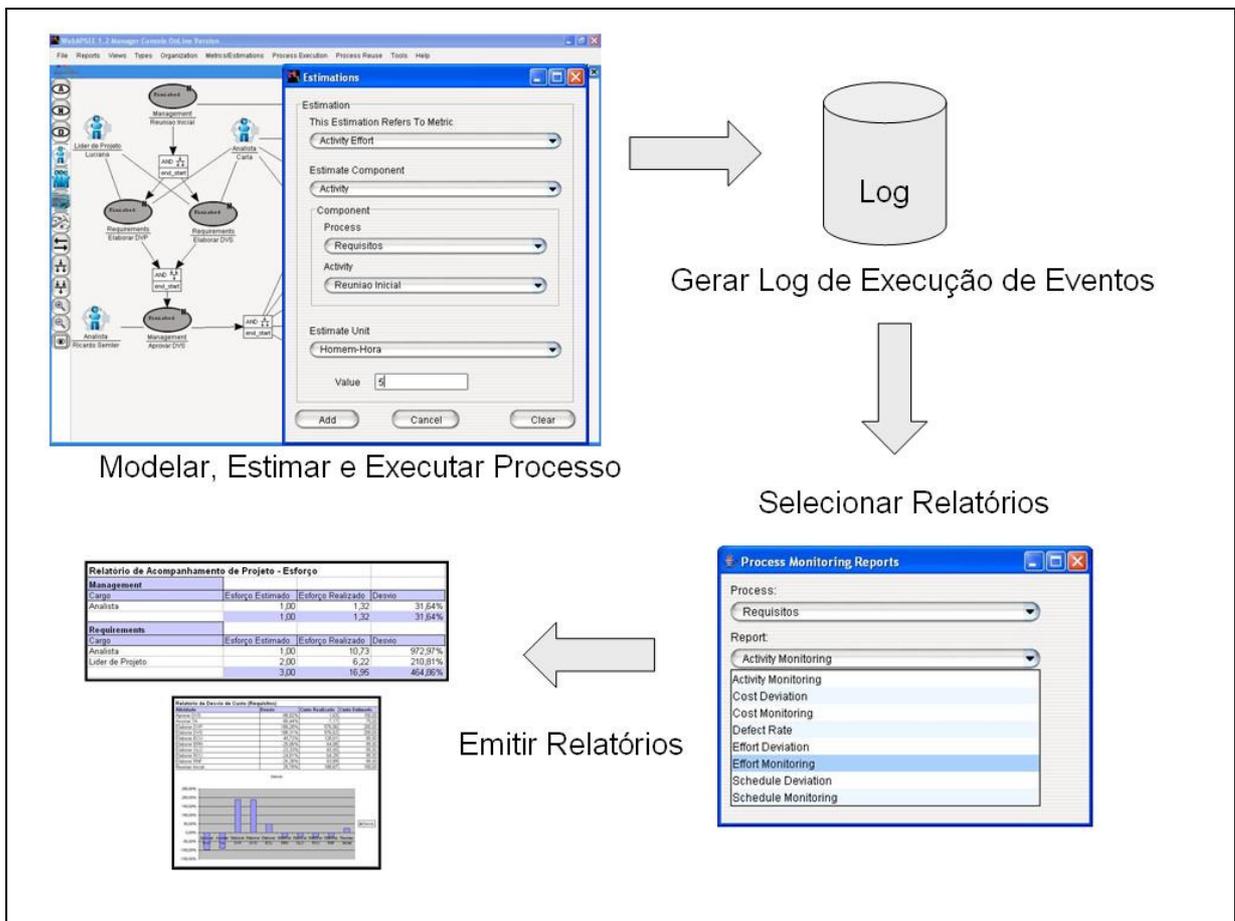
**Figura 24. Exemplo Log de Eventos – Término de Execução**

### 4.3.3 Arquitetura Monitoring Process

Com a finalidade de apoiar o gerente na execução das atividades de acompanhamento dos projetos, este trabalho propõe uma extensão ao mecanismo de gerência de eventos no ambiente WebAPSEE para permitir a emissão de relatórios gerenciais de acompanhamento dos projetos, a ferramenta **Monitoring Process**. A Figura 25 exemplifica o fluxo seguido para a emissão dos relatórios.

Os processos são modelados, estimados e executados no ambiente WebAPSEE. O ambiente WebAPSEE permite que sejam informados estimativas de custo, prazo e esforço para todas as atividades cadastradas no projeto. Estas estimativas podem ser realizadas no planejamento inicial, antes da execução do projeto, e serem alteradas durante a execução do projeto.

Os eventos gerados durante a execução do processo são armazenados no registro de eventos do sistema apresentado na seção 4.1. A partir daí, estas informações são utilizadas para a geração dos relatórios de acompanhamento dos projetos. Estes relatórios são gerados em formato de planilhas eletrônicas com o objetivo de facilitar ao gerente à manipulação das informações apresentadas.



**Figura 25. Fluxo de Geração de Relatórios**

A Figura 26 apresenta a seqüência de atividades necessárias para a emissão de relatórios gerenciais. O primeiro passo consiste em modelar o processo, em seguida realizar as estimativas e executar o processo. Após a execução do processo, pode ser solicitada a emissão de relatórios. Caso todas as condições de emissão de relatórios sejam satisfeitas, estes são gerados, caso contrário o gerente pode completar as estimativas necessárias para a execução dos relatórios e novamente solicitar a emissão dos relatórios.

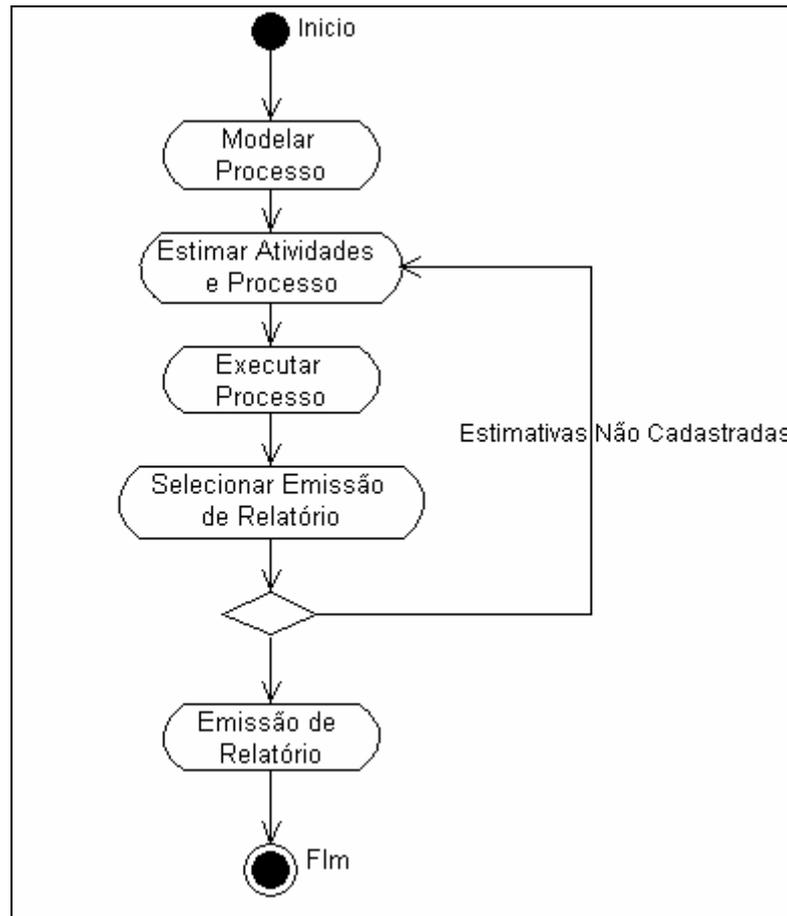


Figura 26. Diagrama de Atividades de Geração de Relatórios

Nesta seção são descritas as modificações arquiteturais necessárias no ambiente para incorporar novas facilidades de geração de relatórios a partir de informações registradas no *Log* de Eventos e no Modelo de dados do ambiente.

Para prover todos os serviços que um PSEE necessita fornecer aos seus usuários, o ambiente webAPSEE utiliza uma abordagem voltada para padrões Web, para interoperabilidade com ferramentas externas através do uso dos *web services* e distribuição de objetos através da linguagem Java, dependendo da configuração mais apropriada para o ambiente do usuário do sistema [webAPSEE, 2006].

Conforme mostrado na Figura 27, a arquitetura do sistema webAPSEE se divide em 3 camadas principais:

A) **Camada Servidora**, que provê serviços de persistência, verificação de consistência para modelagem, controle e armazenamento de artefatos e execução de modelos de processos de software;

B) **Camada Cliente**, que basicamente oferece uma infra-estrutura para acesso aos serviços da camada servidora;

C) **Camada de Ferramentas Clientes**, que possui as ferramentas que interagem diretamente com a interface do usuário para entrada de dados, modelagem de processos, e visualizações de informações obtidas do servidor. Maiores detalhes da arquitetura disponível em [webAPSEE, 2006].

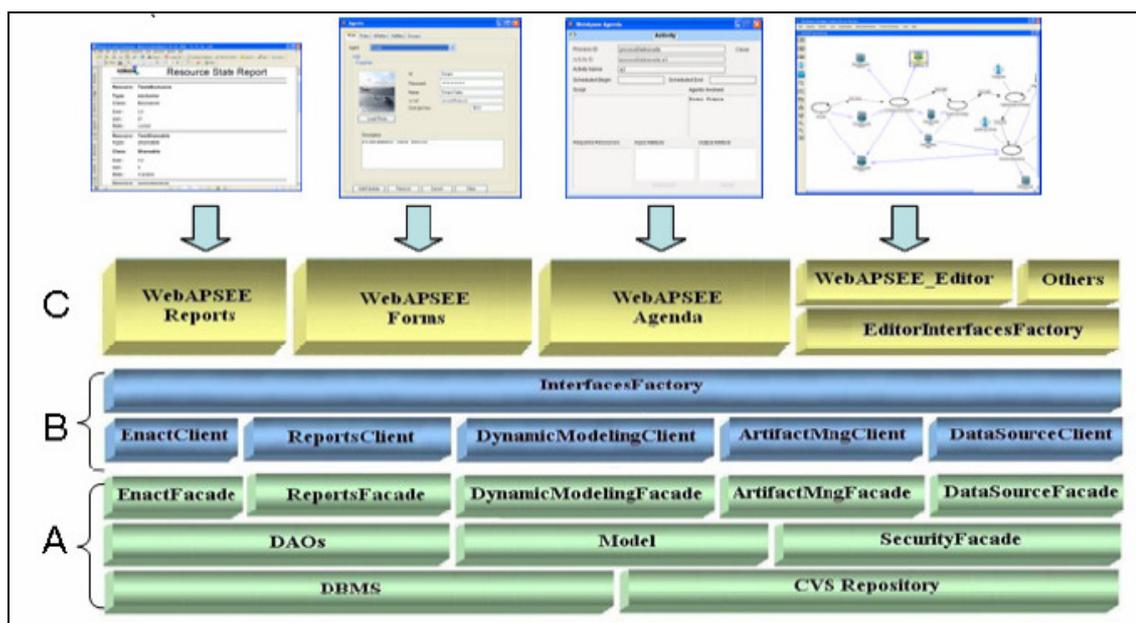


Figura 27. Camadas que compõem a Arquitetura do webAPSEE [Lima et al. 2006]

Para a disponibilização do módulo de geração de relatórios de acompanhamento de projetos no ambiente, foram realizados os seguintes acréscimos na arquitetura da aplicação apresentada na figura 6:

- **Criação de Serviços no Componente *ReportsFacade* (Camada Servidora):** Para implementar a geração de relatórios em formatos de planilhas do Microsoft Excel, foi utilizada a biblioteca jXLS [jXLS]. Embora o *ReportsFacade* implementado anteriormente estivesse apto a gerar arquivos no formato *Portable Document File* (PDF), o formato do Excel foi escolhido para os novos relatórios em função das grandes facilidades de manipulação de dados e geração de gráficos a partir do uso de planilhas eletrônicas. Utilizando a API jXLS são criados arquivos XLS a partir de gabaritos que definem a formatação, fórmulas, macros e a disposição dos dados do relatório usando uma notação específica para tal. Cada serviço novo implementado faz uma invocação a *engine* do jXLS passando como

parâmetros o gabarito e os dados específicos para a geração do relatório e então é obtido como saída um arquivo XLS com os dados exportados. O Apêndice A apresenta maiores informações sobre a API utilizada.

- **Criação de Consultas no Componente DAO (Camada Servidora):** As novas consultas às base de conhecimento necessárias para a disponibilização dos relatórios foram incorporadas no módulo DAO (Camada Servidora) da aplicação.
- **Criação de Classes no Componente ReportsClients (Camada Cliente):** O cliente atual foi estendido, com o acréscimo de novos métodos para realizar as chamadas aos novos relatórios disponibilizados pelo módulo de acompanhamento de projetos.
- **Criação do formulário de acesso a geração dos relatórios** no componente webAPSEE Forms, para que o usuário possa fornecer parâmetros que norteiam a geração dos relatórios.

### 4.3 A Ferramenta Monitoring Process

Como mencionado anteriormente, a ferramenta **Monitoring Process** foi desenvolvida para apoiar a atividade de controlar e monitorar os projetos de uma organização. Para prover este apoio, a ferramenta disponibiliza relatórios que obtêm os valores dos custos, prazos e esforços gastos na realização das atividades e compara com os valores estimados, informados pelo gerente de projetos.

O acesso a **Monitoring Process** é feito através da tela principal do ManagerConsole do webAPSEE. O ManagerConsole exibe a ferramenta usada pelo gerente, mostrando um processo em execução com suas atividades, suas conexões, seus artefatos, e agentes responsáveis. A tela do gerente permite a modelagem do processo e o acompanhamento da sua execução em tempo real.

#### 4.3.1 Relatórios Monitoring Process

Após a modelagem do processo e a execução do mesmo, o gerente pode realizar o acompanhamento do processo através do acesso ao módulo **Monitoring Process** disponível no Manager Console.

A princípio a ferramenta **Monitoring Process** disponibiliza oito relatórios de acompanhamento para apoiar o gerente de projetos na execução desta atividade. Os relatórios serão apresentados a seguir agrupados por objetivo.

O item A apresenta os relatórios definidos para acompanhamento de esforço. O item B os relatórios de acompanhamento de prazo. O item C os relatórios de acompanhamento de custo e finalmente o item D os relatórios de acompanhamento de atividades.

### **A - Acompanhar Esforço**

Para acompanhar o esforço realizado nas atividades a ferramenta disponibiliza dois relatórios de acompanhamento de esforço.

#### **A.1 Effort Monitoring**

Este relatório realiza o acompanhamento de esforço realizado no projeto agrupando por tipo de atividade e cargo. Realiza a comparação entre os valores estimados e os realizados e exibe o desvio.

A Figura 28 apresenta exemplo do Relatório *Effort Monitoring*. O objetivo deste relatório é atender os requisitos de monitoração de esforço e disponibilizar este tipo de informação ao gerente, mostrando uma visão agrupada por tipo de atividade. No ambiente existe uma hierarquia de tipos *default*, que classifica as atividades e utiliza denominação em inglês, por este motivo os tipos de atividade são *Management* e *Requirements*. Esta classificação é realizada pelo gerente ao modelar o processo no ambiente

<b>Relatório de Acompanhamento de Projeto - Esforço (Processo Sincon)</b>			
<b>Coding</b>			
<b>Cargo</b>	<b>Esforço Estimado</b>	<b>Esforço Realizado</b>	<b>Desvio</b>
Implementador	8,00	6,19	-22,67%
	8,00	6,19	-22,67%
<b>Management</b>			
<b>Cargo</b>	<b>Esforço Estimado</b>	<b>Esforço Realizado</b>	<b>Desvio</b>
Analista de Requisitos	1,00	0,27	-73,33%
Gerente	1,00	0,85	-14,78%
Implementador	1,00	0,26	-74,36%
	3,00	1,38	-54,16%
<b>Requirements</b>			
<b>Cargo</b>	<b>Esforço Estimado</b>	<b>Esforço Realizado</b>	<b>Desvio</b>
Analista de Requisitos	4,00	18,23	355,83%
	4,00	18,23	355,83%
<b>Testing</b>			
<b>Cargo</b>	<b>Esforço Estimado</b>	<b>Esforço Realizado</b>	<b>Desvio</b>
Analista de Requisitos	2,00	1,01	-49,69%
	2,00	1,01	-49,69%
Relatório Gerado em 07/06/2007 às 18:44:41			

**Figura 28. Relatório Effort Monitoring**

**Para que este relatório possa ser disponibilizado o gerente deve:**

- Classificar todas as atividades do processo de acordo com o seu tipo.
- Definir o cargo/papel de todos os agentes alocados nos projetos.
- Realizar as estimativas de esforço para todas as atividades.
- Executar o processo.

## **A.2 Effort Deviation**

Este relatório realiza o acompanhamento de esforço realizado no projeto agrupando por atividade e exibindo gráfico de desvios de estimativas por cada atividade do projeto. A Figura 29 apresenta exemplo do relatório *Effort Deviation*.

**Para que este relatório possa ser disponibilizado o gerente deve:**

- Realizar as estimativas de esforço para todas as atividades.
- Executar o processo.

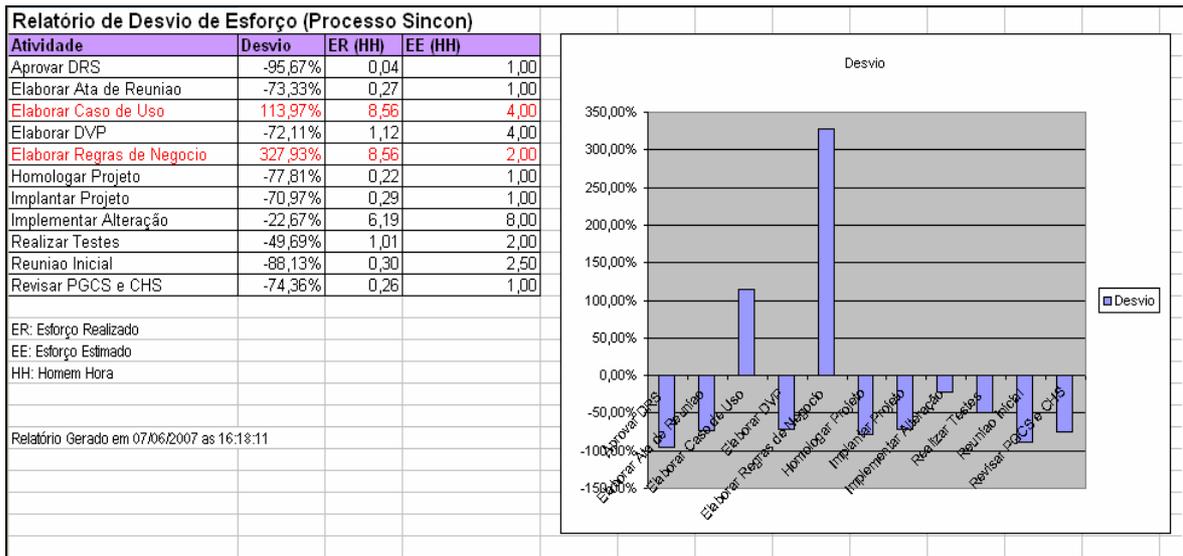


Figura 29. Relatório Effort Deviation

## B – Acompanhar Cronograma

Para acompanhar o prazo das atividades a ferramenta disponibiliza os seguintes relatórios:

### B.1 - Schedule Monitoring

Relatório que realiza o acompanhamento de prazo do projeto indicando o desvio do prazo, o limite de controle - com valor *default* de 10% (este valor pode ser alterado na planilha pelo gerente), e a situação do projeto (apresentando como resultado “Ok” se o desvio menor que limite de controle e “Ação” se o desvio maior que o limite de controle). A Figura 30 apresenta exemplo do relatório *Schedule Monitoring*.

#### Premissas para a geração do relatório:

- A estimativa *Process Schedule*<sup>4</sup> foi realizada.
- O processo está em execução ou finalizado

<sup>4</sup> Métrica definida no ambiente. Esta métrica se refere ao prazo para execução de um processo.

Relatório de Acompanhamento de Projeto - Prazo (Processo Sincon)						
<b>Processo</b>	Processo Sincon					
<b>Data de Início</b>	06/06/2007 22:59:46					
<b>Data de Término</b>	07/06/2007 16:17:54					
<b>Acompanhamento de Prazo</b>	<b>Prazo Estimado</b>	<b>Prazo Realizado</b>	<b>Cumprimento do</b>	<b>Desvio de Prazo</b>	<b>Limite de Controle</b>	<b>Situação</b>
Processo Sincon	24:00:00	17:18:07	72,09%	-27,91%	10%	OK
Relatório Gerado em 07/06/2007 às 18:35:15						

Figura 30. Relatório Schedule Monitoring

## B.2 - Schedule Deviation

Relatório que realiza o acompanhamento de prazo das atividades do projeto, comparando os valores estimados e realizados e indicando os possíveis desvios. A Figura 31 apresenta exemplo do relatório *Schedule Deviation*. Este relatório tem o objetivo de analisar cada atividade modelada no ambiente quanto as suas estimativas e valores realmente executados.

### Premissas para geração do relatório:

- O processo está em execução ou finalizado
- As estimativas de *Activity Schedule*<sup>5</sup> foi informada para todas as atividades.

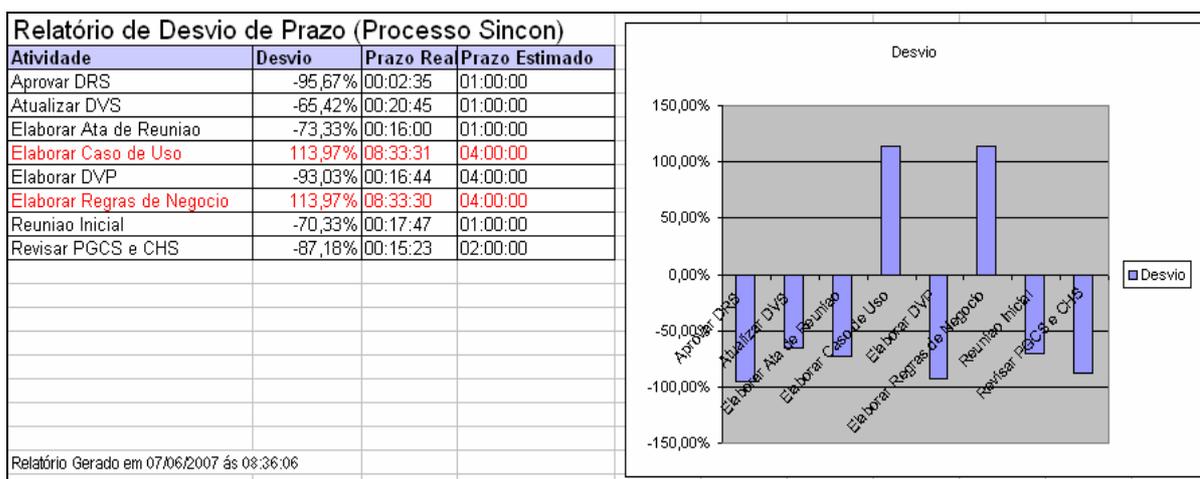


Figura 31. Relatório Schedule Deviation

<sup>5</sup> Métrica definida no ambiente. Esta métrica se refere ao prazo para execução de uma atividade.

## C – Acompanhar Custo

Para prover o acompanhamento de custo o módulo disponibiliza os seguintes relatórios:

### **C.1 - Cost Monitoring**

A Figura 32 apresenta exemplo do relatório *Cost Monitoring*. Este relatório realiza o acompanhamento de custo no projeto agrupando por tipo de atividade e cargo. O objetivo deste relatório é dar uma visão mais geral das atividades com problemas, por isto agrupa as atividades pelos tipos pré-definidos no ambiente (nomenclatura em inglês)

#### **Premissas para Geração do relatório:**

- As atividades do processo estão todas classificadas de acordo com o seu tipo.
- O Papel/Cargo dos Agentes está definido.
- O valor da hora dos Agentes está definido.
- Cadastramento de estimativas de *Activity Effort*<sup>6</sup> para todas as atividades.
- O processo está em execução ou finalizado.

<b>Relatório de Acompanhamento de Projeto - Custo (Processo Sincon)</b>			
<b>Coding</b>			
<b>Cargo</b>	<b>Custo Estimado</b>	<b>Custo Realizado</b>	<b>Desvio</b>
Implementador	800,00	618,67	-22,67%
	800,00	618,67	-22,67%
<b>Management</b>			
<b>Cargo</b>	<b>Custo Estimado</b>	<b>Custo Realizado</b>	<b>Desvio</b>
Analista de Requisitos	150,00	40,00	-73,33%
Gerente	550,00	85,22	-84,51%
Implementador	100,00	25,64	-74,36%
	800,00	150,86	-81,14%
<b>Requirements</b>			
<b>Cargo</b>	<b>Custo Estimado</b>	<b>Custo Realizado</b>	<b>Desvio</b>
Analista de Requisitos	1100,00	1767,51	60,68%
	1100,00	1767,51	60,68%
<b>Testing</b>			
<b>Cargo</b>	<b>Custo Estimado</b>	<b>Custo Realizado</b>	<b>Desvio</b>
Analista de Requisitos	100,00	50,31	-49,69%
	100,00	50,31	-49,69%
Relatório Gerado em 06/06/2007 às 18:14:50			

**Figura 32. Relatório Cost Monitoring**

<sup>6</sup> Métrica definida no ambiente. Esta métrica se refere ao esforço necessário para execução de uma atividade

## C.2 - Cost Deviation

A Figura 33 exibe exemplo do relatório *Cost Deviation* que apresenta o acompanhamento de custo realizado no projeto agrupando por atividade e exibindo gráfico de desvios.

### Premissas para a correta geração do relatório:

- O processo está em execução ou finalizado;
- As estimativas de *Activity Effort*<sup>7</sup> para todas as atividades foram realizadas.

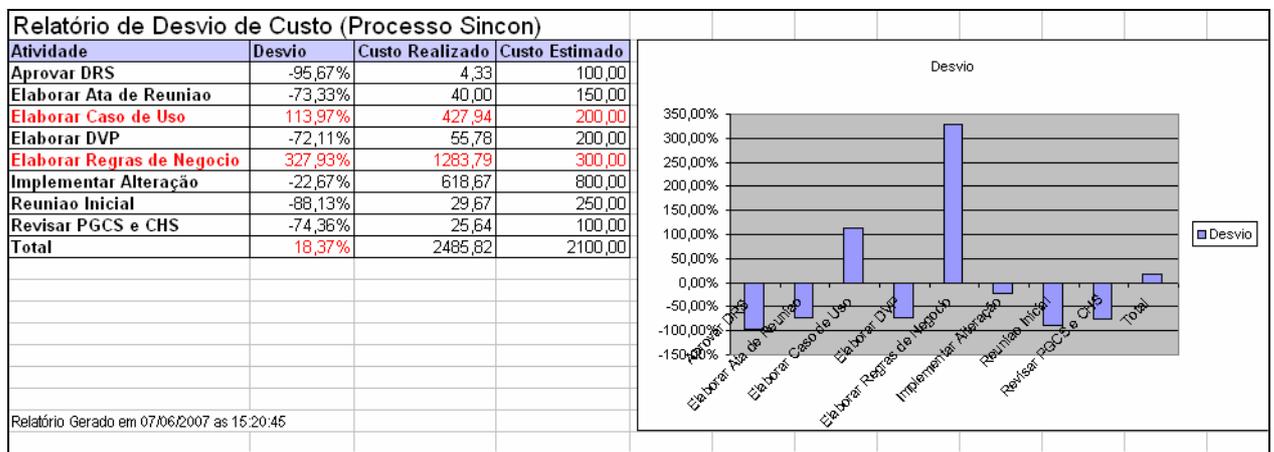


Figura 33. Relatório Cost Deviation

## D - Acompanhar Atividades

Relatórios que realizam o acompanhamento geral das atividades do projeto.

### D.1 Deviation Rate

Este relatório apresenta a quantidade de vezes que uma tarefa é refeita pelos responsáveis pela mesma. O objetivo do relatório é identificar possíveis atividades com problemas e prover visibilidade ao gerente de projetos. A Figura 34 apresenta exemplo deste relatório. A primeira coluna do relatório identifica o Processo, a segunda a atividade, a terceira o agente responsável pela atividade e a quarta coluna quantifica o número de vezes que a atividade foi reexecutada. Também é exibido graficamente o índice de tarefas refeitas por processo.

<sup>7</sup> Métrica definida no ambiente. Esta métrica se refere ao esforço necessário para execução de uma atividade

A seguir a premissa para a correta geração do relatório:

- O processo está em execução ou finalizado

Índice de Tarefas Refeitas			
Processo	Atividade	Agente	Nº de Versões
Requisitos	Elaborar ERN	Luciana	1
Relatório Gerado em 07/06/2007 às 18:51:59			

Figura 34. Relatório Acompanhar Atividade

## D.2 Activity Monitoring

Este relatório apresenta acompanhamento de todas as atividades do projeto indicando a data de início e término estimada e realizada.

Premissas para geração do relatório:

- O processo está em execução ou finalizado
- Para cada atividade foi indicada a data de início e término das atividades.

Acompanhamento de Atividade (Processo Sincon)								
Atividade	Agente	Início Planejado	Início Real	Fim Planejado	Fim Real	Estágio	DI	DF
Reuniao Inicial	Luciana	06/06/2007 - 22:06:32	06/06/2007 - 23:00:43	06/06/2007 - 22:06:32	06/06/2007 - 23:18:48	Finished	00:54:11	01:12:15
Reuniao Inicial	Ricardo Semler	06/06/2007 - 22:06:32	06/06/2007 - 23:01:21	06/06/2007 - 22:06:32	06/06/2007 - 23:18:48	Finished	00:54:48	01:12:15
Reuniao Inicial	Carla	06/06/2007 - 22:06:32	06/06/2007 - 23:00:10	06/06/2007 - 22:06:32	06/06/2007 - 23:18:48	Finished	00:53:38	01:12:15
Reuniao Inicial	Valeria	06/06/2007 - 22:06:32	06/06/2007 - 23:01:09	06/06/2007 - 22:06:32	06/06/2007 - 23:18:48	Finished	00:54:37	01:12:15
Reuniao Inicial	James	06/06/2007 - 22:06:32	06/06/2007 - 23:00:30	06/06/2007 - 22:06:32	06/06/2007 - 23:18:48	Finished	00:53:58	01:12:15
Elaborar Ata de Reuniao	James	06/06/2007 - 22:07:07	06/06/2007 - 23:19:35	06/06/2007 - 22:07:07	06/06/2007 - 23:35:35	Finished	01:12:27	01:28:27
Revisar PGCS e CHS	Valeria	06/06/2007 - 22:22:37	06/06/2007 - 23:19:52	06/06/2007 - 22:22:37	06/06/2007 - 23:35:15	Finished	00:57:15	01:12:37
Implementar Alteração	Ricardo Semler	06/06/2007 - 22:27:39	07/06/2007 - 08:35:06	06/06/2007 - 22:27:39	07/06/2007 - 14:46:07	Finished	10:07:26	16:18:27
Implementar Alteração	Valeria	06/06/2007 - 22:27:39	07/06/2007 - 08:34:39	06/06/2007 - 22:27:39	07/06/2007 - 14:46:07	Finished	10:07:00	16:18:27
Realizar Testes	Luciana	06/06/2007 - 22:28:01	07/06/2007 - 14:46:21	06/06/2007 - 22:28:01	07/06/2007 - 15:46:43	Finished	16:18:19	17:18:41
Homologar Projeto	Carla	06/06/2007 - 22:28:31	07/06/2007 - 15:47:07	06/06/2007 - 22:28:31	07/06/2007 - 16:00:26	Finished	17:18:35	17:31:54
Implantar Projeto	Carla	06/06/2007 - 22:28:58	07/06/2007 - 16:00:29	06/06/2007 - 22:28:58	07/06/2007 - 16:17:54	Finished	17:31:31	17:48:56
DI : Diferença Inicial								
DF: Diferença Final								
Relatório Gerado em 07/06/2007 às 18:58:05								

Figura 35. Relatório Activity Monitoring

### **4.3 Considerações Finais**

Este capítulo apresentou o módulo **Monitoring Process** para acompanhamento e monitoração de projetos.

A aplicação foi desenvolvida para o ambiente de gerenciamento de processos WebAPSEE, utilizando o mecanismo de gerenciamento de eventos do ambiente para geração de relatórios de acompanhamento de projetos em formato de planilhas eletrônicas.

# Capítulo 5 - Avaliação da Proposta

---

Este capítulo apresenta uma análise da proposta desta dissertação, através da simulação da execução de um projeto real e utilização do processo e ferramenta apresentados no capítulo 3 e 4 respectivamente e uma avaliação das características da ferramenta e o atendimento aos requisitos descritos na seção 2, através de estudo comparativo com outras ferramentas existentes;

## 5.1 Simulação

Com o objetivo de simulação, um projeto de pequeno porte, já realizado no Serviço Federal de Processamento de Dados – SERPRO, foi modelado e executado no ambiente webAPSEE. O registro de início e término das atividades foi realizado através da agenda do desenvolvedor e o acompanhamento do projeto foi realizado utilizando a ferramenta proposta neste trabalho. Os resultados obtidos foram semelhantes aos encontrados seguindo o padrão da organização.

A organização citada possui seu próprio processo de desenvolvimento de software que define como os projetos devem ser acompanhados. As atividades destinadas aos gerentes envolvem a produção de Relatórios de Acompanhamentos Mensais, Atualização do Cronograma e Tomada de Ações quando detectados desvios. Para produzir os relatórios os gerentes de projetos devem coletar informações de início e término das tarefas informadas pelos desenvolvedores nos sistemas de informações gerenciais da organização, atualizar manualmente os relatórios, verificar as atividades com problemas e tomar ações corretivas.

### 5.1.1 – O processo

O processo exemplo trata-se de uma manutenção em um sistema já existente. Cada participante do projeto possui um papel específico e pode participar de uma ou mais atividades. A tabela realiza uma breve descrição das atividades a serem realizadas no projeto.

Tabela 4. Seqüência de Atividades Processo Exemplo

1 – Reunião Inicial com todos os envolvidos no projeto.
2 – Um dos participantes da reunião é alocado para elaborar a Ata de Reunião
3 – Um segundo participante é responsável pela atualização dos artefatos de gestão de configuração (PGCS e CHS)
4 – Os analistas de requisitos são responsáveis pela elicitação que compreende uma seqüência de atividades: Elaboração do Documento de Visão do Projeto, Revisão de Documento de Visão do Sistema, Elaboração dos Casos de Uso e Regras de Negócio.
5 – O gerente em conjunto com o cliente aprova os requisitos.
6 – A implementação é realizada após a aprovação dos requisitos.
7 – Com o término da implementação é iniciada a fase de testes.
8 – Após a fase de testes o projeto é homologado e implantado.

Além das atividades descritas o projeto deve ser constantemente monitorado pelo gerente de projetos.

A Figura 36 apresenta a seqüência de atividades do processo exemplo, definido na tabela 4.

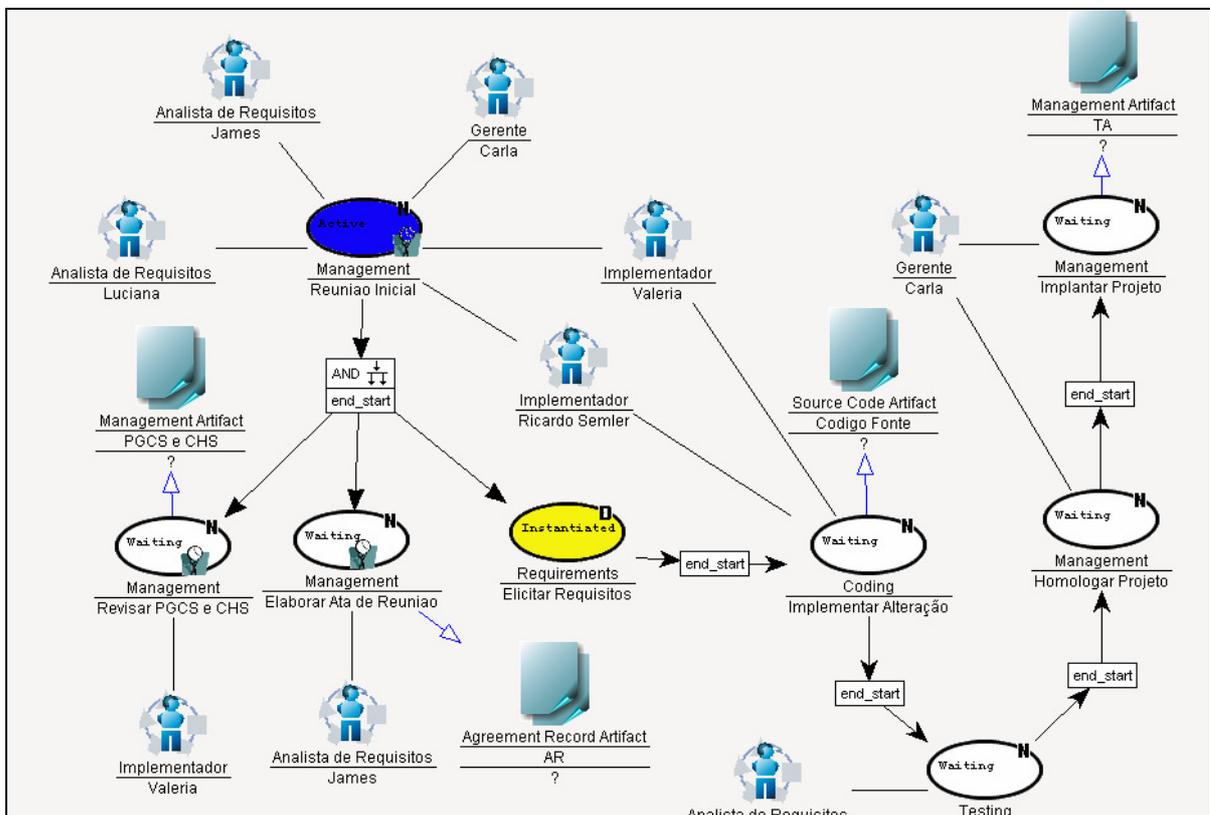
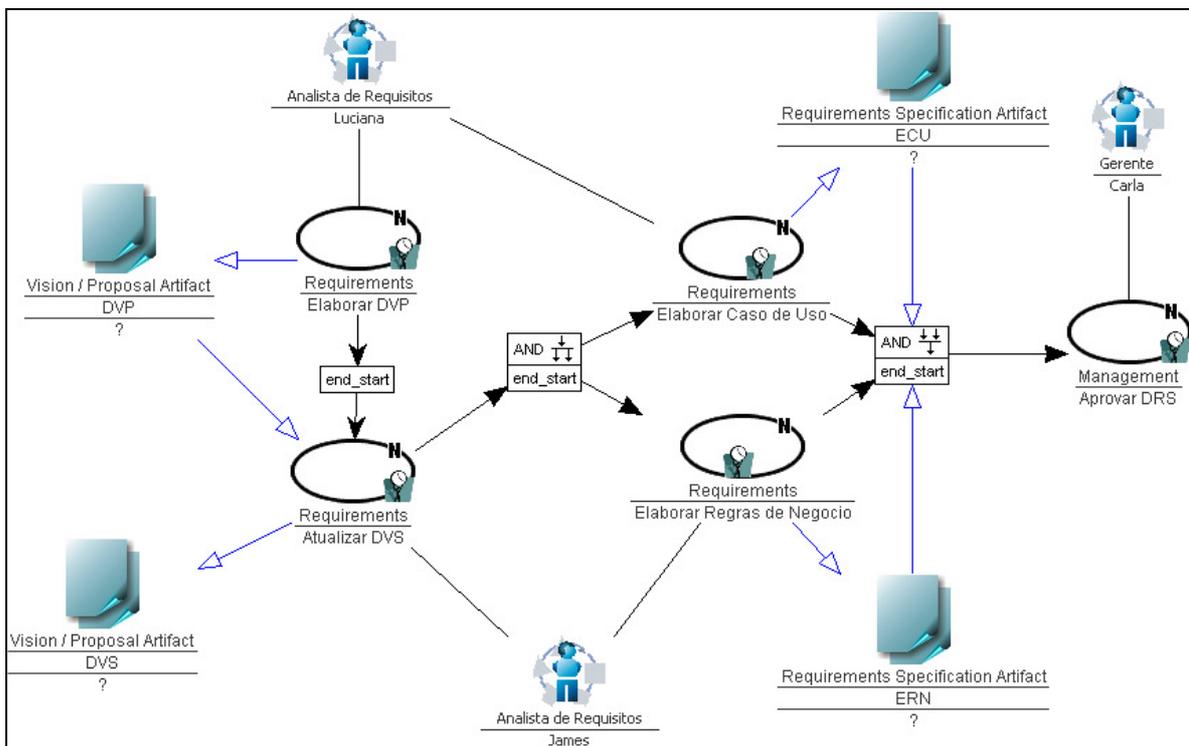


Figura 36. Processo Exemplo Sincrono

Para fins de modelagem do processo utilizou-se o conceito de atividade decomposta definida no ambiente webAPSEE que permite que uma atividade possa ser composta por uma série de atividades menores. Este conceito foi utilizado para a atividade de Elicitar Requisitos.

A Figura 37 apresenta o fluxo da atividade de Elicitação de Requisitos que é composta pelas atividades de Elaboração do Documento de Visão do Projeto, Revisão de Documento de Visão do Sistema, Elaboração dos Casos de Uso e Regras de Negócio e Aprovação dos Documentos de Requisitos do Sistema.



**Figura 37. Atividade Decomposta Elicitação de Requisitos**

O processo exemplo foi modelado na ferramenta e seu acompanhamento foi realizado através do módulo proposto neste trabalho. Os resultados obtidos seguem abaixo enumerados:

- **Possibilidade de Geração de Relatórios a qualquer momento da execução do projeto.** A seguir exemplo de relatório gerado para acompanhamento das atividades dos projetos. O relatório apresentado na Figura 38 foi obtido durante a execução do processo e exibe somente as atividades executadas até o momento de sua geração. Pode-se notar que duas atividades são identificadas como com problemas, desvios de estimativas positivos, ou seja, valor de realização maior que o valor estimado. A vantagem em relação ao processo utilizado na organização é a

possibilidade do gerente obter automaticamente os relatórios de acompanhamento a qualquer momento da execução do projeto.

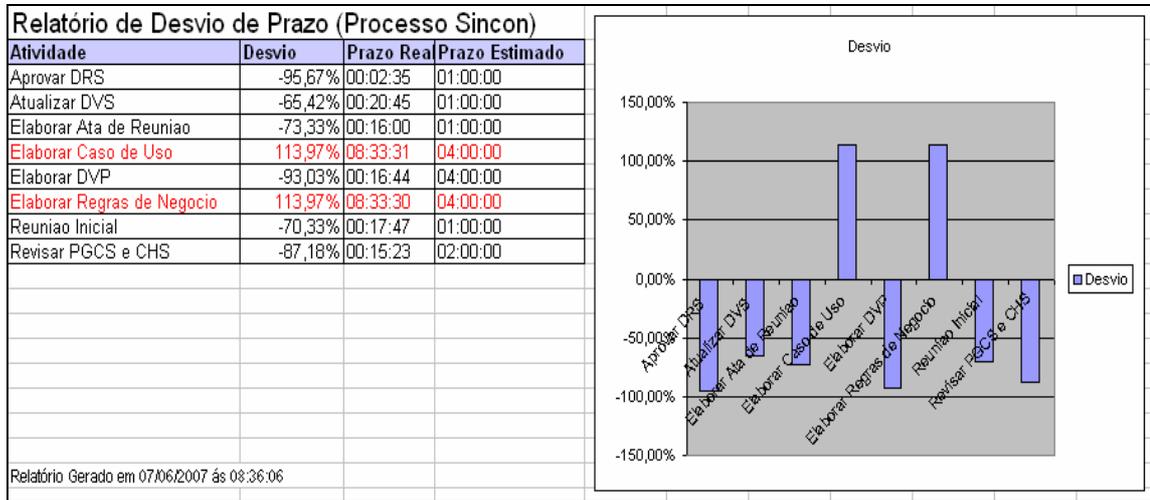


Figura 38. Relatório Schedule Deviation do Processo Exemplo

- Possibilidade de obter automaticamente durante a execução do projeto  **sinalização de problemas**. Isto é exemplificado pela Figura 39, que ilustra uma situação onde o projeto está excedendo o custo previsto até o momento (Desvio positivo de 18,37%).

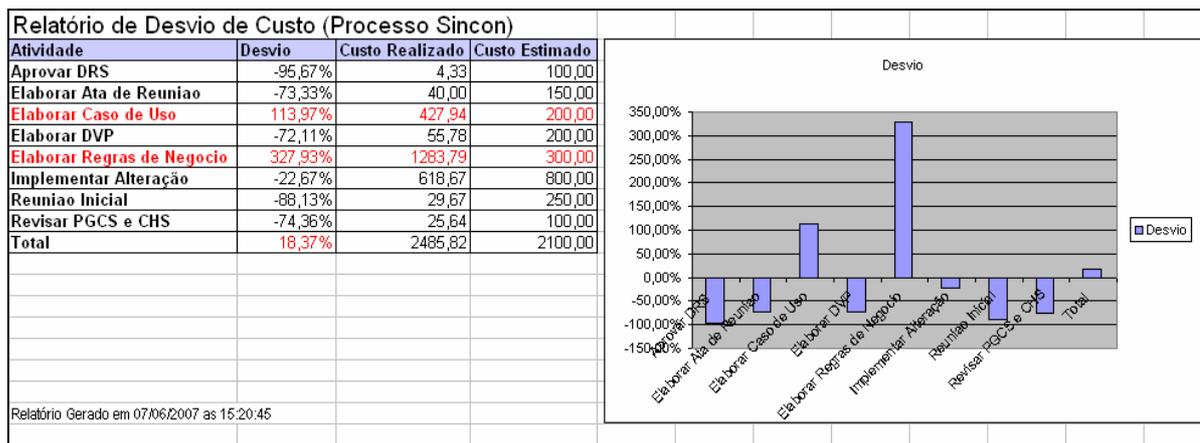


Figura 39. Relatório Cost Deviation do Processo Exemplo

- Possibilidade de utilização dos relatórios para realizar estimativas para  **projetos futuros semelhantes ao executado**. Neste caso de exemplo, projetos de manutenção corretivas em um mesmo sistema costumam seguir os procedimentos listados, e geralmente existe uma equipe fixa para um sistema, logo os relatórios

podem ser utilizados para ao início de um novo projeto verificar o custo, prazo, esforço realizado em projetos semelhantes e apoiar na elaboração de estimativas. O relatório exibido na Figura 40 é um exemplo de relatório que pode ser utilizado na elaboração das estimativas de esforço para um projeto semelhante.

<b>Relatório de Desvio de Esforço (Processo Sincon)</b>			
<b>Atividade</b>	<b>Desvio</b>	<b>Esforço Realizado (HH)</b>	<b>Eforço Estimado (HH)</b>
Aprovar DRS	-95,67%	0,04	1,00
Elaborar Ata de Reuniao	-73,33%	0,27	1,00
<b>Elaborar Caso de Uso</b>	<b>113,97%</b>	<b>8,56</b>	<b>4,00</b>
Elaborar DVP	-72,11%	1,12	4,00
<b>Elaborar Regras de Negocio</b>	<b>327,93%</b>	<b>8,56</b>	<b>2,00</b>
Homologar Projeto	-77,81%	0,22	1,00
Implantar Projeto	-70,97%	0,29	1,00
Implementar Alteração	-22,67%	6,19	8,00
Realizar Testes	-49,69%	1,01	2,00
Reuniao Inicial	-88,13%	0,30	2,50
Revisar PGCS e CHS	-74,36%	0,26	1,00

**Figura 40. RelatórioEffort Deviation do Processo Exemplo**

## 5.2 Análise de Características

A avaliação realizada para verificar o suporte fornecido ao acompanhamento de projetos é denominada *feature analysis* (Kitchenham, 1996). Nesse tipo de avaliação identifica-se um conjunto de requisitos necessários para que usuários realizem determinada atividade. Essa avaliação pode ser realizada por uma única pessoa que, após identificar os requisitos, mapeia os mesmos para um conjunto de *features* e então verifica se estas estão presentes nas ferramentas sendo avaliados.

### 5.2.1 Requisitos para Acompanhamento e Monitoração de Projetos

Para realizar a avaliação, primeiramente é necessário identificar os critérios. Para isso, foi definido um conjunto de requisitos desejáveis em ferramentas de gerenciamento de acompanhamento de projetos. Esse levantamento foi feito através dos requisitos definidos nos principais modelos de qualidade de software e gerenciamento de projetos. Dessa forma, os requisitos necessários para um apoio efetivo a monitoração de projetos definidos neste trabalho foram mapeados para um conjunto de características abaixo relacionadas. Em seguida estas características são utilizadas para efeitos comparativos com algumas ferramentas de gerenciamento e acompanhamento de projetos existentes.

Tabela 5. Características da Ferramenta

<b>Características</b>	<b>Detalhamento</b>
Monitorar as Atividades do Projeto	A ferramenta deve possibilitar acompanhar o status de todas as atividades do projeto, acompanhando desvios de datas de início e fim.
Monitorar os Custos do Projeto	A ferramenta deve emitir relatório de acompanhamento de custos sinalizando as atividades que apresentem desvios significativos.
Monitorar os Prazos do Projeto	A ferramenta deve emitir relatório de acompanhamento dos prazos no projeto, sinalizando as atividades com problema --> Desvios Significativos.
Monitorar os Esforços no Projeto	A ferramenta deve emitir relatório de acompanhamento do esforço gasto no projeto, sinalizando as atividades que apresentem desvios significativos.
Identificar Possíveis Desvios do Projeto	A ferramenta deve calcular os desvios de estimativa de prazo, esforço e custo para todas as atividades do processo.
Indicar Atividades com Problemas	As atividades que apresentem desvios significativos devem ser sinalizadas para o usuário,
Permitir o Acompanhamento de todos os projetos da Organização	Os relatórios devem ser emitidos para todos os projetos em execução e finalizados no ambiente de gerenciamento de processos.
Permitir Fácil Manipulação dos Relatórios de Acompanhamento	Os relatórios devem ser emitidos em planilhas eletrônicas que facilitem a manipulação dos dados pelo gerente.
Apresentar resultados que facilitem a análise dos desvios	Os desvios devem ser calculados automaticamente e apresentados em modo visual e/ou gráfico em um modo que facilite a análise dos mesmos.

### 5.2.2 Ferramentas Analisadas

Segundo o Project Management Software (em [Project Management Software, 2007]) há atualmente há mais de 180 ferramentas de gerenciamento de projetos disponíveis no mercado. A maioria delas apresenta funcionalidades semelhantes. Entre as ferramentas listadas, destacam-se seis como as que mais fornecem funcionalidades ao gerente de projetos.

A seguir será apresentada breve descrição das seis ferramentas que serão utilizadas com fins comparativos com a abordagem apresentada neste trabalho - **Monitoring Process** , cuja apresentação é detalhada no capítulo 4

- **Microsoft Project Server 2003 -**  
**<http://www.microsoft.com/brasil/office/project>**

O Microsoft Project é uma das ferramentas mais utilizadas para gerenciamento de projetos nas empresas. O Microsoft Project é um software *desktop* que oferece recursos tanto para o processo de seleção e priorização de projetos quanto ao processo de controle dos mesmos. A versão Server do Microsoft Project permite um repositório central dos projetos, acessível aos membros dos projetos e com possibilidade de relatórios web. Os membros da equipe atualizam as informações do projeto, colaboram e permanecem informados através de correio eletrônico e ferramentas baseadas na Web.

- **KM Project - <http://www.kmproject.com>**

Software para gerenciamento de projeto disponível em versão web com ênfase no processo de seleção e priorização de projetos, mas que também provê suporte ao controle de riscos, e cronograma dos projetos. Possui uma base de conhecimento central compartilhada por todos os membros do projeto.

- **Copper 2004 - <http://www.copperproject.com>**

Copper é uma ferramenta de colaboração e gerenciamento disponível em versão web com ênfase no processo de controle do projeto. Provê lembretes automáticos de tarefas pendentes para os membros de projeto.

- **ACE Project - <http://www.aceproject.com>**

Software de gerenciamento de projeto desenvolvido em plataforma web. Permite o gerenciamento de projetos dentro de uma organização, acompanhamento do cronograma do projeto através de gráficos de Gantt, notificação de tarefas pendentes via e-mail.

- **RIQTek Manager- <http://www.riqtek.com/product.htm>**

Ferramenta de gerenciamento de projetos desenvolvida em ambiente web, integra um módulo CRM - *Customer Relationship Management* - para melhorar a colaboração entre os times de engenharia e negócios, além de gerar notificações que podem ser enviadas via e-mail e lembretes automáticos de tarefas pendentes que requerem ações imediatas.

- **DotProject-** <http://www.dotproject.net>

O SotProject é um *framework* de gerenciamento de projetos desenvolvido em ambiente web. Inclui módulos para acompanhamento de projetos e tarefas, disponibilizando gráficos de Gantt para acompanhamento.

### 5.2.3 Quadro Comparativo

O quadro abaixo ilustra as características presentes e ausentes nas ferramentas analisadas. O símbolo ✓ indica a presença da característica e o símbolo ✗ indica a ausência da mesma. O juízo acerca da presença ou ausência de cada característica decorre da análise da documentação eletrônica disponível para cada produto no momento da redação deste texto.

Tabela 6. Comparação de Ferramentas

<b>Ferramentas</b>	Monitorar as Atividades do Projeto	Monitorar os Custos do Projeto	Monitorar os Prazos do Projeto	Monitorar os Esforços no Projeto	Identificar Possíveis Desvios do Projeto	Indicar Atividades que apresentem desvios significativos	Permitir o Acompanhamento de todos os projetos da Organização	Permitir Fácil Manipulação dos Relatórios de Acompanhamento	Apresentar resultados que facilitem a análise dos desvios
<b>Microsoft Project</b>	✓	✗	✓	✓	✓	✗	✓	✗	✗
<b>KM Project</b>	✓	✗	✓	✓	✓	✗	✓	✗	✗
<b>Cooper 2004</b>	✓	✗	✓	✓	✓	✗	✓	✗	✗
<b>ACE Project</b>	✓	✗	✓	✓	✓	✗	✓	✗	✗
<b>RIQTek Manager</b>	✓	✗	✓	✓	✓	✗	✓	✗	✗
<b>DotProject</b>	✓	✗	✓	✓	✓	✗	✓	✗	✗
<b>Monitoring Process</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓

#### 5.2.4 Requisitos de Qualidade

Neste item serão analisados alguns atributos de qualidade essenciais e verificados como estes são atendidos na proposta apresentada neste trabalho. A tabela 7 apresenta os atributos que serão analisados e os conceitos envolvidos.

Tabela 7. Requisitos de Qualidade [RUP]

<b>Atributos</b>	<b>Conceito</b>
Segurança	Envolve todos os requisitos que definem a política de segurança adotada para a aplicação, que envolvem desde o controle de acesso da aplicação a confiabilidade dos dados armazenados na mesma.
Performance	Envolve considerações sobre tempo de resposta, velocidade de processamento, eficiência, consumo de recursos e volume de produção.
Usabilidade	Envolve todos os requisitos que melhor definem as facilidades de uso do software, o nível de consistência dos dados apresentados, e de documentação.
Suportabilidade	Engloba os requisitos que melhor definem a capacidade do sistema de suportar mudanças, evoluções e reparos. Definem a testabilidade, extensibilidade, adaptabilidade, manutenibilidade, compatibilidade, entre outros.
Confiabilidade	Engloba aspectos como previsibilidade, acurácia de resultados, resistência a falhas, recuperabilidade, entre outros.

A tabela 8 apresenta análise da ferramenta **Monitoring Process** em relação aos atributos identificados.

Tabela 8. Requisitos de Qualidade Monitoring Process

Atributos	Descrição
Segurança	O módulo <b>Monitoring Process</b> é parte integrante do ambiente de engenharia de software centrado em processo, webAPSEE que possui controle de acesso e disponibiliza funções de acordo com o perfil do usuário.
Performance	Mediante atendimento dos requisitos para utilização do sistema, a performance é adequada com tempo de resposta, velocidade de processamento e eficiência considerados em bom nível.
Usabilidade	O módulo está integrado ao ambiente webAPSEE, com fácil disponibilização dos relatórios em arquivos do tipo planilha com bom grau de usabilidade.
Suportabilidade	A arquitetura do ambiente é estruturada em camadas de serviço que permitem maior facilidade em tarefas de evoluções e reparos, extensibilidade, adaptabilidade, manutenibilidade, entre outras
Confiabilidade	Os dados disponibilizados são consistentes e obtidos a partir da execução dos processos no ambiente.

### 5.3 Considerações Finais

Neste capítulo, foi apresentada análise sobre a proposta de Acompanhamento de Projetos em um Ambiente de Gestão de Processos para verificar se os objetivos aos quais se presta a alcançar foram atingidos. A análise crítica foi realizada em duas etapas. Na primeira etapa, foi realizada uma simulação de utilização da proposta em um projeto que já foi realizado em uma organização de desenvolvimento de software. Todos os relatórios de acompanhamento foram gerados e analisados. Essa simulação é um indicador da viabilidade do uso da proposta apresentada neste trabalho para auxiliar a gestão em uma organização real de desenvolvimento de software.

Na segunda etapa, foi realizado um levantamento dos requisitos que uma ferramenta de acompanhamento deveria prover e foi realizada uma comparação entre a abordagem proposta neste trabalho e ferramentas de acompanhamento de projetos mais utilizadas atualmente. O objetivo de avaliação da ferramenta foi atendido através da execução dos passos descritos.

# Capítulo 6 – Considerações Finais

---

Este capítulo apresenta as conclusões obtidas no desenvolvimento deste trabalho, assim como as principais contribuições que ele oferece para a área de monitoração de projetos em ambientes de gerência de processo. Ainda, são apresentados possíveis trabalhos futuros que podem ser realizados a partir deste e as considerações finais desta dissertação.

## 6.1 Principais Contribuições

De uma forma geral, a proposta deste trabalho buscou atender os seguintes requisitos encontrados na literatura:

Tabela 9. Principais Requisitos Atendidos

Requisitos	Atendimento
Necessidade de prover monitoramento de projetos normalmente envolvendo as atividades de: medir os valores reais dos parâmetros de planejamento do projeto, comparar os valores reais com os estimados no plano e identificar os desvios significativos [Chrissis, 2003].	Foi tratado neste trabalho através do fornecimento de relatórios que recuperam os valores estimados, realizados e analisam os desvios de estimativas.
Necessidade de medir regularmente o projeto para identificar variações em relação ao plano de gerenciamento do projeto.	Foi tratado neste trabalho com a incorporação do módulo <b>Monitoring Process</b> a um ambiente de modelagem e execução de processos – o WebAPSEE – que permitiu que os projetos executados nesta ferramenta sejam medidos regularmente a fim de identificar todos os desvios.

Além disso, o trabalho desenvolvido forneceu um conjunto de contribuições, listadas a seguir:

- Proposta de um processo de acompanhamento de projetos utilizando um ambiente de gestão de processos.
- Construção de um módulo de acompanhamento de processos acoplado ao ambiente de gestão de processos WebAPSEE. Um dos méritos da ferramenta é a disponibilização dos relatórios em formatos de planilha eletrônicas com o objetivo de facilitar ao gerente à manipulação das informações apresentadas, uma vez que essa funcionalidade não está disponível atualmente por nenhuma ferramenta *consultada*.
- Realização de uma simulação de acompanhamento de projeto com a ferramenta, obtendo resultados satisfatórios que atendiam os requisitos identificados para a ferramenta.
- Publicação de dois artigos sobre o tema em eventos nacionais. O primeiro (Paxiúba et al, 2005) foi no Seminário Integrado de Software e Hardware – SEMISH e o segundo (Paxiúba et al, 2007) no Simpósio Brasileiro de Qualidade de Software - SBQS.

## 6.2 Trabalhos Futuros

A idéia inicial deste trabalho era um estudo completo de todos os benefícios obtidos a partir do *log* de eventos proposto para o ambiente WebAPSEE, porém devido ao tempo para realização deste trabalho, a proposta está restrita a utilização do *log* para acompanhamento e avaliação de projetos.

Vislumbra-se a oportunidade de utilização das informações obtidas no *log* para apoiar a tomada de decisões dos gerentes, realizar análise *post-mortem* dos processos e ainda fornecer informações para a realização de uma série de atividades dos gerentes como alocação de pessoas, realização de estimativas, entre outras.

Além disso, durante o desenvolvimento deste trabalho, no escopo restrito de acompanhamento e avaliação de projetos, algumas funcionalidades foram observadas. No entanto, devido à limitação de tempo para a conclusão, elas não puderam ser inseridas. A seguir são listadas:

- Geração de relatórios seguindo a técnica de análise do valor agregado utilizada para avaliar o que foi obtido em relação ao que foi realmente gasto e ao que se planejava gastar [Fleming e Koppelman, 1999].
- Alteração na Agenda do Desenvolvedor do Ambiente para possibilitar o registro do percentual de conclusão das atividades e utilização destes valores para geração de relatórios de previsão de término, gasto e esforço por atividade.
- Incluir geração dos relatórios para outros formatos além de planilhas Excel (Open Office ou Br Office, por exemplo). Isto é necessário pois os gráficos dos relatórios não se apresentam da maneira esperada em ambientes diferentes do Excel.
- Emissão de avisos para o gerente através de e-mail ou mensagens no ambiente das atividades que estiverem apresentando com problemas como desvios significativos.

### **6.3 Considerações Finais**

A atividade de acompanhamento de projetos não é uma tarefa trivial, além de ser de grande importância para o sucesso de um projeto de software. As principais abordagens de gerenciamento de projetos definem uma série de práticas que devem ser seguidas com o objetivo de obter êxito na atividade de acompanhamento de projetos.

Este trabalho apresentou uma proposta para atender estas práticas utilizando um ambiente de gestão de processo e seu mecanismo de gerência de eventos para apoiar o gerente na atividade de monitoramento de projetos. Para isto estabeleceu um mini-processo que deve ser seguido pelo gerente e implementou o módulo **Monitoring Process** para emissão de relatórios gerenciais.

Ainda se faz necessária a validação deste trabalho através da utilização efetiva do módulo **Monitoring Process** no acompanhamento de vários projetos distintos e avaliação da efetividade da ferramenta. Esta validação não foi realizada neste trabalho, constituindo-se, portanto do principal trabalho futuro vislumbrado.

# Referências

---

- Bardohl, R. (2000) *GenGED - Visual Definition of Visual Languages based on Algebraic Graph Transformation*. PhD Thesis. Technische Universität Berlin. Kovac Verlag, 2000.
- Chrissis, M.B., Konrad, M. E, Shrum, S. (2003) *CMMI: Guidelines for Process Integration and Product Improvement*, Addison Wesley.
- Cook, J.; Wolf (1995), A. Automating process discovery through event-data analysis. In: International Conference On Software Engineering, Icse, 17., 1995, Seattle, USA. Proceedings... New York: ACM Press, 1995.
- CMMI (2006) : CMMI for Development, Version 1.2 Disponível em <http://www.sei.cmu.edu/cmmi/models/model-v12-components-word.html>
- DeMarco T., *Controlling Software Project*, Prentice Hall, 1982, ISBN 0-13-171711-1.
- Fernandes A.A., *Gerência de Software Através de Métricas*, Atlas S.A, 1995, ISBN 85-224-1264-2.
- Fiorini S.T., Staa A.V., Baptista R.M., (1998) *Engenharia de Software com CMM*, Brasport, 1998, ISBN 85-85840-84-6.
- Fleming , Q. W. , Koppelman, J.M.(1999). *Earned Value Project Management*, 2 Second Edition. Newton Square: Project Management Institute.
- Fuggeta, A. (2000) *Software Process: A Roadmap*. In: International Conference On Software Engineering, Icse, 22., 2000, Limerick, Ireland. **Proceedings...** New York: ACM Press, 2000.
- Grady R.B. (1992), *Practical Software Metrics for Project Management and Process Improvement*, Prentice Hall, 1992, ISBN 0-13-720384-5.
- Heldman, K. (2005) *Gerência de Projetos: Guia para o Exame Oficial PMI. 2ª Edição*. Rio de Janeiro, 2005.
- Henderson-Sellers B., Younessi H., Graham I.S.(1997), *The Open Process Specification*, Addison Wesley, Open Series, 1997

- JXLS (2007): JXLS Project Documentation. Disponível em <http://jxls.sourceforge.net>.
- Jacobson, I., Bylund S. (2002) *A Multi-Agent System Assisting Software Developers. SE Development Team*. Disponível na Internet em [http://www.jaczone.com/papers/intelligent\\_agents.pdf](http://www.jaczone.com/papers/intelligent_agents.pdf)
- LABES. *webAPSEE – Flexible Process Management*. Disponível na Internet por [www](http://www.webAPSEE.com) em <http://www.webAPSEE.com>
- Lima, A.M., LIMA REIS, C. A. , REIS, R. Q. (2006) Análise do Ambiente webAPSEE no atendimento aos requisitos de Gerência de Processos de Software. XX Semana Paraense de Informática (SEPAI/CTIC 2006). Belém, PA. Outubro de 2006.
- Lima, C.A.G., Reis R.Q., Nunes D. J. (1998) Gerenciamento do Processo de Desenvolvimento Cooperativo de Software no Ambiente PROSOFT. In: Simpósio Brasileiro de Engenharia de Software, 12., (SBES'98) **Proceedings...**, Maringá, Brasil , Outubro, 1998, p. 221-236.
- Lima Reis, C. A., Reis, R. Q., Abreu, M. M., Schlebbe, H. , Nunes, D.J.(2002) *Using Graph Transformation as the Semantical Model for Software Process Execution in the APSEE Environment. International Conference on Graph Transformation (ICGT)*. Barcelona. Lecture Notes in Computer Science 2505. Springer, 2002, p. 254-269.
- Lima Reis, C. A. (2003) Uma abordagem flexível para execução de processo de softwares evolutivos. Porto Alegre : PPGC da UFRGS, 2003. Tese de Doutorado.
- Maciel T.M (2000)., *Garantindo a Qualidade na Especificação do Plano de Projeto Iterativo de Software*, XIV Simpósio Brasileiro de Engenharia de Software, Anais do Workshop de Qualidade, João Pessoa, out. 2000.
- Meneses J.B. (2001), *Inspector: Um Processo de Avaliação de Progresso para Projetos de Software*. Dissertação de mestrado do Centro de Informática - UFPE, fev. 2001.
- Paxiúba, C. M. C., Nascimento, L. M. A., Reis, R. Q., Lima Reis, C. A.(2005) *Towards an Event Recording Mechanism for a Process-based Environment*. Anais do Seminário Integrado de Software e Hardware, 32. São Leopoldo, 2005.

- Paxiúba, C. M. C., Pereira, M., Reis, R. Q., Lima Reis, C. A. (2007) *Acompanhamento e Avaliação de Projetos através da Monitoração de Eventos em um Ambiente de Gestão de Processos de Software*. Anais do Simpósio Brasileiro de Qualidade de Software, 2007.
- Prado, Darci (2000). Gerenciamento de projetos nas organizações. Belo Horizonte: Ed. DG, 2000. 203 p. ISBN 8586948233
- Project Management Software (2007). Disponível em <http://www.project-management-software.org>. Último acesso em Junho de 2007.
- PMBOK. (2004) PMI Standards Committee. "A Guide to the Project Management Body of Knowledge", Third Edition, PMI Publishing Division, Philadelphia, USA.
- Reis, R.Q., Lima Reis, C.A., Nunes, D.J. (2002) Automatic Verification of Static Policies on Software Process Models. *Annals of Software Engineering*. Vol.14. Kluwer Academic Publishers, Oct. 2002 .
- RUP, Rational Unified Process. Disponível em <http://www-128.ibm.com/developerworks/rational/products/rup/>. Últmo acesso em Junho de 2007.
- STANDISH GROUP. Chaos. Pesquisa sobre o desenvolvimento de software. Disponível na Internet em [http://www.standishgroup.com/sample\\_research/index.php](http://www.standishgroup.com/sample_research/index.php). Último Acesso em Junho de 2007.
- Van der Aalst, W. M.P, Van Dongen, B. F. (2004) *EMiT: A Process Mining Tool. Application and Theory of Petri Nets*. Lecture Notes in Computer Science 3099. Springer, 2004, p. 454-463.
- Softex (2007) MPS.Br – Melhoria de Processo do Software Brasileiro: Guia Geral, Versão 1.0, disponível em [http://www.softex.br/mpsbr/\\_home/default.asp](http://www.softex.br/mpsbr/_home/default.asp) (Acesso em Junho de 2007)
- SEI (2007) – Software Engineering Institute, informações institucionais disponíveis em <http://www.sei.cmu.edu/cmm> (Acesso em Junho de 2007)
- Vargas, Ricardo Viana (2003). Gerenciamento de projetos: estabelecendo diferenciais competitivos. 5.ed. Rio de Janeiro: Brasport.

Waina R. (2002) Waina, Richard. Capability Maturity Model Benefits. Multi-Dimensional Maturity, Celina, Texas,2002.

webAPSEE (2007): Documento de Referência do Sistema webAPSEE Versão 1.0, disponível em <http://www.webAPSEE.com> (acesso em ...)

# Apêndice A - JXLS

---

JXLS é uma biblioteca Java para gerar arquivos Excel utilizando *templates* XLS. Também pode ser utilizada para ler arquivos XLS e carregar suas informações para Java Beans de acordo com a configuração em arquivos XML.

No *template* XLS pode ser utilizado qualquer funcionalidade do Excel que esta será preservada quando da leitura e escrita em arquivos XLS. Isto significa que pode ser utilizado Gráficos e Macros do Excel nos arquivos templates.

jXLS permite predefinir tags XML nos templates com o objetivo de definir a formatação XLS.

A seguir é apresentado um uso típico de tag XLM.

```
<jx:forEach items="{departments}" var="department">
    {department.name} | {department.chief}
</jx:forEach>
```

Esta linha substitui o conteúdo da linha da planilha pelos valores retornados no resultado da consulta. Estes dados retornam através da coleção de dados *{departments}*.

O contrário também pode ser realizado, a leitura de arquivos XML, para inserir valores em classe java. A seguir é apresentado exemplo.

```
<loop startRow="7" endRow="7" items="department.staff" var="employee"
varType="net.sf.jxls.reader.sample.Employee">
  <section startRow="7" endRow="7">
    <mapping row="7" col="0">employee.name</mapping>
    <mapping row="7" col="1">employee.age</mapping>
    <mapping row="7" col="3">employee.payment</mapping>
    <mapping row="7" col="4">employee.bonus</mapping>
  </section>
  <loopbreakcondition>
    <rowcheck offset="0">
      <cellcheck offset="0">Employee Payment Totals:</cellcheck>
    </rowcheck>
  </loopbreakcondition>
</loop>
</worksheet>
</workbook>
```

O principal elemento do arquivo xml é **workbook** e pode conter qualquer número de elementos **worksheet**. A tag **worksheet** pode conter o atributo nome que indica o nome da planilha excel. No exemplo citado é Sheet1.

O elemento **worksheet** pode conter qualquer número de elementos seções e loop. O elemento **section** representa um simples bloco de células da planilha. A primeira e última linha do bloco são especificados com os atributos **startRow** e **ndRow**.

Para mapear as células do arquivo XLS para as propriedades do Java Bean é utilizada a tag **mapping**, que é utilizada conforme exemplo.

```
<mapping cell="B1">department.name</mapping>
```

O elemento **cell** pode ser utilizado para especificar a célula mapeada e o corpo da tag deve identificar o nome do atributo do Java Bean a ser populado com o valor da célula.

Maiores referências podem ser encontradas em <http://jxls.sourceforge.net/reference/reader.html>

## Apêndice B – Notação APSEE-PML

---

A linguagem de modelagem de processos de software do WebAPSEE (denominada APSEE-PML) é baseada em redes de atividades que podem ser decompostas. Neste formalismo, um modelo de processo pode ser construído a partir de símbolos gráficos conectados e o detalhamento do relacionamento com os outros componentes do modelo (como por exemplo, as informações organizacionais e descrição dos artefatos, entre outros) é feito através de formulários específicos que apóiam essa tarefa [Lima, 2003].

Em seguida serão apresentados uma descrição dos principais componentes visuais da linguagem.

### B.1 Atividades

A Figura 41 mostra a representação adotada para atividades de processos de software distinguindo as atividades decompostas, normais e automáticas. Atividades decompostas serão definidas através de um modelo de processo e também são chamadas de fragmentos de processo, enquanto as atividades do tipo automática e normal são chamadas de atividades-folha [Lima, 2003].

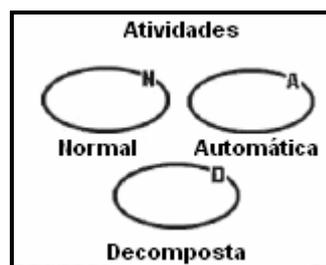


Figura 41. Notação para atividades na APSEE-PML

### B.2 Agentes, Grupos, Recursos e Artefatos

A Figura 42 apresenta a representação adotada para agentes, grupos de agentes, recursos e artefatos.



Figura 42. Notação para Agente, Grupo, Recursos e Artefatos na APSEE-PML

**Agentes** representam quaisquer pessoas envolvidas com o processo de software modelado (desenvolvedores, gerentes e usuários, por exemplo).

**Grupos** representam qualquer grupo da organização.. Os grupos possuem tipos na hierarquia de tipos de grupos no *APSEE-Types* (por exemplo, “revisão”, “programação” e “gerência”). Um grupo possui vários agentes como membros.

**Recursos** representam quaisquer recursos da organização. Para o ambientes os recursos podem ser classificados como:

- **Exclusive** (Exclusivos): São recursos que não podem ser alocados para várias atividades simultaneamente. Portanto, caso duas atividades necessitem do recurso ao mesmo tempo, uma delas deverá aguardar pelo término da outra. Exemplos de recursos exclusivos são computadores e salas;
- **Shareable** (Compartilháveis): Podem ser utilizados por várias atividades simultaneamente, sem necessidade de alocação exclusiva. Assim, não é necessário que a atividade faça alocação e liberação desse tipo de recurso. Uma impressora conectada em rede é um exemplo de recurso compartilhável;
- **Consumable** (Consumíveis): Este tipo de recurso não pode ser utilizado novamente como os anteriores. Uma instância de recurso consumível pode ser utilizada total ou parcialmente por uma atividade. Deste modo, uma vez consumido, o recurso passará para o estado “*Finished*” e não mais poderá ser alocado. Recursos financeiros e papel são exemplos de recursos consumíveis

**Artefatos** são utilizados para representar produtos de software e os objetos gerados durante do desenvolvimento de software.

### B.3 Conexões

As figuras 43 e 44 apresentam os diferentes tipos de conexão e dependência disponíveis no webAPSEE. O webAPSEE trabalha com o conceito de conexões simples e múltiplas, sendo que as conexões possuem dependência (simples e múltiplas) e tipo (múltiplas).

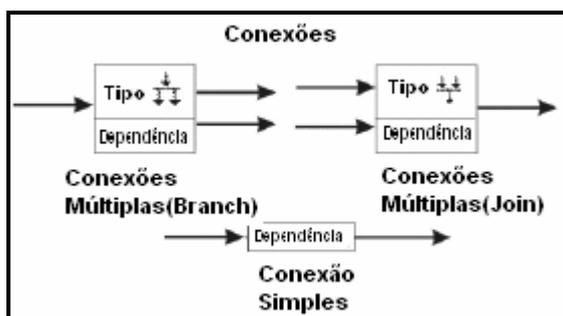


Figura 43. Notação para Conexões na APSEE-PML

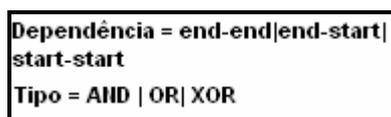


Figura 44. Tipos e Dependência de Conexões

A conexão de seqüência simples tem início e fim em atividades diferentes (que podem ser de qualquer tipo) e o nodo que a representa é um retângulo contendo o tipo de dependência escolhido - “*end-start*”, “*start-start*”, “*end-end*” – conforme é ilustrado nas figuras 43 e 44.

O Tipo de dependência *end-start* indica que a primeira atividade deve terminar para a segunda começar.

O tipo de dependência *start-start* indica que assim que a primeira atividade começar a segunda já pode ser iniciada.

O tipo de dependência *end-end* indica que a segunda atividade somente pode ser encerrada se a primeira atividade já estiver sido encerrada.

A representação de conexões múltiplas na APSEE-PML é feita através de um retângulo dividido onde a parte superior indica o tipo de conexão (Branch ou Join) e o operador lógico associado (AND, OR, XOR), enquanto que a parte inferior indica o tipo de dependência (end-start, start-start, end-end). A Figura 40 mostra a representação gráfica para conexões múltiplas. Por definição, uma conexão Join pode ter vários antecessores e somente um sucessor, enquanto que a conexão Branch pode ter apenas um antecessor e vários sucessores. Na APSEE-PML, os antecessores e sucessores de conexões múltiplas podem ser atividades ou

conexões múltiplas (conexões simples não podem participar de conexões múltiplas). Portanto, o fluxo de controle de um processo pode ser definido mais detalhadamente através da combinação de várias conexões múltiplas encadeadas [Lima, 2003].

## Apêndice C – Extensão WebAPSEE

---

Para a disponibilização do módulo de geração de relatórios de acompanhamento de projetos no ambiente, foram realizados os seguintes acréscimos na arquitetura da aplicação apresentada no capítulo 4.

- **Criação de Serviços no Componente *ReportsFacade* (Camada Servidora)**

Para implementar a geração de relatórios em formatos de planilhas do Microsoft Excel, foi utilizada a biblioteca jXLS [jXLS].

A seguir, exemplo de um dos serviços implementados, responsável por gerar o relatório *CostDeviation*, utilizando a API jXLS.

```
1 - public byte[] generateScheduleDeviationReport_Excel( String  
processIdent, 2 - String path ) throws RemoteException  
3 - { HibernateSession.setSessionFactory(factory);  
4 - return  
ReportsGenerator.generateScheduleDeviationReport_Excel( processIdent,  
path ); } ....
```

Listagem 1 – Trecho da Classe ReportsRemotImpl

Este trecho de código se refere a **Classe ReportsRemotImpl** que na aplicação é responsável pela chamada aos métodos que geram os relatórios da aplicação. A linha 4 realiza a chamada ao método que gera o relatório de Desvio de Cronograma. Passando como parâmetro o nome do processo e o caminho onde o relatório deve ser gerado.

```

1 - public static byte[] generateScheduleDeviationReport_Excel( String
2 - processIdent, String path )
3 - {String templateFilePath =
4 - "br/ufpa/labes/server/reports/xlsTemplates/ScheduleDeviationReportTemplate.xls";
5 - QueryDAO query = new QueryDAO();
6 - List<Object[]> data
query.getDataForScheduleDeviationReport( processIdent );
7 -     if ( data == null ) return null;
8 -     XLSTransformer transformer = new XLSTransformer();
9 -     List<ScheduleDeviationBean> beansList = new
10 - ArrayList<ScheduleDeviationBean>();
11 - Iterator<Object[]> dataIterator = data.iterator();
12 - while ( dataIterator.hasNext() ) {
13 -     Object[] reportData = dataIterator.next();
14 -     String activityName = (String) reportData[0];
15 -     String actualSchedule = (String) reportData[1];
16 -     String estimatedSchedule = (String) reportData[2];
17 -     Float scheduleDeviation = (Float) reportData[3];
18 -     ScheduleDeviationBean reportBean = new ScheduleDeviationBean(
19 -     activityName, actualSchedule, estimatedSchedule,
scheduleDeviation);
20 -     beansList.add( reportBean );
21 -     Map<String, List> beans = new HashMap<String, List>();
22 -     beans.put( "reportData", beansList );
23 - try
24 - {InputStream ioStream =
25 - ReportsGenerator.class.getClassLoader().getResourceAsStream(
templateFilePath );
26 - HSSFWorkbook wb = transformer.transformXLS( ioStream, beans );
27 - HSSFCell cell = wb.getSheet( "Plan1" ).getRow( 0 ).getCell( (short)0
);
28 - cell.setCellValue( cell.getStringCellValue() + " (" + processIdent +
")" );
29 - ByteArrayOutputStream output = new ByteArrayOutputStream();
30 - wb.write( output );
31 - byte[] bytes = output.toByteArray();
32 - output.close();
33 - return bytes;
34 - }catch ( ParsePropertyException e )
35 -     { e.printStackTrace();}
36 -     catch ( IOException e )
37 -     { e.printStackTrace();}
38 -     return null;} ....

```

Listagem 2 – Trecho da Classe ReportsRemoteInterface

A classe ReportsRemoteInterface é responsável pela implementação da geração dos relatórios. O trecho exibido na listagem 2 é responsável pela geração do relatório de desvios de custos. A linha 3 recupera o arquivo Excel que contém o modelo do relatório. A linha 6 é responsável por realizar a chamada ao método que realiza a consulta no ambiente, retornando o custo estimado e realizado de todas as atividades referentes ao processo que foi enviado como parâmetro. O trecho referente a linha 25 a 29 é responsável por gerar o arquivo e preencher as células

da planilha com as informações retornadas pela consulta. O arquivo é gerado no caminho especificado como parâmetro.

- **Criação de Consultas no Componente DAO (Camada Servidora)**

As novas consultas às base de conhecimento necessárias para a disponibilização dos relatórios foram incorporadas no módulo DAO (Camada Servidora) da aplicação. A seguir é apresentado trecho da implementação referente a consulta para disponibilização do relatório *Schedule Deviation*. O trecho de exemplo recupera os prazos estimados e realizados de todas as atividades de um processo específico, além de realizar os cálculos referentes aos desvios de estimativa. A consulta (Linha 5 a 32) obtém os valores realizados a partir do log de eventos do ambiente. As linhas 43 a 53 realizam o cálculo do desvio de prazo por atividade.

```
1 - public List<Object[]> getDataForScheduleDeviationReport( String  
processIdent )  
2 - {  
3 - try  
4 - {Session session = BaseDAO.getNewSession();  
5 - String hql = "SELECT at.name, " +  
6 - "a.name, " +  
7 - "e.when, " +  
8 - "est.unit, " +  
9 - "est.value, " +  
10 - "ce.description " +  
11 - "FROM " + Event.class.getName() + " e, " +  
12 - AgendaEvent.class.getName() + " ae, " +  
13 - CatalogEvents.class.getName() + " ce, " +  
14 - Task.class.getName() + " t, " +  
15 - Activity.class.getName() + " at, " +  
16 - ProcessAgenda.class.getName() + " pa, " +  
17 - TaskAgenda.class.getName() + " ta, " +  
18 - Agent.class.getName() + " a, " +  
19 - ActivityEstimation.class.getName() + " ate, " +
```

```

20 - Estimation.class.getName() + " est " +
21 - "WHERE ae.oid = e.oid " +
22 - AND ae.oid = ce.theAgendaEvent.oid " +
23 - "AND ae.theTask.oid = t.oid " +
24 - "AND t.theNormal.oid = at.oid " +
25 - "AND at.oid = ate.activity.oid " +
26 - "AND ate.oid = est.oid " +
27 - "AND pa.oid = t.theProcessAgenda.oid " +
28 - "AND ta.oid = pa.theTaskAgenda.oid " +
29 - "AND a.oid = ta.theAgent.oid " +
30 - "AND (e.theLog.theProcess.ident=:processID) " +
31 - "AND est.metricDefinition.oid = 7 " +
32 - "ORDER BY at.name";

33 - Query query = session.createQuery( hql );
34 - query.setParameter( "processID", processIdent );
35 - List<Object[]> data = query.list();
36 - List<Object[]> result = new ArrayList<Object[]>();

37 -     if ( data.size() == 0 )

38 -         return null;
39 - BaseDAO.closeSession( session );
40 -     data = sortDataForExcelReport( data, 0, 1, 5 );
41 -     data = groupByActivity2( data, 0, 5, 2 );
42 -     result = new ArrayList<Object[]>();

43 - for ( int i = 0; i < data.size(); i++ )          {
44 - Object[] currentObject = data.get( i );
45 - Object[] resultObject = new Object[ 4 ];
46 - resultObject[ 0 ] = currentObject[ 0 ];
47 -     float deviation = ( ( Float)currentObject[ 2 ] /
48 -(Float)currentObject[ 4 ] ) - 1 );
49 -     resultObject[ 3 ] = deviation;

50 -     resultObject[ 1 ] = formatFloatToTime( (String)currentObject
51 - [3 ], (Float)currentObject[ 2 ] );
52 -     resultObject[ 2 ] = formatFloatToTime( (String)currentObject
53 - [ 3 ], (Float)currentObject[ 4 ] );
54 -     result.add( resultObject );}
55 -     return result;}
56 - catch ( DAOException e )
57 - {e.printStackTrace();}
58 - catch ( HibernateException e )
59 - {e.printStackTrace();      }
60 - catch ( Exception e )
61 - {e.printStackTrace();}
62 - return null;

```

Listagem 3 – Trecho da Classe QueryDAO

- **Criação de Classes no Componente ReportsClients (Camada Cliente)**

O cliente atual foi estendido, com o acréscimo de novos métodos para realizar as chamadas aos novos relatórios disponibilizados pelo módulo de acompanhamento de projetos.

O trecho apresentado na listagem 4 é referente a Classe ReportsInterface responsável por realizar as chamadas para criação dos relatórios passando como parâmetro o processo que está sendo monitorado e o caminho onde está sendo gerado.

```
1 - public byte[] generateProjectScheduleMonitoringReport_Excel( String  
processIdent, String path ) throws RemoteException  
2 - { try {  
3 - return  
getReportRemoteInterface().generateProjectScheduleMonitoringReport_Excel(  
processIdent, path );  
4 -     } catch (RemoteException e) {  
5 -         e.printStackTrace();}  
6 -     return null;}
```

Listagem 4 – Trecho da Classe ReportsInterface

