



Universidade Federal de Pernambuco

Centro de Informática

Especialização em Gestão da Tecnologia da Informação

# **Inspector Panel: Um Painel de Controle para Acompanhamento de Progresso de Projetos de Software**

por

Aristides Vicente de Paula Neto

Recife, dezembro de 2008

---

## Agradecimentos

---

A realização desta monografia só foi possível graças à colaboração direta e indireta de muitas pessoas. Meus sinceros agradecimentos a todas elas e em particular:

A Deus, fonte de todo o conhecimento e da vida eterna, pela sua presença constante na minha vida, pelo auxílio nas minhas escolhas e por realizar os desejos do meu coração. Porque Dele, por Ele e para Ele são todas as coisas.

Ao meu pai, José Régis, e à mãe, Mirian de Paula, pelo amor, carinho, ensinamentos, paciência e confiança dedicada a minha pessoa.

Ao professor Hermano Perrelli de Moura pela amizade, confiança e por sua valiosa orientação na condução deste trabalho.

Ao Centro de Informática pela excelência.

A diretoria da empresa xE, por fornecer as informações necessárias para realização do estudo de caso.

A Roberta Queiroz, secretária do curso, pelo apoio e dedicação incondicional a todos os alunos do curso.

E finalmente aos professores e colegas que encontrei durante este curso de Especialização em Gestão de Tecnologia da Informação.

---

## Resumo

---

O atual mundo globalizado faz com que as empresas precisem estar mais preparadas e competitivas, bem como as obriga a procurarem melhores técnicas para serem capazes de produzirem produtos ou serviços, em menor tempo, maior qualidade e menor custo possível. Entretanto a criação de um novo produto não consiste em uma tarefa simples.

No contexto apresentado, este trabalho tem como objetivo especificar e construir uma ferramenta (web) para facilitar e auxiliar o Gerente de Projetos em suas atribuições. O Gerente de Projetos precisa avaliar o andamento do projeto, medir o progresso, comparar o realizado com o planejado, fazer ajustes se necessários e tomar outras decisões para garantir que os objetivos do projeto sejam alcançados.

O escopo deste trabalho é definir e implementar um painel de controle para acompanhamento dos custos, do cronograma e do progresso de projetos de software, baseado no *Rational Unified Process* (RUP) [18] e nos critérios já estabelecidos pelo processo de avaliação de progresso para projetos de software, o Inspector [11].

O Inspector define um conjunto de atividades que visam tornar o acompanhamento de projetos uma tarefa sistemática, onde os problemas no desenvolvimento são identificados antes que atinjam proporções maiores, e mecanismos são inseridos de modo a permitir que o gerente de projeto forneça uma avaliação precisa do status do projeto ao cliente.

**Palavras-chaves:** Gerência de Projetos; Acompanhamento de Progresso; RUP; Inspector; Inspector Panel;

---

## Abstract

---

The current globalized world means that companies need to be more prepared and competitive, and require them to seek better techniques to be able to produce goods or services in less time, higher quality and lower cost. Meanwhile the creation of a new product is not a simple task.

In the context presented, this paper aims to clarify and build a tool (web) to facilitate and assist the Project Manager in its mission. The Project Manager needs to assess the progress of the project, measure progress, compare with the planned done, make adjustments if necessary and take other decisions to ensure that the goals of the project are achieved.

The scope of this study is to define and implement a control panel for monitoring of costs, schedule and the progress of projects of software, based on Rational Unified Process (RUP) [18] and the criteria already established by the process of evaluation of progress to software projects, the Inspector [11].

The Inspector defines a set of activities that aim to make the project a task tracking system, where problems are identified in development before they reach major proportions, and mechanisms are inserted to allow the project manager will provide an accurate assessment of the status the project to the client.

**Keywords:** Project Management; Monitoring Progress; RUP; Inspector; Inspector Panel;

---

## Sumário

---

<b><u>CAPÍTULO 1</u></b>	<b><u>INTRODUÇÃO.....</u></b>	<b><u>1</u></b>
1.1	MOTIVAÇÃO.....	2
1.2	OBJETIVOS.....	2
1.3	TRABALHO REALIZADO.....	3
1.4	ORGANIZAÇÃO DO DOCUMENTO.....	3
<b><u>CAPÍTULO 2</u></b>	<b><u>GERÊNCIA DE PROJETOS.....</u></b>	<b><u>5</u></b>
2.1	GERENCIAMENTO DE PROJETOS.....	5
2.2	ACOMPANHAMENTO DE PROJETOS.....	7
2.3	PAPEL DO GERENTE DE PROJETOS.....	9
2.4	GERENCIAMENTO DE PROJETOS ITERATIVOS E INCREMENTAIS DE SOFTWARE.....	11
	RUP – RATIONAL UNIFIED PROCESS.....	11
<b><u>CAPÍTULO 3</u></b>	<b><u>O PROCESSO INSPECTOR.....</u></b>	<b><u>14</u></b>
3.1	CONCEITOS.....	14
3.2	ATIVIDADES DO INSPECTOR.....	16
3.3	VISÕES DO INSPECTOR.....	18
	VISÃO DE DESEMPENHO.....	18
	VISÃO DE FUNCIONALIDADE.....	20
<b><u>CAPÍTULO 4</u></b>	<b><u>INSPECTOR PANEL.....</u></b>	<b><u>24</u></b>
4.1	OBJETIVOS DA FERRAMENTA.....	24
4.2	REQUISITOS DA FERRAMENTA.....	25

---

<b>4.3</b>	<b>FUNCIONALIDADES .....</b>	<b>28</b>
<b>4.4</b>	<b>IMPLEMENTAÇÃO .....</b>	<b>30</b>
	AMBIENTE DO INSPECTOR PANEL .....	30
	ARQUITETURA DO INSPECTOR PANEL.....	31
<b><u>CAPÍTULO 5 ESTUDO DE CASO.....</u></b>		<b>33</b>
<b>5.1</b>	<b>ATIVIDADES REALIZADAS .....</b>	<b>34</b>
	AVALIANDO O STATUS DAS MÉTRICAS NA ORGANIZAÇÃO.....	34
	ADAPTANDO A ORGANIZAÇÃO .....	35
	INSTANCIANDO INSPECTOR.....	36
	AVALIANDO O PROGRESSO E DESEMPENHO DAS EQUIPES .....	36
	ACOMPANHAMENTO DO CRONOGRAMA .....	44
	ACOMPANHAMENTO DOS CUSTOS .....	45
<b>5.2</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>46</b>
<b><u>CAPÍTULO 6 CONCLUSÃO.....</u></b>		<b>48</b>
<b>6.1</b>	<b>CONTRIBUIÇÕES .....</b>	<b>48</b>
<b>6.2</b>	<b>DIFICULDADES ENFRENTADAS.....</b>	<b>49</b>
<b>6.3</b>	<b>TRABALHOS FUTUROS.....</b>	<b>49</b>
<b><u>REFERÊNCIAS BIBLIOGRÁFICAS.....</u></b>		<b>51</b>
<b><u>APÊNDICES.....</u></b>		<b>54</b>

---

---

## Lista de Figuras

---

FIGURA 2.1 – DIFERENTES ABORDAGENS DE ACOMPANHAMENTO DE PROJETOS .....	8
FIGURA 2.2 – ARQUITETURA GERAL DO RUP .....	12
FIGURA 2.3 – PASSAGEM PELAS DISCIPLINAS (ITERAÇÃO) .....	13
FIGURA 3.1 – FLUXO DE ATIVIDADES DO PROCESSO INSPECTOR .....	16
FIGURA 4.1 – OBTENÇÃO DOS REQUISITOS PARA FERRAMENTA INSPECTOR PANEL .....	26
FIGURA 4.2 – DIAGRAMA DE CASOS DE USO DO INSPECTOR PANEL .....	27
FIGURA 4.3 – ARQUITETURA DO INSPECTOR PANEL.....	32
FIGURA 5.1 – PROGRESSO DO PROJETO XP .....	37
FIGURA 5.2 – PROGRESSO DOS CASOS DE USO DO PROJETO XP.....	38
FIGURA 5.3 – VISÃO DE DESEMPENHO OBTIDA NA PRIMEIRA INSPEÇÃO .....	40
FIGURA 5.4 – VISÃO DE DESEMPENHO OBTIDA NA SEGUNDA INSPEÇÃO .....	42
FIGURA 5.5 – VISÃO DE DESEMPENHO OBTIDA NA TERCEIRA INSPEÇÃO.....	44
FIGURA 5.6 – GRÁFICO DE <i>GANTT</i> .....	45
FIGURA 5.7 – EXEMPLO: GRÁFICO DE CUSTO POR FASES .....	46

---

## Lista de Tabelas

---

TABELA 5.1 – DEFINIÇÃO DOS PESOS DO PROJETO .....	36
TABELA 5.2 – VISÃO DE FUNCIONALIDADE OBTIDA NA PRIMEIRA INSPEÇÃO.....	39
TABELA 5.3 – VISÃO DE FUNCIONALIDADE OBTIDA NA SEGUNDA INSPEÇÃO.....	41
TABELA 5.4 – VISÃO DE FUNCIONALIDADE OBTIDA NA TERCEIRA INSPEÇÃO .....	43

---

## Capítulo 1 Introdução

---

No cenário econômico e tecnológico atual, globalizado, as transformações no mercado são constantes. As organizações que visam o crescimento, o sucesso e a sobrevivência no mercado nacional e internacional, devem estar preparadas para grandes desafios técnicos e o aproveitamento das oportunidades de negócios através de um processo contínuo de transformação e melhoria.

Nos últimos anos, a necessidade de mudança na forma de condução dos negócios imposta pela crescente competitividade entre as empresas, a luta pela sobrevivência e pelo progresso das organizações passaram a depender, também, de sua capacidade de identificar e executar as melhores mudanças. O planejamento estratégico, estabelecido pelas organizações, permite às lideranças identificar e selecionar as melhores estratégias e mudanças, e a gestão de projetos passam a ser o agente executor destas mudanças. E cada mudança é um projeto [4].

Devido à existência de projetos de software cada vez mais complexos e com muitas pessoas envolvidas, o gerenciamento de projetos tem participação vital para o sucesso e é fundamental o uso de ferramentas para o planejamento e acompanhamento do software, durante todo o seu ciclo de vida.

Gerenciar projetos de software, visando aumentar a competência da organização na identificação, avaliação, planejamento e execução de projetos envolvem um conjunto bastante variado de aspectos que influenciam no desenvolvimento de um sistema, dentre os quais temos: aspectos cognitivos, que estão relacionados aos fatores humanos envolvidos no projeto, aspectos econômicos, que são mais influenciados pelos interesses comerciais da organização, e aspectos técnicos, que se relacionam às etapas e atividades relacionadas ao desenvolvimento do projeto. Para Kerzner, a gerência de projetos, é o planejamento, programação e controle de uma série de tarefas integradas de forma a atingir seus objetivos

---

com êxito, para benefício dos participantes do projeto [1].

## **1.1 Motivação**

Avaliar o andamento de projetos de software, medir o progresso e comparar o realizado com o previsto; se preciso fazer ajustes necessários e tomar decisões para garantir a execução dos objetivos do projeto verificando os resultados obtidos em relação às estimativas, aos compromissos e aos planos que foram documentados, sempre fazendo todos os ajustes necessários.

O processo Inspector define um conjunto de atividades que buscam sistematizar o uso de técnicas e métricas para avaliação do progresso técnico de um projeto de desenvolvimento de software orientado a objetos.

Neste contexto, o progresso do projeto é obtido pela comparação dos valores realizados com os estimados e quando o status real desvia de forma significativa dos valores esperados, devem ser tomadas as ações corretivas apropriadas. Para atender a todos estes requisitos sem o apoio de uma ferramenta é uma tarefa que implica em grande esforço, pois significa acompanhar e coletar manualmente uma grande quantidade de informações sobre a realização de todas as atividades previstas em um projeto. Outras variáveis também implicam neste processo manual, entre eles, destaca-se o tempo. Portanto, apoiar o gerente na realização destas tarefas é o objetivo maior da abordagem apresentada neste trabalho.

## **1.2 Objetivos**

Em um contexto maior o objetivo deste trabalho é especificar um sistema de acompanhamento de progresso de projetos de software, atendendo aos requisitos do processo Inspector, especialmente aqueles vinculados com a coleta e processamento de dados.

Outro objetivo é o desenvolvimento de uma ferramenta web para automatizar as atividades do processo Inspector a fim de que ela possa ser usada em pequenas, médias e

---

grandes organizações desenvolvedoras de software, durante todo o seu ciclo de vida.

Destacam-se, ainda, os indicadores visuais (gráficos) e relatórios para o acompanhamento do progresso funcional do software, com isso o gerente pode avaliar o progresso do projeto em qualquer fase do mesmo. A ferramenta ainda visa o detalhamento do status das atividades definidas para cada fase do projeto e identificar possíveis problemas na execução ou desenvolvimento dos casos de uso.

### 1.3 Trabalho Realizado

Esta seção descreve o trabalho realizado e a metodologia empregada para o desenvolvimento deste trabalho.

A princípio foi realizado um estudo geral em gerenciamento de projeto [9] [18] [10] identificando os pontos chaves e posteriormente definindo um escopo preliminar do trabalho; em seguida, um estudo aprofundado do processo Inspector, que veio agregar mais conhecimento, bem como definir o escopo deste trabalho; a identificação dos requisitos e funcionalidades da ferramenta; o planejamento das atividades na construção da ferramenta; a elaboração de uma *homepage* para disponibilizar toda documentação produzida e para o acompanhamento das atividades, através do cronograma; aplicação da ferramenta em um projeto real; análise e condensação dos *feedbacks* capturados durante a aplicação, expondo pontos de melhoria e sugestões; e por fim, a escrita da monografia.

### 1.4 Organização do Documento

Além desta introdução, esta monografia conta com mais cinco capítulos, como se segue:

**Capítulo 2 – Gerência de Projeto:** Esse capítulo descreve o estado da arte em gerência de projetos, posteriormente uma explanação sobre o acompanhamento de projetos em diferentes abordagens, e por fim, destaca-se a figura do gerente de projetos.

**Capítulo 3 – O Processo Inspector:** Esse capítulo apresenta a conceituação do processo Inspector, abordando as visões de desempenho e de funcionalidade.

---

**Capítulo 4 – Inspector Panel:** Esse capítulo apresenta a ferramenta Inspector Panel, desenvolvida com base no processo de avaliação de progresso de projetos de software, o Inspector.

**Capítulo 5 – Estudo de Caso:** Esse capítulo tem como objetivo a exploração de todas as funcionalidade da ferramenta em um projeto real, realizar inspeções para o acompanhamento de progresso, e capturar as experiências dos envolvidos durante este estudo de caso.

**Capítulo 6 – Conclusão:** Esse capítulo apresenta uma conclusão acerca do trabalho desenvolvido, resumindo os principais pontos e suas contribuições, dificuldades enfrentadas e proposta de trabalhos futuros.

---

## Capítulo 2 Gerência de Projetos

---

O desenvolvimento de um produto é uma seqüência de atividades que uma organização empreende com a finalidade de conceber, desenvolver e comercializar novos produtos. A criação deste novo produto consiste em definir características próprias, design e funcionalidades específicas. Com a crescente demanda do mercado consumidor por novos produtos e produtos de qualidade, é notória a importância da gerência ou gestão de projetos.

Entretanto a criação de um novo produto não consiste em uma tarefa simples. A gestão de recursos envolvidos na criação deste produto possui características aparentemente contraditórias tais como: a criatividade, a novidade, a pressão do tempo para chegar a um resultado, a inexperiência do desenvolvimento, a incerteza e a necessidade por resultados tangíveis.

A gerência ou gestão de projetos é fundamental em qualquer organização que queira obter resultados tangíveis e melhorar tanto seus processos quanto seus produtos finais. Sobretudo é importante que as duas áreas sejam contempladas em conjunto, ou seja, não adianta nada efetuar um planejamento se não for seguido de acompanhamento e é ainda pior tentar estabelecer procedimentos de acompanhamento em projetos não planejados ou mal planejados.

### 2.1 Gerenciamento de Projetos

Gerenciamento de Projetos (GP) é a aplicação de conhecimentos, habilidades e técnicas para projetar atividades que visem atingir os requerimentos do projeto [9]. Por sua vez, os projetos são empreendimentos ou conjuntos de atividades únicos e não repetitivos com metas fixadas dentro de parâmetros de custo, qualidade e prazo [9].

Humphrey define um projeto como sendo um esforço de trabalho realizado por um

---

grupo de pessoas com um objetivo definido, dentro do prazo estabelecido, e com recursos limitados [8]. Para Kerzner, um projeto é um empreendimento com objetivos identificáveis que utiliza recursos e opera sob pressões de prazos, custos e qualidade [1].

A Gerência de Projetos envolve criar um equilíbrio entre as demandas de escopo, tempo, custo, qualidade e bom relacionamento com o cliente. O sucesso na gestão de um projeto está relacionado ao alcance dos seguintes objetivos: entrega dentro do prazo previsto, dentro do custo orçado, com nível de desempenho adequado, aceitação pelo cliente, atendimento de forma controlada às mudanças de escopo e respeito à cultura da organização [12].

Em [7] apontam alguns benefícios da gerência ou gestão de projetos:

- Melhora da produtividade, fornecendo o caminho mais direto para a solução de problemas.
- Aumento dos lucros pela redução do desperdício de tempo e de energia em soluções erradas.
- Melhora no estado de ânimo dos funcionários mediante maior satisfação no emprego.
- Melhores tomadas de decisões na continuação e no término dos esforços de trabalho.
- Melhor posição de competitividade dentro da indústria com a apresentação de resultados mais rápidos para as situações.
- Apresentação mais rápida de produtos que satisfaçam às exigências dos clientes;
- Menor esforço (horas de trabalho) com melhores resultados.
- Confiança na capacidade de completar o trabalho.

Para realizar um gerenciamento de projetos, são necessários alguns conhecimentos. De acordo com o *Project Management Body of Knowledge (PMBOK)*,

---

*“O Conjunto de conhecimentos em gerenciamento de projetos é a soma dos conhecimentos intrínsecos à profissão de gerenciamento de projetos. Assim como em outras profissões, como advocacia, medicina e contabilidade, o conjunto de conhecimentos pertence aos profissionais e acadêmicos que o aplicam e o desenvolvem. O Conjunto de conhecimentos em gerenciamento de projetos completo inclui práticas tradicionais comprovadas amplamente aplicadas, além de práticas inovadoras que estão surgindo na profissão, inclusive materiais publicados e não publicados. Como resultado disso, o Conjunto de conhecimentos em gerenciamento de projetos está em constante evolução.” [PMBOK, 2004, p.19]*

Para que o gerente de projetos desempenhe suas atividades e às suas responsabilidades é necessária a execução de algumas atividades específicas. Tais atividades são definidas em diferentes abordagens, a exemplo do RUP, do PMBOK, do modelo CMMI e do MPS.Br. Um ponto comum em todas as abordagens citadas é considerar a atividade de acompanhar a execução do projeto como essencial ao gerenciamento de projeto, já que possibilita a detecção e correção de problemas.

## **2.2 Acompanhamento de Projetos**

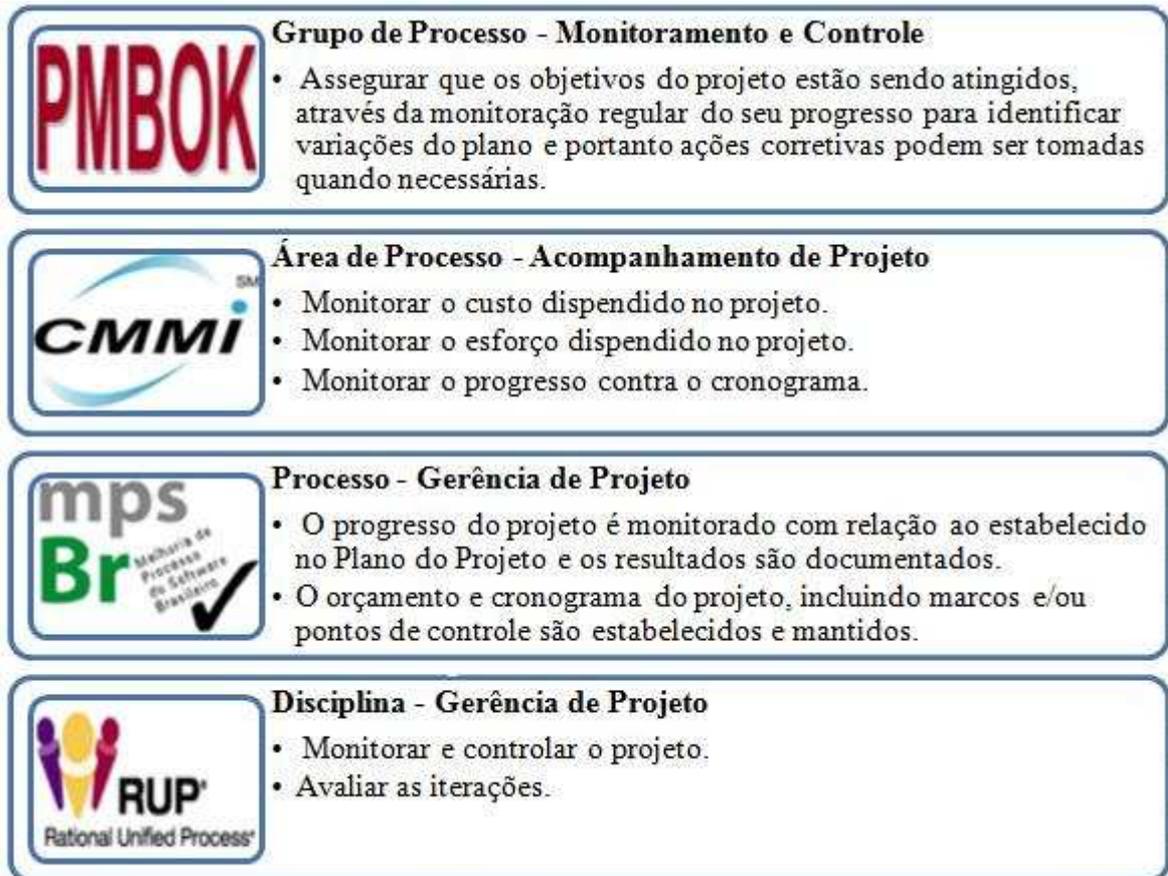
A obediência a um plano de boa qualidade e a aderência a um processo não assegura que os resultados sejam satisfatórios, ou seja, os artefatos produzidos sejam de qualidade. O acompanhamento é o processo de monitorar a execução de um plano.

O acompanhamento de projeto deve avaliar, a cada instante, se os projetos em desenvolvimento apresentam o alinhamento estratégico desejado, os riscos aceitáveis e o desempenho esperado. No caso de serem observados desvios, deve-se procurar identificar suas causas [3].

Um dos objetivos do gerenciamento de projetos é identificar, estabelecer, coordenar, e monitorar as atividades, tarefas e recursos necessários de um projeto para produzir um produto e/ou serviço, dentro do contexto dos requisitos e restrições do projeto.

---

Outro objetivo é acompanhar, dirigir e registrar os resultados produzidos e os recursos consumidos no desenrolar da execução do projeto, confrontando-os com o que era esperado pelos planos [5].



**Figura 2.1 – Diferentes abordagens de acompanhamento de projetos**

A Figura 2.1 – Diferentes abordagens de acompanhamento de projetos, apresentam as atividades e resultados esperados do acompanhamento de projetos de software em diferentes abordagens: no grupo de processo, Monitoramento e Controle, do *PMBOK Guide*; na área de processo, Acompanhamento de Projeto, do modelo CMMI; no Processo, Gerência de Projetos, do modelo de referência MPS.Br; e na disciplina de Gerência de Projetos, do RUP. Para todas as abordagens apresentadas, o acompanhamento de projetos engloba o monitoramento das atividades, do cronograma e dos custos do projeto.

---

## 2.3 Papel do Gerente de Projetos

O Gerente de Projetos é o elemento instituído de poder para gerir, dirigir, e administrar determinado projeto ou projetos. Para isso, é necessário estarem preparados para poder praticar e desempenhar bem seu papel trazendo os benefícios que as organizações desejam. Ou seja, utiliza-se de toda a sua experiência e de seus especialistas para dimensionar o prazo e os recursos necessários, definindo cada atividade a ser desenvolvida, quem deverá realizá-la, em quanto tempo e em que momento durante o ciclo de vida do projeto.

Dependendo da restrição de tempo imposta pelo cliente, o Gerente pode alocar mais ou menos recursos para viabilizar a conclusão do projeto no prazo determinado. É comum também utilizar um *buffer time* de segurança, ou seja, uma quantidade de horas a mais como folga (geralmente um percentual do tempo total estimado), para eventuais atrasos que possam ocorrer durante a execução das atividades.

Segundo [7] os gerentes de projetos devem tanto liderar como gerenciar o projeto. Para desempenhar tais papéis o gerente precisa ter as seguintes competências:

- Ter compreensão geral da tecnologia envolvida no projeto.
- Ter habilidades interpessoais que promovam um ambiente cultural para a equipe e para os *stakeholders* do projeto, de modo a refletir confiança, lealdade, comprometimento e respeito. Os *stakeholders* são indivíduos e organizações envolvidas no projeto, ou que serão afetados positivamente ou negativamente pelo resultado final.
- Compreender os processos gerenciais e suas aplicações ao projeto
- Ter visão do contexto “sistêmico” do projeto.
- Ser capaz de tomar e implementar decisões acerca do projeto.
- Ser capaz de produzir os resultados desejados no projeto.

Segundo [15] o gerente precisa ter aptidões e as classifica em três: os conhecimentos (o que sabe?), as atitudes (suas reflexões sobre si, o ambiente e outras

---

interfaces), e as habilidades (o que sabe fazer?):

- Conhecimento – estar relacionado com a competência técnica pertinente ao projeto no qual o indivíduo vai gerenciar. Ou seja, o gerente precisa ter conhecimento sobre a organização e suas práticas, políticas e valores; conhecimento administrativo, financeiro e recursos humanos; domínio nos métodos de pesquisa; conhecimento e competência técnica em gerência de projetos.
- Atitudes – os projetos são diretamente afetados e a equipe fortemente influenciada pelas atitudes do gerente. Ou seja, o posicionamento do gerente em relação a aspectos internos e externos afeta todos os *stakeholders*.
- Habilidades – estar relacionada com a capacidade de planejar, organizar e coordenar o projeto, as equipes e todos os fatores envolvidos. Ou seja, o gerente precisa ter a capacidade de trabalhar em equipe e resolver conflitos; capacidade de liderança e criatividade; capacidade de gerir confiança e alocar recursos necessários para execução das atividades; capacidade de negociação.

Neste contexto, o gerente de projetos desempenhado bem o seu papel, alinhado a sua experiência e a boas práticas de gerenciamento de projetos produz resultados expressivos para a organização. Segundo Prado, os resultados expressivos esperados pelas organizações são: (1) redução no custo e prazo de desenvolvimento de novos produtos; (2) aumento no tempo de vida dos novos produtos; (3) aumento de vendas e receita; (4) aumento do número de clientes e de sua satisfação e (5) aumento da chance de sucesso nos projetos [12].

Em [7] e [15] exemplificam o que se espera de um gerente de projeto. Para a *Rational Unified Process* (RUP) um gerente de projetos deve realizar as seguintes atividades [16]:

- Iniciar o projeto – utilizando o documento de visão e o caso de negócio, o gerente realiza uma estimativa de custo e estabelece um cronograma.
- Estabelecer e manter todos os planos do projeto – plano de testes, plano de iteração,

---

plano de qualidade, plano de métricas, plano de gerência de riscos, etc.

- Iniciar, executar e finalizar o projeto, uma fase ou uma iteração – planejar com detalhes as atividades e os artefatos a serem produzidos a cada fase ou iteração.
- Monitoramento do projeto através de revisões ou inspeções nos resultados alcançados, a fim de garantir a conformidade do plano de projeto e a qualidade desejada.

## **2.4 Gerenciamento de Projetos Iterativos e Incrementais de Software**

A idéia de um gerenciamento de projetos iterativos e incrementais de software consiste no desenvolvimento de um sistema de software incremental, permitindo a equipe do projeto obter as seguintes vantagens: (1) pequenos objetivos, foco em curto-prazo; (2) permite *feedback* dos usuários desde cedo; (3) os riscos mais críticos são resolvidos no início; (4) testes e integração são realizados desde o início do projeto.

O passo fundamental está em iniciar o desenvolvimento com um subconjunto de requisitos de software e iterativamente alcançar evoluções subseqüentes das versões até todo o sistema está implementado.

### **RUP – Rational Unified Process**

Os processos do RUP oferecem uma abordagem prescritiva nas melhores práticas de engenharia de software. Utiliza a abordagem da orientação a objetos em sua concepção e é projetado e documentado utilizando a notação UML (*Unified Modeling Language*) [17] para ilustrar os processos em ação. Tem como características principais ser dirigido a casos de uso, centrado na arquitetura, iterativo e incremental.

A Figura 2.2 apresenta a arquitetura geral do RUP, onde o projeto passa por quatro fases básicas. Estas fases são:

- Iniciação – entendimento da necessidade do cliente e visão do projeto.

- Elaboração – especificação e abordagem dos pontos de maior risco.
- Construção – desenvolvimento principal do sistema.
- Transição – ajustes, implantação e transferência de propriedade do sistema.

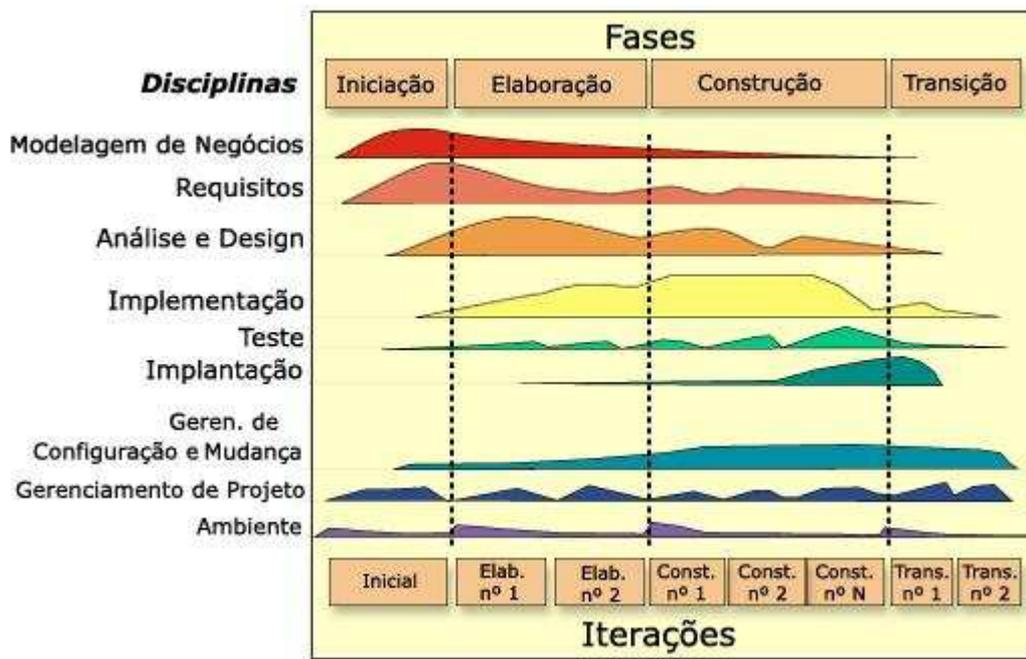
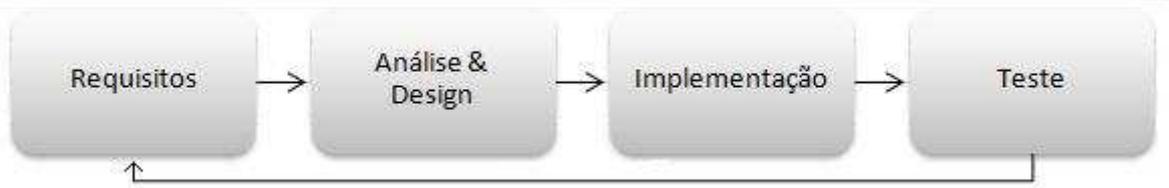


Figura 2.2 – Arquitetura geral do RUP

O desenvolvimento de um produto de software consiste em uma grande atividade que pode levar pouco ou muito tempo, dependendo do nível de complexidade e do tamanho do software a ser construído. Portanto é mais fácil dividir o trabalho em parte menores, demoninadas de iterações, e cada iteração resulta um incremento.

Segundo [18] uma iteração é uma seqüência distinta de atividades com um plano criado através de baseline e critérios de avaliação que resultam em um *release* (interno ou externo). Portanto, uma iteração de desenvolvimento é de certa forma uma passagem completa por todas as disciplinas: pelo menos Requisitos, Análise & Design, Implementação e Teste, conforme a Figura 3.2.



**Figura 2.3 – Passagem pelas disciplinas (Iteração)**

Os principais aspectos observados na abordagem iterativa e incremental é que o desenvolvimento do produto ocorre em várias iterações, conseqüentemente, várias versões do produto são geradas e os maiores riscos são priorizados. O desenvolvimento incremental é uma estratégia de planejamento estagiado ou por estágios em que partes do produto são desenvolvidas. Já o desenvolvimento iterativo é uma estratégia de planejamento de retrabalho em que o tempo de revisão e melhorias no sistema é pré-definido.

---

## Capítulo 3 O Processo Inspector

---

O Inspector é um processo de avaliação de progresso para projeto de software orientado a objetos, que visa fornecer ao gerente do projeto um conjunto diretrizes e atividades, que permitirão identificar problemas imediatos e analisar a possibilidade de problemas futuros. A área de concentração deste processo é o progresso no desenvolvimento de um determinado projeto, sendo assim, não focaliza aspectos de qualidade envolvendo o produto final. Apesar disso, o acompanhamento do progresso de um projeto indica que o gerente utiliza métodos e técnicas de gerenciamento, que geralmente resultam em produtos finais entregues na data correta e com qualidade.

O processo define um conjunto de atividades que buscam sistematizar o uso de técnicas e métricas para avaliação do progresso técnico de um projeto de desenvolvimento de software orientado a objetos. A aplicação deste processo é indicada para projetos de médio e grande porte, onde existe a necessidade de um gerenciamento mais próximo, capaz de identificar problemas e possíveis atrasos. Além disso, o processo visa garantir mecanismos que permitam fornecer relatórios, identificando o status do projeto sempre que o cliente desejar [11].

### 3.1 Conceitos

A terminologia adotada pelo Inspector é bastante similar àquela utilizada pela Rational para descrever o Processo de Desenvolvimento Unificado (RUP) [17]. O RUP representa um processo comercial, iterativo e incremental, que apresenta todas as características desejáveis para aplicação do Inspector (utiliza linguagem de modelagem UML e é dirigido através do desenvolvimento de casos de uso). Desse modo, assim como no RUP, os principais conceitos encontrados na definição do Inspector são: responsáveis, artefatos e atividades [11].

- 
- Responsáveis – Corresponde a um papel (ou posição) que pode ser atribuído a uma pessoa ou grupo, requer responsabilidade e ações, tais como realizar atividades e produzir artefatos. O papel define o comportamento e as responsabilidades de um indivíduo ou de um conjunto de indivíduos que trabalham juntos como uma equipe, no contexto de uma organização de engenharia de software [18]. O processo de avaliação de progresso para projeto de software orientado a objetos, Inspector, define três responsáveis, são eles: o Engenheiro de Processo, o Gerente de Projetos e o Coletor de Informações.
  - Artefatos – Termo geral utilizado para qualquer tipo de informação criada, produzida, alterada ou utilizada pelos responsáveis no desenvolvimento do sistema. São produtos de trabalho finais ou intermediários produzidos e usados durante os projetos. Os artefatos são usados para capturar e transmitir informações do projeto [18]. O processo de avaliação de progresso para projeto de software orientado a objetos, Inspector, classifica em três categorias distintas: artefatos produzidos (são gerados durante a avaliação do progresso do projeto), artefatos inspecionados (são os artefatos que devem ser inspecionados durante a avaliação) e artefatos de apoio (são artefatos de planejamento e gerenciamento que são produzidos durante o desenvolvimento do projeto).
  - Atividades – Uma atividade corresponde a uma unidade de trabalho tangível, coordenada por um responsável, em um fluxo de trabalho que: atribui tarefas para cada responsável, produz um resultado bem definido (conjunto de artefatos) e representa uma unidade de trabalho com limites definidos em um plano de projeto, onde as tarefas são atribuídas aos indivíduos. Uma atividade é algo que um papel faz e produz um resultado significativo no contexto do projeto. Uma atividade é uma unidade de trabalho que um indivíduo, desempenhando o papel descrito, pode ser chamado a realizar. A atividade tem uma finalidade clara, normalmente expressa em termos da criação ou atualização de alguns artefatos como um modelo, uma classe ou um plano [18]. O processo de avaliação de progresso para projeto de software orientado a objetos, Inspector, define oito macros atividades, são elas:

---

Avaliar o status da organização; Adaptar a organização; Instanciar o Inspector; Planejar a avaliação do progresso técnico; Coletar e processar dados de desempenho; Coletar e processar dados do progresso; Avaliar os resultados; e Solucionar problemas.

## 3.2 Atividades do Inspector

Esta seção apresenta uma breve descrição das atividades estabelecidas no processo Inspector. Formado por um conjunto de oito macros atividades, a Figura 3.1, apresenta o fluxo de atividades do processo Inspector, contendo as atividades e seus respectivos responsáveis, e a dependência entre as mesmas.

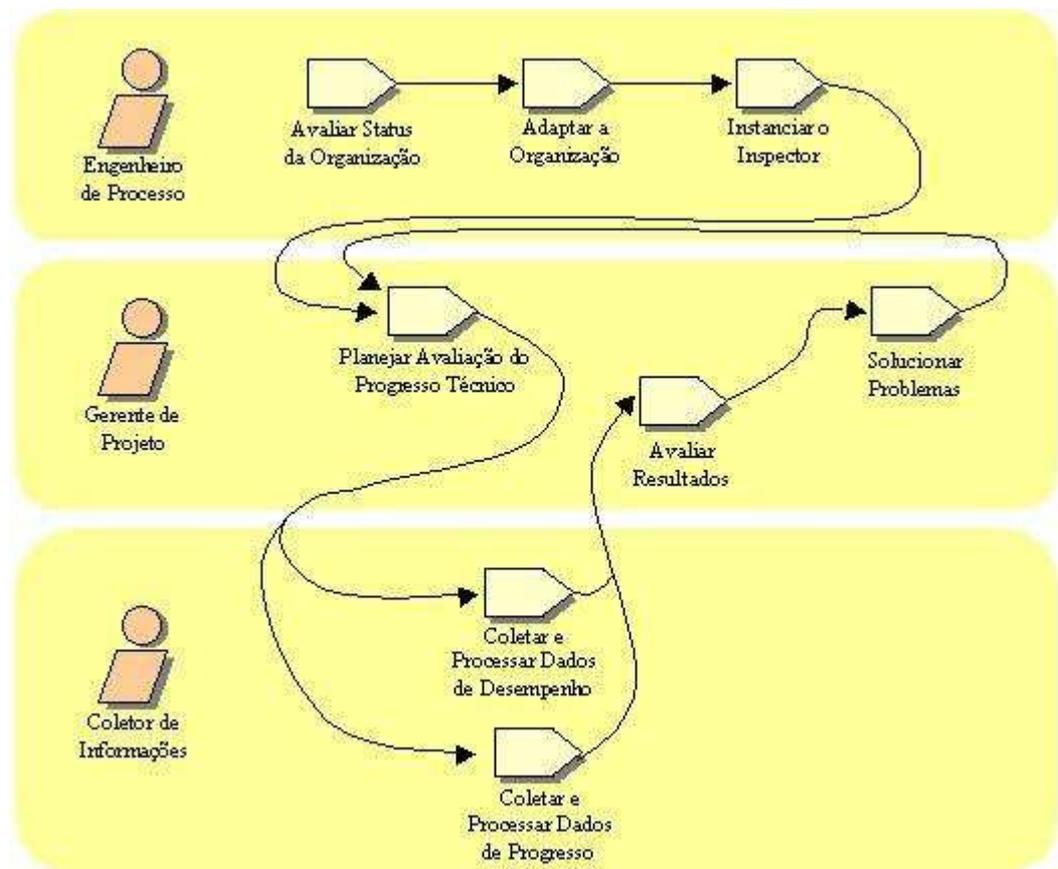


Figura 3.1 – Fluxo de atividades do processo Inspector

- Avaliar o Status da Organização – Esta atividade tem como responsável o

---

Engenheiro de Processo e o objetivo desta atividade é analisar os processos atuais de desenvolvimento da organização, identificar as ferramentas utilizadas, as habilidades e competência técnicas da equipe, o status da organização de desenvolvimento de software.

- Adaptar a Organização – No planejamento e na execução desta atividade, o Engenheiro de Processos tem como objetivo capacitar à organização na utilização das métricas estabelecidas pelo processo.
- Instanciar o Inspector – Esta atividade tem como responsável o Engenheiro de Processo e tem como objetivo adequar o processo às características e peculiaridades do projeto.
- Planejar a Avaliação do Progresso Técnico – Esta atividade é realizada de acordo com a demanda, ou seja, quando surgir à necessidade de se avaliar o progresso do projeto, tanto para fornecer uma satisfação ao cliente, quanto como forma do Gerente de Projeto acompanhar o desenvolvimento [11]. Tem como objetivo definir a data, estabelecer o escopo da avaliação e o pessoal envolvido durante a avaliação.
- Coletar e Processar Dados de Desempenho – Esta atividade tem como responsável o Coletor de Informações e tem como objetivo capturar informações de status das atividades realizadas no tempo, para posteriormente comparar com as previsões do gerente. É nesta atividade que as métricas de desempenho  $\mu_{concluído}$ ,  $\mu_{atraso\ médio}$  e  $\mu_{novas\ atividades}$  são coletadas.
- Coletar e Processar Dados do Progresso – Esta atividade tem como responsável o Coletor de Informações e consiste na atividade fundamental para uma avaliação de progresso consistente e abrangente. Preocupa-se com a funcionalidade adquirida pelo sistema desde a última avaliação. É uma atividade dirigida a casos de uso, onde para cada caso de uso do sistema se calcula o progresso que o mesmo teve em relação ao tempo. É nesta atividade que as métricas de progresso  $\mu_{caso\ de\ uso}$  e

---

$\mu_{sistema}$  são coletadas.

- Avaliar dos Resultados – Esta atividade tem como responsável o Gerente de Projetos e tem como objetivo a análise das visões de desempenho e de funcionalidade, resultando na identificação da necessidade de alteração no cronograma, identificação de casos de uso que estão com problemas, e das equipes com maiores dificuldades para realização das atividades planejadas no tempo previsto.
- Solucionar Problemas – Esta atividade tem como responsável o Gerente de Projetos e tem como objetivo identificar os reais motivos na execução das atividades, bem como soluções para os problemas encontrados.

### **3.3 Visões do Inspector**

O Inspector fornece ao gerente de projeto duas visões de progresso complementares que fornecem suporte para um bom gerenciamento do processo de desenvolvimento. São elas: visão de desempenho e visão de funcionalidade.

A primeira visão verifica o desempenho das equipes de desenvolvimento a partir da análise das atividades planejadas para as mesmas, definindo três métricas que mostram a qualidade do planejamento e a produtividade da equipe de desenvolvimento. Já a visão de funcionalidade define uma métrica, dirigida a casos de uso, que verifica o progresso funcional do sistema. De posse das duas visões, o gerente de projeto é capaz de identificar equipes com dificuldades e casos de uso com problemas no desenvolvimento [11].

#### **Visão de Desempenho**

A principal preocupação dessa visão é fornecer informações suficientes ao gerente de projetos para permitir que o mesmo encontre possíveis atrasos no cronograma e qual a influência deste atraso no restante do projeto. Além disso, essa visão preocupa-se em mostrar quantitativamente problemas de planejamento das atividades das diversas equipes, e o quanto se foi concluído das atividades realmente planejadas. A partir das informações desta visão, pode-se verificar a necessidade de inserir, realocar ou retirar recursos do

---

projeto.

O Inspector define três métricas que auxiliam na avaliação de desempenho as equipes de desenvolvimento. São elas:  $\mu_{concluído}$ ,  $\mu_{atraso\ médio}$  e  $\mu_{novas\ atividades}$ .

- $\mu_{concluído}$  – esta métrica representa o quanto uma determinada equipe já concluiu das atividades que estavam inicialmente planejadas para terminarem dentro do escopo de tempo da avaliação. Esta métrica é calculada através da seguinte equação:

$$\mu_{concluído}(equipe) = \frac{\sum_{x \in Atividades(equipe)} tempoEstimado(x) \times concluído(x)}{\sum_{x \in Atividades(equipe)} tempoEstimado(x)}$$

onde *equipe* indica a equipe que está sendo avaliada, *Atividades(equipe)* representa o conjunto de atividades relacionadas com *equipe* que estavam inicialmente planejadas para serem concluídas no escopo de tempo da avaliação, *concluído(x)* consiste na função que retorna o quanto da atividade *x* se realizou, com  $0 \leq concluído(x) \leq 1$ , e *tempoEstimado(x)* é a função que retorna o tempo estimado para a atividade *x*, ou seja, é a função que indica o quanto se espera que a atividade *x* demore para ser realizada [11].

- $\mu_{atraso\ médio}$  – esta métrica indica, quantitativamente, o percentual de atraso médio por atividade que está sendo enfrentado por uma determinada equipe de desenvolvimento. Esta métrica é calculada através da seguinte equação:

$$\mu_{atraso\ médio}(equipe) = \frac{\sum_{x \in Atividades(equipe)} \frac{tempoEstimado_f(x) - tempoEstimado_i(x)}{tempoEstimado_i(x)}}{\# Atividades(equipe)}$$

onde *equipe* indica a equipe que está sendo avaliada, *Atividades(equipe)* representa o conjunto de atividades relacionadas com *equipe*, que estavam inicialmente planejadas para serem concluídas no escopo de tempo da avaliação, *tempoEstimado<sub>i</sub>(x)* é a função que retorna o tempo inicialmente estimado para a

---

atividade  $x$ ,  $tempoEstimado_f(x)$  é a função que retorna o tempo gasto para realização da atividade  $x$ , caso ela tenha sido completada, ou, caso contrário, retorna a estimativa do novo tempo necessário para realização desta atividade, e  $\# Atividades(equipe)$  é a cardinalidade de  $Atividades(equipe)$  [11].

- $\mu_{novas\ atividades}$  – esta métrica tem como objetivo indicar a porcentagem média de tempo gasto com novas atividades que surgem dentro do escopo da avaliação. Esta métrica é calculada através da seguinte equação:

$$\mu_{novas\ atividades} = \frac{\sum_{x \in Atividades\ Iniciadas} tempoGasto(x) - \sum_{x \in (AtividadesIniciadas \cap AtividadesPlanejadas)} tempoGasto(x)}{\sum_{x \in AtividadesIniciadas} tempoGasto(x)}$$

onde  $AtividadesIniciadas$  é o conjunto que contém as atividades realmente iniciadas dentro do escopo da avaliação,  $AtividadesPlanejadas$  é o conjunto que contém as atividades planejadas para serem iniciadas dentro desse escopo, e  $tempoGasto(x)$  é a função que retorna a quantidade de tempo já trabalhado para realização da atividade  $x$ .

### Visão de Funcionalidade

A visão de funcionalidade visa superar as limitações existentes na visão de desempenho, e tem como objetivo fundamental indicar o aumento percentual (%) da funcionalidade do sistema como um todo, a partir da análise de cada caso de uso do projeto individualmente. A métrica  $\mu_{sistema}$  representa a métrica global que indica quantitativamente o aspecto funcional já inserido, ou parcialmente inserido, dentro do sistema.

A métrica  $\mu_{sistema}$  possui as seguintes propriedades:

- Conceito de progresso funcional – representa o quanto das funcionalidades previstas para o projeto foram implementadas.

- 
- Dirigida a casos de uso – o cálculo da métrica deriva diretamente do progresso funcional de cada caso de uso do sistema. É feita uma análise do progresso funcional de cada caso de uso, onde são considerados os artefatos necessários para desenvolver o caso de uso e, a partir dessas análises individuais, é calculado o progresso funcional de todo o sistema.
  - Baseada na inspeção de artefatos – a métrica considera a produção e/ou atualização de determinados artefatos, como sendo a forma de se alcançar o progresso no desenvolvimento de cada caso de uso.
  - Facilidade de observação – a métrica definida apresenta como resultado final um valor único, que indica o progresso funcional do sistema (%), facilitando assim a comparação com resultados obtidos anteriormente.

O progresso funcional do projeto deriva diretamente do progresso funcional dos diversos casos de uso relacionados ao sistema [11]. Esta métrica é calculada através da seguinte equação:

$$\mu_{sistema} = \sum_{casodeuso \in DCU} \mu_{casodeuso}$$

onde  $\mu_{casodeuso}$  indica o progresso de um caso de uso e DCU representa o conjunto do(s) Diagrama(s) de Caso de Uso para o sistema em questão.

Inicialmente, na definição dessa métrica, observou-se, que ela considerava todos os casos de uso tendo a mesma participação no progresso do sistema. O que se observa na prática é que, casos de uso, frequentemente, apresentam complexidade diferente e necessitam de níveis de esforço variados para sua realização [11]. Neste contexto, o Inspector, define a seguinte equação:

$$\mu_{sistema} = \frac{\sum_{casodeuso \in DCU} p_{casodeuso} \times \mu_{casodeuso}}{\sum_{casodeuso \in DCU} p_{casodeuso}}$$

---

onde  $p_{casodeuso}$  indica o peso do caso de uso para o progresso do sistema,  $\mu_{casodeuso}$  indica o progresso do caso de uso, e DCU representa o conjunto do(s) Diagrama(s) de Caso de Uso do sistema.

Posteriormente surgiu a necessidade de considerar a possibilidade de avaliação de progresso em um processo de desenvolvimento iterativo. Para atender a esta necessidade o Inspector define a seguinte equação:

$$\mu_{sistema}(i) = \frac{\sum_{casodeuso \in DCU} p_{casodeuso}(i) \times \mu_{casodeuso}(i)}{\sum_{casodeuso \in DCU} p_{casodeuso}(i)} \quad \text{com } 1 \leq i \leq n$$

onde  $i$  indica a iteração na qual o projeto se encontra no momento da avaliação,  $n$  representa o número de iterações planejadas,  $p_{casodeuso}$  indica o peso do caso de uso para o progresso do sistema,  $\mu_{casodeuso}$  indica o progresso do caso de uso, e DCU representa o conjunto do(s) Diagrama(s) de Caso de Uso do sistema.

Como  $\mu_{sistema}$  deriva diretamente dos progressos dos casos de uso, e estes são calculados a partir da realização das grandes etapas necessárias para seu desenvolvimento, pode-se considerar o progresso de cada uma dessas etapas, ao sistema como um todo [11]. O processo Inspector definiu a seguinte variação:

$$\mu_{sistema}(i, j) = \frac{\sum_{casodeuso \in DCU} p_{casodeuso}(i, j) \times \mu_{casodeuso}(i, j)}{\sum_{casodeuso \in DCU} p_{casodeuso}(i, j)}$$

onde  $i$  indica a iteração que o projeto se encontra no momento da avaliação,  $j$  representa a etapa de desenvolvimento cujo progresso está sendo avaliado,  $p_{casodeuso}(i, j)$  indica o peso relativo ao desenvolvimento da etapa  $j$  do caso de uso na iteração  $i$ , e  $\mu_{casodeuso}(i, j)$  indica o progresso da etapa de desenvolvimento  $j$  de cada caso de uso, na iteração  $i$ .

A partir da definição da métrica global de avaliação de progresso, dirigida a casos

---

de uso, verificou-se a necessidade de se definir  $\mu_{casodeuso}(i)$ , ou seja, o progresso relativo a um caso de uso. A métrica relacionada ao progresso de um caso de uso considera as principais etapas necessárias para o seu desenvolvimento, partindo desde a identificação e documentação inicial do caso de uso até seu desenvolvimento e integração [11]. Neste contexto o Inspector define a seguinte equação:

$$\mu_{casodeuso}(i) = \frac{\sum_{j \in ETAPAS} p_{casodeuso}(i, j) \times \mu_{casodeuso}(i, j)}{\sum_{j \in ETAPAS} p_{casodeuso}(i, j)}$$

onde  $i$  indica a iteração atual,  $j$  representa a etapa de desenvolvimento do caso de uso avaliada,  $p_{casodeuso}(i, j)$  indica o peso relativo ao caso de uso na iteração  $i$  e etapa de desenvolvimento  $j$ ,  $\mu_{casodeuso}(i, j)$  indica o progresso do caso de uso na iteração  $i$  e etapa de desenvolvimento  $j$ .

O progresso de um caso de uso em uma determinada etapa ( $\mu_{casodeuso}(i, j)$ ), é calculado a partir da contribuição dos diversos artefatos necessários para que esta etapa seja realizada. Deste modo temos:

$$\mu_{casodeuso}(i, j) = \frac{\sum_{x \in A_{ij}} p_x(casodeuso) \times \mu_x(casodeuso)}{\sum_{x \in A_{ij}} p_x(casodeuso)}$$

onde  $p_x(casodeuso)$  indica o peso do artefato  $x$  para construção de  $casodeuso$ ,  $\mu_x(casodeuso)$  representa a métrica que indica percentualmente se o artefato  $x$  já foi produzido ou alterado de modo a realizar a etapa  $j$  no desenvolvimento do caso de uso, e  $A_{ij}$  representa o conjunto de Artefatos que devem ser produzidos ou atualizados na iteração  $i$  e etapa  $j$ .

---

## Capítulo 4 Inspector Panel

---

O Inspector Panel é uma ferramenta de controle para acompanhamento e avaliação do progresso de projetos de software orientados a objetos que dá apoio a automação do processo Inspector. Neste contexto, foi acrescentado o acompanhamento do cronograma e dos custos do projeto de software.

Essa ferramenta pode ser utilizada por pequenas, média e grandes empresas de desenvolvimento de software orientado a objetos, baseando-se na metodologia RUP.

### 4.1 Objetivos da Ferramenta

A proposta inicial da ferramenta Inspector Panel é automatizar algumas atividades do processo Inspector, descrito no Capítulo 3 desta monografia. Atualmente a ferramenta cobre/contempla as seguintes atividades: "Instanciar o Processo", "Coletar e Processar Dados de Desempenho", "Coletar e Processar Dados do Progresso Funcional", "Avaliação dos Resultados", definidas no processo de avaliação de progresso de projetos de software, o Inspector.

Porém destacam-se outros objetivos, como o acompanhamento das atividades (cronograma) e dos custos do projeto. Portanto, a relevância e originalidade deste trabalho estão no fato de que atualmente não existem ferramentas que automatizem as atividades do processo de avaliação de progresso de software contempladas pelo Inspector Panel. Assim, os principais objetivos alcançados pela ferramenta Inspector Panel são:

- Automatização do processo e redução de falha na coleta. Ou seja, com a ferramenta é possível a realização/automatização das atividades do processo Inspector.
- Rapidez e eficiência na coleta e no processamento de dados. Depois que os dados foram levantados e devidamente registrados na ferramenta, já se pode realizar uma inspeção a fim de verificar o status do projeto. Nesta avaliação, a ferramenta

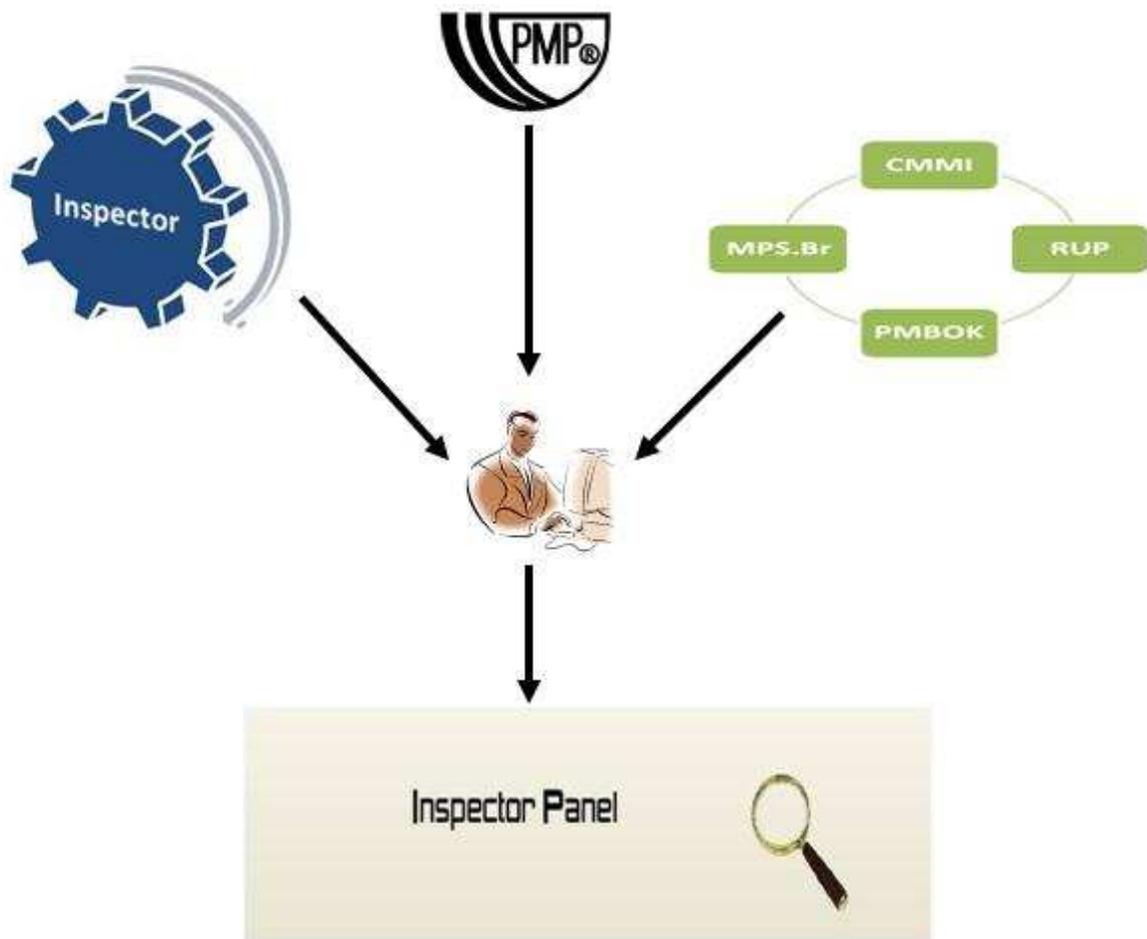
---

Inspector Panel, realiza todo o processo de inspeção definido no processo Inspector, abordado no Capítulo 3. Iniciado o processo de inspeção, a ferramenta calcula as métricas de desempenho e de funcionalidade do projeto.

- Estabelece um status de progresso do projeto a partir das métricas.
- Oferece suporte à tomada de decisão. Ou seja, a ferramenta pode ser utilizada a fim de conceber fatos relevantes para a gerência de projeto. A partir daí, a gerência têm dados suficientes e saberá tomar uma decisão mais precisa em relação ao projeto de software.
- Internacionalização. Atualmente, está disponível em português e em inglês, mas pode ser facilmente disponibilizado para outros idiomas.
- Geração de indicadores visuais. A ferramenta disponibiliza os seguintes gráficos: o gráfico de custos por projeto, de custos por fases do projeto, de custos por iterações, e de custos por casos de uso – onde se pode facilmente comparar o custo estimado e o real; o gráfico de *gantt* (cronograma) estimado e o real – onde se pode visualizar o quanto das atividades foram concluídas e o quanto falta para concluir, além de facilitar a identificação das atividades que estão em atraso; e por fim, o gráfico de progresso do projeto e o gráfico de progresso por casos de uso do projeto – indicam em porcentagem o progresso alcançado durante as inspeções realizadas.

## 4.2 Requisitos da Ferramenta

Os requisitos da ferramenta vieram de três fontes diferentes (Figura 4.1 – Obtenção dos requisitos para ferramenta Inspector Panel). A primeira fonte de informações foi o processo Inspector, a segunda foi obtida através de conversas com um Gerente de Projetos que possui certificado PMP, orientador desta monografia e por fim complementado as informações através de estudos realizados nas seguintes abordagens: na disciplina de Gerência de Projetos, do RUP; no grupo de processo, Monitoramento e Controle, do PMBOK; na área de processo, Acompanhamento de Projeto, do modelo CMMI; e no processo, Gerência de Projetos, do modelo de referência MPS.Br.



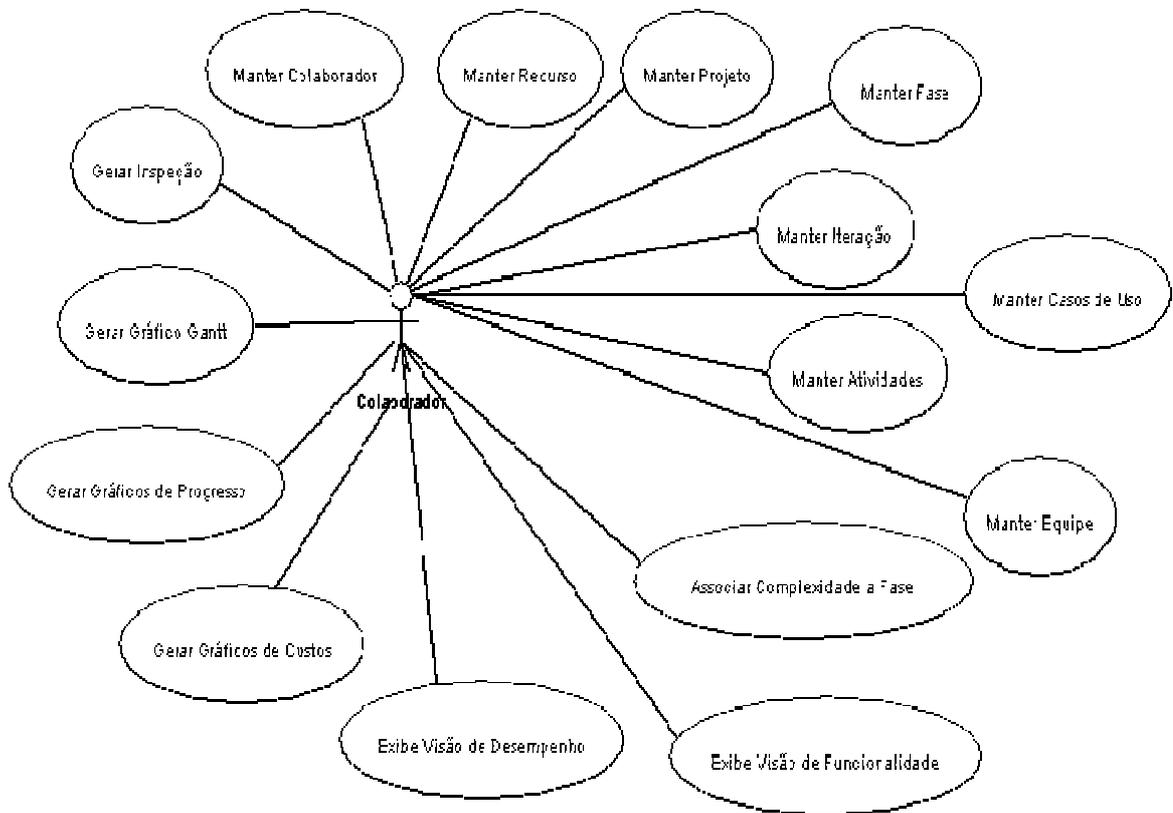
**Figura 4.1 – Obtenção dos requisitos para ferramenta Inspector Panel**

Durante todo o ciclo de vida do Inspector Panel, sempre houve a preocupação de planejar as fases, as iterações, o cronograma e as atividades a serem desenvolvidas ao longo do projeto. Neste contexto, criou-se uma *homepage* para o devido acompanhamento das atividades do projeto, bem como a publicação dos artefatos produzidos.

A Figura 4.2 – Diagrama de casos de uso do Inspector Panel representa de uma forma mais macro, o modelo de casos de uso da ferramenta Inspector Panel. O modelo de caso de uso apresenta apenas uma visão estática de todas as funcionalidades da ferramenta, sem se preocupar com a ordem que as atividades acontecem. O modelo de caso de uso trata-se de um dos diagramas da *Unified Modeling Language* (UML) amplamente utilizada ao longo do ciclo de vida de software [17]. A UML é uma linguagem de especificação,

---

modelagem e documentação de sistemas e seus diagramas permitem a visualização de sistemas de software sob diferentes perspectivas.



**Figura 4.2 – Diagrama de casos de uso do Inspector Panel**

Para cada caso de uso (Manter) apresentado, foram implementados as funcionalidades básicas de consulta, inserção, alteração e de exclusão (lógica) de dados. Encontram-se fora do escopo do projeto os níveis de permissões do usuário as funcionalidades do sistema. De acordo com a Figura 4.2, podemos observar a existência do ator Colaborador que pode ser o Gerente de Projetos, o Coletor de Informações ou o Engenheiro de Processos.

Além dos conceitos já abordados pelo processo Inspector, foram acrescentados outros conceitos, são eles:

- Colaborador – o colaborador é todo e qualquer *stakeholders* responsável em desempenhar determinadas atividades durante o ciclo de vida do software,

---

permitem também o acesso às funcionalidades da ferramenta Inspector Panel, após autenticação do mesmo.

- Recurso – abrange tudo que for necessário para a realização das atividades do projeto, podem ser equipamentos e até o próprio colaborador, principal responsável na execução das atividades do projeto.

Estes novos conceitos servem como base para atender as necessidades do Cronograma e Custos do projeto, inicialmente proposto no início desta monografia.

### **4.3 Funcionalidades**

Esta seção descreve todas as funcionalidades do Inspector Panel implementadas neste trabalho.

4.3.1. Manter Projeto – esta funcionalidade é responsável por manter os dados relativos ao projeto, por exemplo, o responsável (Gerente) pelo projeto, nome do projeto e status do projeto (em andamento, concluído, cancelado).

4.3.2. Manter Fase – esta funcionalidade é responsável por manter os dados relativos às fases estabelecidas para o projeto, por exemplo, o nome da fase do projeto.

4.3.3. Manter Iteração – esta funcionalidade é responsável por manter os dados relativos às iterações estabelecidas para uma fase do projeto, por exemplo, o nome da iteração.

4.3.4. Manter Casos de Uso – esta funcionalidade é responsável por manter os dados relativos aos casos de uso estabelecidos para o projeto, por exemplo, o nome do caso de uso, descrição do caso de uso e a complexidade do caso de uso que varia de zero a noventa e nove, assim cada empresa pode estabelecer seu próprio critério de complexidade.

4.3.5. Manter Atividades – esta funcionalidade é responsável por manter os dados relativos às atividades necessárias para o desenvolvimento de um caso de uso,

---

por exemplo, o nome da atividade, descrição da atividade e complexidade da atividade (varia de zero a noventa e nove), data de início e de fim (estimada e real), quantidade de horas de trabalho (estimada e real) por atividade, equipe responsável pela execução das atividades, recursos necessários e identificação da iteração onde a atividade deve ser desenvolvida.

4.3.6. Manter Equipe – esta funcionalidade é responsável por manter os dados relativos às equipes estabelecidas o projeto, por exemplo, o nome da equipe.

4.3.7. Manter Recurso – esta funcionalidade é responsável por manter os dados relativos aos recursos necessários para o projeto, por exemplo, nome do recurso.

4.3.8. Manter Colaborador – esta funcionalidade é responsável por manter os dados relativos aos colaboradores do projeto, por exemplo, o nome do colaborador, email do colaborador, CPF, cargo/função que desempenha no projeto e permite o acesso ao sistema, após autenticação.

4.3.9. Associar Complexidade a Fase – esta funcionalidade é responsável por manter os dados relativos da complexidade do caso de uso em cada fase do projeto. Cada fase estabelecida possui um grau de complexidade diferente na realização das atividades de um determinado caso de uso.

4.3.10. Gerar Inspeção – esta funcionalidade é responsável em gerar inspeções no projeto, ou seja, coleta e armazena dados necessários para gerar as visões de funcionalidade e de desempenho estabelecidas pelo processo Inspector.

4.3.11. Exibir Visão de Funcionalidade – esta funcionalidade exibe a visão de funcionalidade estabelecida pelo processo Inspector. As métricas  $\mu_{sistema}$  e  $\mu_{casodeuso}$  são apresentadas para cada inspeção realizada no projeto.

4.3.12. Exibir Visão de Desempenho – esta funcionalidade exibe a visão de desempenho estabelecida pelo processo Inspector. As métricas  $\mu_{concluido}$ ,  $\mu_{atraso\ médio}$  e  $\mu_{novas\ atividades}$  são apresentadas para cada equipe do projeto.

---

4.3.13. Gerar Gráfico de *Gantt* – esta funcionalidade visa à geração do gráfico de *gant*, também conhecido como cronograma. Um *Gantt* de atividades consiste em um gráfico que indica as atividades necessárias para realização do projeto, indicando as datas de início e fim, a equipe responsável pela realização de cada atividade, entre outras informações.

4.3.14. Gerar Gráficos de Custos – esta funcionalidade visa à geração de gráficos de barras, onde é possível comparar os custos estimados com real. Atualmente, a ferramenta disponibiliza quatro gráficos diferentes para esta função, são eles: o gráfico de Custo por Projeto, o de Custo por Fase, o de Custo por Iteração e o de Custo por Caso de Uso. Com estes gráficos é possível identificar o caso de uso, ou a iteração, ou a fase está com problemas.

4.3.15. Gerar Gráficos de Progresso – esta funcionalidade visa à geração de gráficos de linhas, possibilitando visualizar o progresso atingido durante as inspeções. Atualmente, a ferramenta disponibiliza dois gráficos diferentes para esta função, são eles: o gráfico de Progresso por Projeto e o de Progresso por Casos de Uso.

## 4.4 Implementação

Esta seção tem como objetivo apresentar a arquitetura definida na construção do Inspector Panel, o ambiente e ferramentas utilizadas durante o desenvolvimento realizado conforme o cronograma do projeto disponível no apêndice A deste trabalho.

### Ambiente do Inspector Panel

Para o ambiente do Inspector Panel foi definido e utilizado o Eclipse [19] adotando a linguagem Java [24], seguindo cuidadosamente as técnicas de orientação a objetos, objetivando de criar um produto com maior extensibilidade, reusabilidade e flexibilidade. Para apoiar o projeto foram adotados os *frameworks* Spring [26] e Hibernate [25]. O Spring é um *framework* baseado em programação orientada a aspectos e container IOC (*Inversion of Control*), enquanto que o Hibernate é um *framework* de persistência de

---

dados.

Durante a construção do Inspector Panel utilizou nos ambientes de desenvolvimento, teste, homologação e produção o banco de dados *PostgreSQL* [27] para fazer o armazenamento dos dados.

### **Arquitetura do Inspector Panel**

Para a arquitetura, foram considerados fatores como finalidade do sistema, tipo de usuários e ambiente de execução. Sendo assim, a arquitetura a ser adotada precisaria atender às seguintes características:

- **Modularidade:** Faz com que o sistema possua partes não acopladas, facilitando a possível substituição de componentes do mesmo. Essa característica se mostra muito importante, por se tratar de um aplicativo voltado para a Web, onde frequentemente surgem novas tecnologias.
- **Manutenibilidade:** O sistema deve estar no ar 24 horas por dia, o que exige uma flexibilidade a permitir a manutenção fácil e rápida do mesmo.
- **Extensibilidade:** Como já foi mencionado, possivelmente serão necessárias atualizações e extensões do software. Portanto, sua arquitetura deve ser projetada com o objetivo de facilitar esta tarefa.
- **Reusabilidade:** O reuso é considerada hoje como um dos mais importantes fatores durante o desenvolvimento de um sistema. Com isto é possível a utilização de classes e componentes em outros projetos, favorecendo o tempo de produção e a qualidade do produto gerado.

A representação da arquitetura é apresentada como modelos do JUDE UML Modeling Tool [23] e utiliza a linguagem unificada de modelagem (UML – *Unified Modeling Language*) [17]. A arquitetura adotada no sistema é disposta em camadas independentes, e são apresentadas na Figura 4.3 – Arquitetura do Inspector Panel:



**Figura 4.3 – Arquitetura do Inspector Panel**

Onde: a camada de Interface Gráfica é composta pelos componentes de apresentação da aplicação, responsáveis pela interação do usuário com o sistema; a camada de Comunicação implementa os mecanismos de comunicação entre a GUI e a camada de negócio. Isto torna o sistema independente da interface da qual ele faz uso no momento, facilitando a extensibilidade; a camada de Negócio implementa os serviços e as regras de negócio da aplicação; e por fim, a camada de Dados é responsável pelo armazenamento e recuperação dos dados do sistema.

---

## Capítulo 5 Estudo de Caso

---

Ao desenvolver uma ferramenta é necessário verificar se a mesma atende aos requisitos especificados. Para validar é necessário analisá-la do ponto de vista do usuário: o engenheiro de processos, o gerente de projetos e do coletor de métricas, principais atores envolvidos. Este capítulo tem como objetivo capturar *feedback* dos usuários, identificar pontos positivos, pontos de melhoria e pontos negativos, mostrar os resultados obtidos durante a aplicação do Inspector Panel a um projeto real de desenvolvimento de software.

Durante o estudo de caso realizou-se o acompanhamento e monitoramento de um projeto real de desenvolvimento de software orientado a objetos. Devido ao fato da empresa e o projeto serem reais, houve algumas limitações impostas na realização deste estudo de caso. Uma das limitações estabelecidas pela diretoria da empresa foi a não publicação do nome da empresa, do projeto e do cliente. Outra limitação imposta refere-se a não disponibilização de informações dos custos do projeto. Para atender as exigências impostas pela empresa, retirou-se do escopo do estudo de caso o acompanhamento dos custos do projeto e denominou-se a empresa como empresa xE e o projeto como projeto xP.

A empresa xE atua a três anos na área de tecnologia e possui cerca de dezoito funcionários, dos quais três exercem as funções na área administrativa/financeira e na área comercial, os demais estão distribuídos na área de design e programação. O projeto xP é desenvolvido pela empresa xE de desenvolvimento de software e a equipe é composta por um gerente de projetos e quatro analistas/desenvolvedores. O gerente de projetos também é responsável pelos outros projetos da empresa (xE). Apesar do pouco tempo de atuação no mercado, o xP atendeu os requisitos básicos para utilização do processo Inspector, conseqüentemente o Inspector Panel, ou seja, utiliza a UML como linguagem de modelagem de dados, processo de desenvolvimento orientado a objetos e as funcionalidades são baseadas em casos de uso.

---

As seções seguintes descrevem com maiores detalhes a aplicação do Inspector Panel ao projeto xP, apresentando as atividades realizadas durante o acompanhamento deste projeto.

## **5.1 Atividades Realizadas**

### **Avaliando o Status das Métricas na Organização**

De acordo com o processo de avaliação de progresso de projetos de software, o Inspector, a primeira atividade a ser realizada é Avaliar o Status das Métricas na Organização, sob responsabilidade do Engenheiro de Processos. Esta atividade tem como objetivo obter uma visão geral da organização. Para obter esta visão foi realizada uma análise dos processos atuais de desenvolvimento da organização, identificando as ferramentas utilizadas, mapeando as habilidades e competência técnicas da equipe.

Observou-se que a organização como todo, direção e colaboradores, estão preocupados em seguir uma metodologia formal de desenvolvimento de software. As equipes de análise e de desenvolvimento estão capacitadas a desenvolver suas atividades, devido ao incentivo dado pela organização para os cursos de aperfeiçoamento e também pela experiência adquirida em projetos anteriores.

Apesar do pouco tempo de atuação no mercado, a organização estabeleceu uma pequena instância do RUP que atenda as necessidades do projeto, utilizando o paradigma de orientação a objetos. O ambiente de desenvolvimento da organização utiliza o Eclipse [19], como ferramenta de desenvolvimento de software orientado a objetos, ArgoUML [20], como ferramenta de modelagem, o GanttProject [21] como ferramenta de gerenciamento e acompanhamento das atividades relacionadas ao projeto e o Bugzilla [22] para reportar os erros e defeitos encontrados na etapa de testes.

A organização ainda não estabeleceu nenhuma métrica formal, na prática existe e é possível identificar as atividades que estão em atrasos, a partir do cronograma que é atualizado semanal. Porém, nenhuma informação é salva, sabe-se que é necessária a alocação de mais recursos para finalizar a atividade em atraso. Outro problema enfrentado

---

pela organização é a falta de comunicação entre o departamento de vendas (comercial) e os gerentes de projetos. O comercial não sabe das dificuldades enfrentadas pela equipe e não estão atentos aos prazos de entrega dos projetos, e muitas vezes prometem aos clientes prazos impossíveis de serem cumpridos pela equipe de desenvolvimento.

A princípio a implantação de um novo processo, desconhecido, na organização gerou um grande desconforto para o gerente de projetos, pois se tratava de mais uma tarefa/obrigação a ser cumprida. Posteriormente, o mesmo desconforto atingiu a equipe de análise e de desenvolvimento do projeto no que diz respeito, principalmente, à avaliação de desempenho.

Para evitar problemas de aceitação, foi essencial destacar a importância no uso de métricas na organização, bem como esclarecer que a avaliação de desempenho não prejudicaria a equipe.

### **Adaptando a Organização**

A próxima atividade a ser realizada é Adaptar a Organização. No planejamento e na execução desta atividade, o Engenheiro de Processos tem como objetivo capacitar à organização na utilização das métricas estabelecidas pelo processo Inspector. Através de treinamento foi possível capacitar as pessoas envolvidas no projeto, bem como esclarecer todas as dúvidas sobre o processo Inspector, enfatizando as métricas estabelecidas e o uso da ferramenta desenvolvida, o Inspector Panel.

---

Esta atividade compreende a definição de alguns critérios, por exemplo: estabelecer, definir o nível de complexidade (peso) para as atividades. A definição da complexidade na realização das tarefas era feita apenas no planejamento das atividades durante a criação do cronograma, e não levava em consideração a fase do projeto. A equipe do projeto (gerente, equipe de desenvolvimento e de análise) definiu pesos para os casos de uso e etapas, conforme a Tabela 5.1 – Definição dos pesos do projeto.

**Tabela 5.1 – Definição dos pesos do projeto**

<b>Classificação da complexidade (UC/Etapa)</b>	<b>Peso</b>
<b>Simple</b>	7
<b>Média</b>	10
<b>Alta</b>	14

### **Instanciando Inspector**

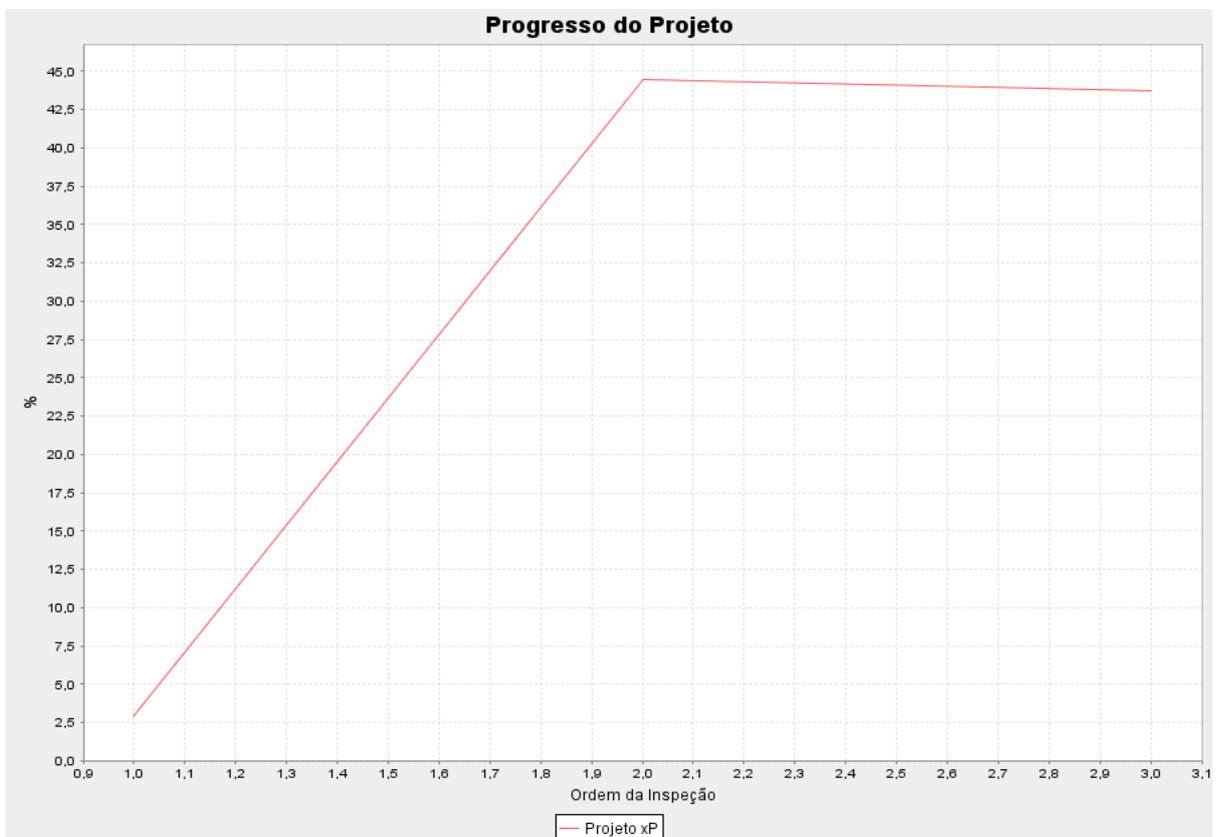
Esta atividade tem como responsável o Engenheiro de Processo e tem como objetivo adequar o processo as características e peculiaridades do projeto. Durante a realização desta atividade definiu as *ETAPAS* necessárias para o desenvolvimento dos casos de uso e os artefatos a serem produzidos durante o ciclo de vida de software. A equipe do projeto xP definiu as seguintes *ETAPAS*: Especificação Inicial, Análise e Projeto, Implementação e, por fim, Teste.

### **Avaliando o Progresso e Desempenho das Equipes**

Durante a aplicação do Inspector Panel ao um projeto real de desenvolvimento de software, foram realizadas três inspeções que tem como objetivo realizar o acompanhamento de progresso do projeto. A partir das inspeções realizadas foi possível obter as visões de desempenho e de progresso, e o acompanhamento das atividades.

---

A Figura 5.1 – Progresso do projeto xP, apresenta a variação de progresso funcional encontrado durante as três inspeções realizadas no projeto xP. A primeira inspeção, realizada no dia 15 de agosto de 2008, obteve um progresso de apenas  $\mu_{sistema} = 2,91$ . Já na segunda inspeção, realizada no dia 19 de setembro de 2008, o resultado obtido foi de  $\mu_{sistema} = 44,50$ . A terceira inspeção, realizada no dia 10 de outubro de 2008, o progresso obtido foi  $\mu_{sistema} = 43,71$ .



**Figura 5.1 – Progresso do projeto xP**

Além do acompanhamento de progresso do projeto xP, foi monitorado também o progresso de cada caso de uso relacionado ao projeto, conforme apresentado na Figura 5.2 – Progresso dos casos de uso do projeto xP. A Figura 5.2 apresenta o status do progresso dos casos de uso do projeto durante as inspeções realizadas. A princípio foram identificados treze casos de uso, posteriormente, a partir da segunda inspeção foi

identificada a existência de um novo caso de uso, totalizando um total de quatorze casos de uso necessários para a construção do projeto xP.

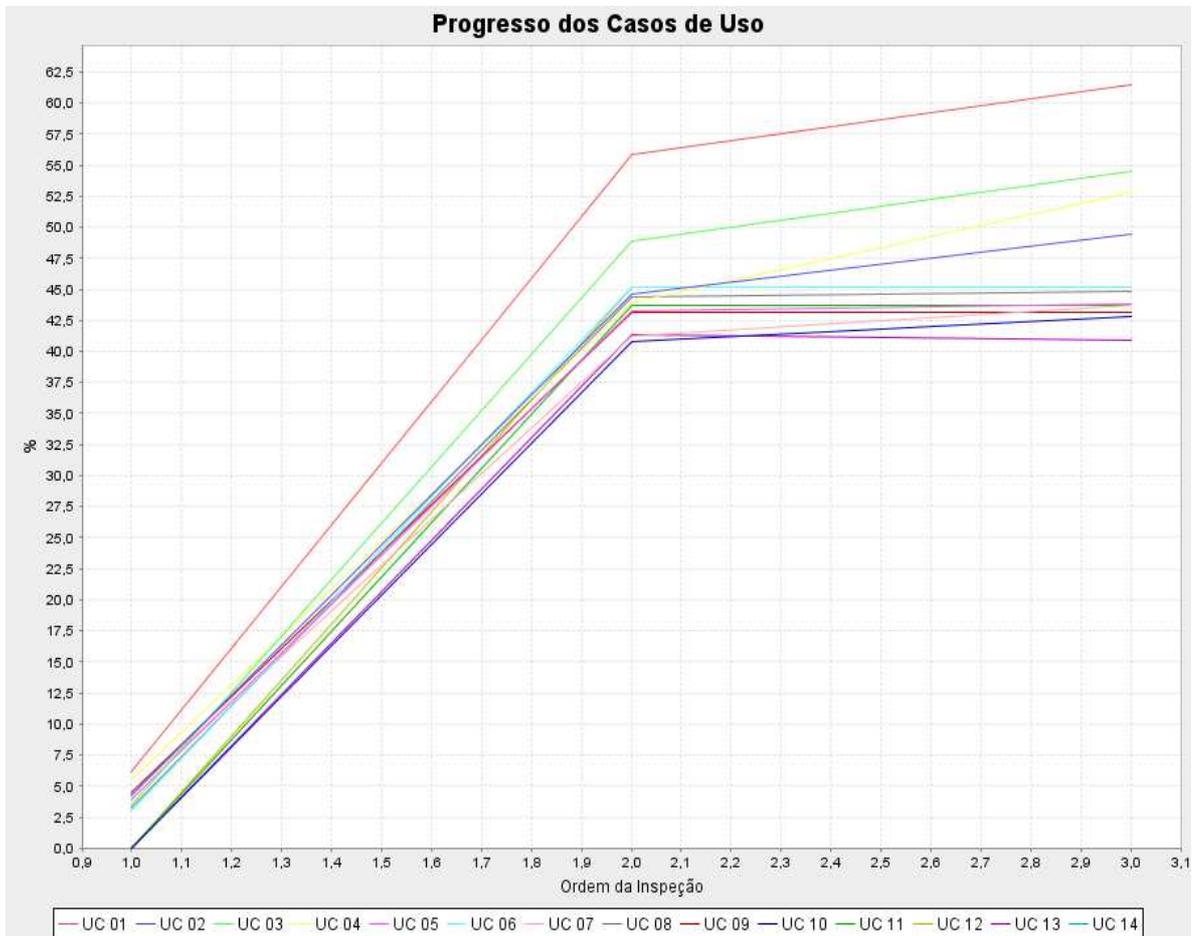


Figura 5.2 – Progresso dos casos de uso do projeto xP

### Primeira Inspeção

O gerente de projetos estabeleceu, de maneira *ah-doc*, um objetivo de atingir inicialmente um progresso superior a 20,00%. No dia 15 de agosto de 2008, quando foi realizada a primeira inspeção após o levantamento da maioria das atividades relativas ao desenvolvimento dos casos de uso do projeto e foi considerado/calculado o quanto, em porcentagem (%), das atividades estavam em andamento, concluídas ou ainda não

iniciadas. O progresso do projeto encontrado foi de apenas  $\mu_{sistema} = 2,91$ , ou seja, apenas 2,91% das funcionalidades tinham sido incorporadas ao sistema.

Nessa inspeção observou-se a existência de treze casos de uso, podendo analisar o progresso funcional encontrado para cada caso de uso e para o projeto, conforme a Tabela 5.2 – Visão de funcionalidade obtida na primeira inspeção.

**Tabela 5.2 – Visão de funcionalidade obtida na primeira inspeção**

<b>Casos de Uso</b>	<b>Especificação Inicial</b>	<b>Análise e Projeto</b>	<b>Implementação</b>	<b>Teste</b>	<b>Progresso Total</b>
<b>UC 01 (Peso:7)</b>	15,00%	12,50%	0,00%	0,00%	6,21%
<b>UC 02 (Peso:7)</b>	8,75%	10,00%	0,00%	0,00%	4,23%
<b>UC 03 (Peso:7)</b>	12,75%	2,50%	0,00%	0,00%	3,44%
<b>UC 04 (Peso:7)</b>	13,25%	11,00%	0,00%	0,00%	5,48%
<b>UC 05 (Peso:7)</b>	9,75%	7,50%	0,00%	0,00%	3,90%
<b>UC 06 (Peso:7)</b>	13,25%	0,00%	0,00%	0,00%	2,99%
<b>UC 07 (Peso:10)</b>	17,00%	2,50%	0,00%	0,00%	4,40%
<b>UC 08 (Peso:7)</b>	6,75%	7,50%	0,00%	0,00%	3,22%
<b>UC 09 (Peso:7)</b>	12,50%	7,50%	0,00%	0,00%	4,52%
<b>UC 10 (Peso:10)</b>	0,00%	0,00%	0,00%	0,00%	0,00%
<b>UC 11 (Peso:7)</b>	0,00%	0,00%	0,00%	0,00%	0,00%
<b>UC 12 (Peso:7)</b>	0,00%	0,00%	0,00%	0,00%	0,00%
<b>UC 13 (Peso:7)</b>	0,00%	0,00%	0,00%	0,00%	0,00%
<b>Progresso do Projeto</b>	8,38%	4,69%	0,00%	0,00%	2,91%

---

Os casos de uso 07 e 10 apresentam uma complexidade média e receberam o peso 10, correspondente a complexidade adquirida. Os demais casos de uso foram classificados como simples e receberam o peso 7. Nesta inspeção o progresso obtido é fruto das dificuldades encontradas e enfrentadas pelas equipes do projeto. Uma das dificuldades mencionadas pela equipe foi à instabilidade dos requisitos do projeto, ou seja, requisitos não estáveis. A existência dessa dificuldade gerou atrasos significativos no cronograma do projeto e resultou no baixo progresso do projeto.

O atraso pode ser facilmente observado na visão de desempenho, conforme Figura 5.3 – Visão de desempenho obtida na primeira inspeção. A Figura 5.3 apresenta as métricas estabelecidas pelo processo Inspector, são elas:  $\mu_{concluído}$ ,  $\mu_{atraso\ médio}$  e  $\mu_{novas\ atividades}$ .

<b>Projeto:</b> Projeto xP	
<b>Data da Inspeção:</b> 15/08/2008 09:40:56	
<b>Equipe:</b>	Equipe Análise
Concluído:	5,54 %
Atraso Médio:	2,07 %
Nova Atividades:	0,00 %
<b>Equipe:</b>	Equipe Desenvolvimento
Concluído:	1,79 %
Atraso Médio:	1,88 %
Nova Atividades:	0,00 %

Figura 5.3 – Visão de desempenho obtida na primeira inspeção

### Segunda Inspeção

No dia 19 de setembro de 2008, foi realizada uma nova inspeção para determinar o status do projeto xP. O atraso observado na primeira inspeção resultou no replanejamento das atividades e exigiu um maior esforço das equipes do projeto. O esforço das equipes do projeto resultou em resultado significativo e o progresso do projeto encontrado nesta inspeção foi de  $\mu_{sistema} = 44,50$ , ou seja, 44,50% das funcionalidades tinham sido incorporadas ao sistema, conforme a Tabela 5.3 – Visão de funcionalidade obtida na

---

segunda inspeção.

A maioria dos casos de uso obteve resultados parecidos e significativos nas *ETAPAS* do projeto, porém o progresso do projeto ficou um pouco abaixo da expectativa do Gerente de Projeto. A organização não possui nenhuma base histórica e as métricas nunca foram coletadas, neste contexto, o Gerente de projetos não soube estabelecer o progresso esperado, apenas relatou que a expectativa esperada era atingir um progresso superior a 50,00%.

**Tabela 5.3 – Visão de funcionalidade obtida na segunda inspeção**

<b>Casos de Uso</b>	<b>Especificação Inicial</b>	<b>Análise e Projeto</b>	<b>Implementação</b>	<b>Teste</b>	<b>Progresso Total</b>
<b>UC 01 (Peso:7)</b>	100,00%	97,50%	35,00%	0,00%	55,89%
<b>UC 02 (Peso:7)</b>	98,75%	99,00%	0,00%	0,00%	44,65%
<b>UC 03 (Peso:7)</b>	100,00%	95,00%	15,00%	0,00%	48,87%
<b>UC 04 (Peso:7)</b>	100,00%	94,50%	0,00%	0,00%	43,92%
<b>UC 05 (Peso:7)</b>	97,50%	94,00%	0,00%	0,00%	43,24%
<b>UC 06 (Peso:7)</b>	100,00%	100,00%	0,00%	0,00%	45,16%
<b>UC 07 (Peso:10)</b>	95,00%	87,50%	0,00%	0,00%	41,21%
<b>UC 08 (Peso:7)</b>	100,00%	96,50%	0,00%	0,00%	44,37%
<b>UC 09 (Peso:7)</b>	100,00%	91,00%	0,00%	0,00%	43,13%
<b>UC 10 (Peso:10)</b>	96,25%	84,50%	0,00%	0,00%	40,81%
<b>UC 11 (Peso:7)</b>	98,75%	95,00%	0,00%	0,00%	43,75%
<b>UC 12 (Peso:7)</b>	100,00%	100,00%	0,00%	0,00%	45,16%
<b>UC 13 (Peso:7)</b>	99,25%	84,00%	0,00%	0,00%	41,38%
<b>Progresso do Projeto</b>	98,89%	93,73%	3,85%	0,00%	44,50%

---

A visão de desempenho pode ser observada na Figura 5.4, caracteriza o esforço da equipe de desenvolvimento e de análise no desempenho das atividades, bem como o acompanhamento, mais próximo, das atividades pelo gerente de projeto. O resultado satisfatório nesta inspeção é fruto da alocação de mais dois recursos humanos (temporários) e do desempenho das equipes do projeto.

Observa-se na Figura 5.4 a diminuição de atrasos nas atividades da equipe de desenvolvimento e a eliminação das atividades em atrasos na equipe de análise. Em comparação a primeira inspeção a equipe de análise conseguiu reduzir o atraso que antes era de 2,07% para 0,00%, ou seja, conseguiu eliminar 100% das suas atividades em atraso. Já a equipe de desenvolvimento reduziu o atraso que antes era de 1,88% para apenas 0,22%.

<b>Projeto: Projeto xP</b>	
<b>Data da Inspeção: 19/09/2008 15:09:44</b>	
<b>Equipe:</b>	<b>Equipe Análise</b>
Concluído:	65,03 %
Atraso Médio:	0,00 %
Nova Atividades:	0,00 %
<b>Equipe:</b>	<b>Equipe Desenvolvimento</b>
Concluído:	37,52 %
Atraso Médio:	0,22 %
Nova Atividades:	0,00 %

**Figura 5.4 – Visão de desempenho obtida na segunda inspeção**

### Terceira Inspeção

No dia 10 de outubro de 2008, foi realizada a terceira inspeção para determinar o novo status do projeto xP. O progresso do projeto encontrado nesta inspeção foi de  $\mu_{sistema} = 43,71$ , ou seja, 43,71% das funcionalidades tinham sido incorporadas ao sistema, conforme a Tabela 5.4 – Visão de funcionalidade obtida na terceira inspeção.

Houve uma pequena regressão no progresso funcional do projeto xP, que antes era

de 44,50% para 43,71%. Esta redução deve-se ao surgimento de um novo caso de uso do projeto, o UC 14 de complexidade média.

**Tabela 5.4 – Visão de funcionalidade obtida na terceira inspeção**

<b>Casos de Uso</b>	<b>Especificação Inicial</b>	<b>Análise e Projeto</b>	<b>Implementação</b>	<b>Teste</b>	<b>Progresso Total</b>
<b>UC 01 (Peso:7)</b>	100,00%	97,50%	52,50%	0,00%	61,53%
<b>UC 02 (Peso:7)</b>	98,75%	99,00%	15,00%	0,00%	49,49%
<b>UC 03 (Peso:7)</b>	100,00%	95,00%	32,50%	0,00%	54,52%
<b>UC 04 (Peso:7)</b>	100,00%	94,50%	27,50%	0,00%	52,79%
<b>UC 05 (Peso:7)</b>	100,00%	94,00%	0,00%	0,00%	43,81%
<b>UC 06 (Peso:7)</b>	100,00%	100,00%	0,00%	0,00%	45,16%
<b>UC 07 (Peso:10)</b>	98,75%	95,00%	0,00%	0,00%	43,75%
<b>UC 08 (Peso:7)</b>	100,00%	98,50%	0,00%	0,00%	44,82%
<b>UC 09 (Peso:7)</b>	100,00%	91,00%	0,00%	0,00%	43,13%
<b>UC 10 (Peso:10)</b>	100,00%	89,50%	0,00%	0,00%	42,79%
<b>UC 11 (Peso:7)</b>	98,75%	95,00%	0,00%	0,00%	43,75%
<b>UC 12 (Peso:7)</b>	100,00%	100,00%	0,00%	0,00%	45,16%
<b>UC 13 (Peso:7)</b>	86,75%	94,50%	0,00%	0,00%	40,93%
<b>UC 14 (Peso:10)</b>	37,50%	22,50%	0,00%	0,00%	13,55%
<b>Progresso do Projeto</b>	94,32%	90,43%	9,10%	0,00%	43,71%

Observe um decline no progresso funcional do caso de uso UC 13 na *ETAPA* Especificação Inicial, que antes era de 99,25% (segunda inspeção) para 86,75%. Este

---

decline foi influenciado pelo surgimento do novo caso de uso, onde foi necessário rever os artefatos produzidos para esta *ETAPA*. Durante a revisão e a análise do negócio, a equipe de análise identificou que o surgimento do novo caso de uso (UC 14), implicará no andamento do UC 13, ou seja, este caso de uso (UC 13) deve de ser alterado parcialmente para dar suporte ao novo caso de uso.

A visão de desempenho pode ser observada na Figura 5.5, onde as métricas  $\mu_{concluído}$ ,  $\mu_{atraso\ médio}$  e  $\mu_{novas\ atividades}$  são apresentadas.

<b>Projeto:</b> Projeto xP	
<b>Data da Inspeção:</b> 10/10/2008 11:29:18	
<b>Equipe:</b>	Equipe Análise
Concluído:	61,98 %
Atraso Médio:	0,03 %
Nova Atividades:	0,03 %
<b>Equipe:</b>	Equipe Desenvolvimento
Concluído:	39,07 %
Atraso Médio:	0,11 %
Nova Atividades:	0,01 %

**Figura 5.5 – Visão de desempenho obtida na terceira inspeção**

### **Acompanhamento do Cronograma**

Com a ferramenta, Inspector Panel, é possível realiza o acompanhamento das atividades estabelecidas no cronograma do projeto e visa à geração do gráfico *gantt*. Um *Gantt* de atividades consiste em um gráfico que indica as atividades necessárias para realização do projeto, indicando as datas de início e fim, a equipe responsável pela realização de cada atividade, entre outras informações.

A Figura 5.6 – Gráfico de *Gantt* gerado pela ferramenta exemplifica o gráfico gerado pela ferramenta. Com este gráfico é possível identificar as atividades que estão com problemas na execução, com base nas informações o gerente de projeto pode facilmente tomar ações corretivas.

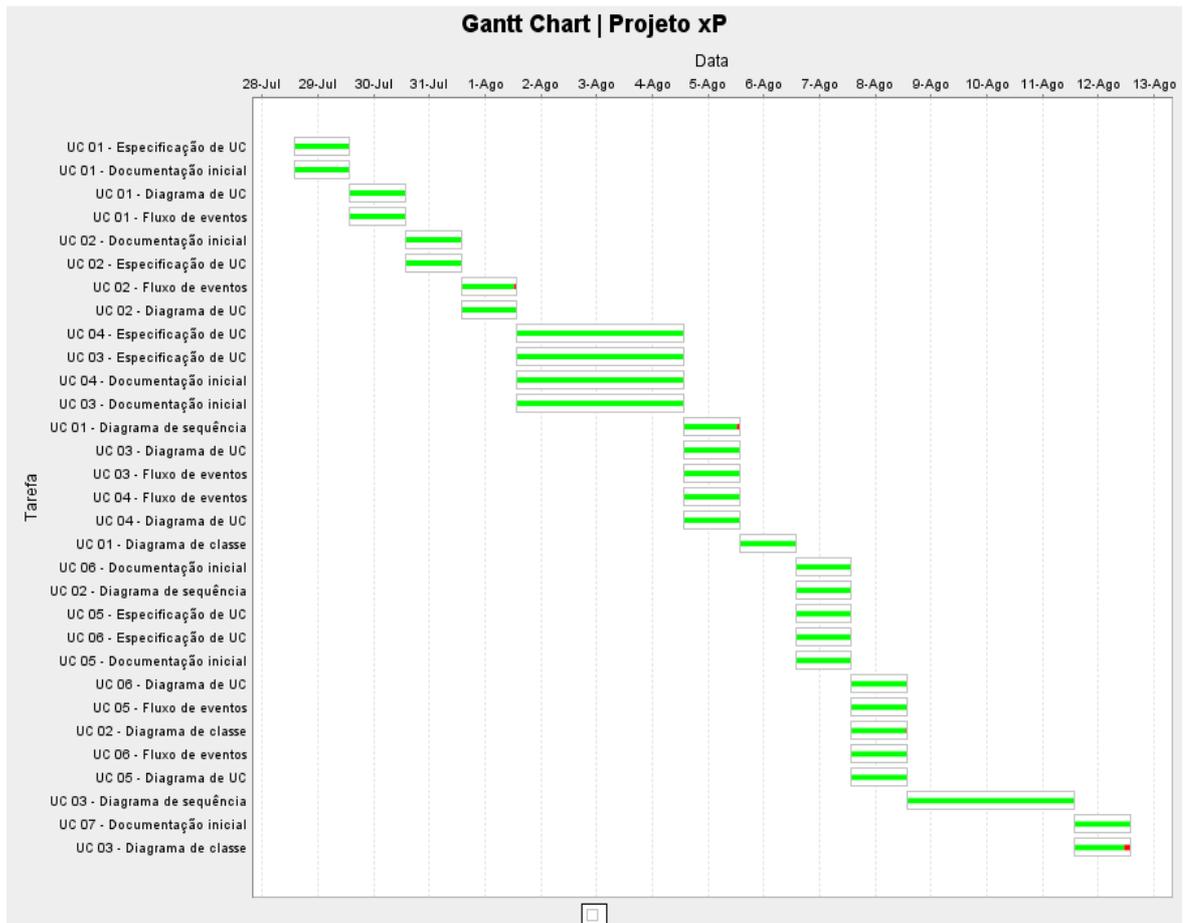


Figura 5.6 – Gráfico de Gantt

## Acompanhamento dos Custos

O acompanhamento dos custos não foi considerado no escopo deste estudo de caso. Mas a ferramenta, Inspector Panel, gera gráficos de barras, onde é possível comparar os custos estimados com real.

A ferramenta disponibiliza quatro gráficos diferentes para esta função, são eles: o gráfico de Custo por Projeto, o de Custo por Fase, o de Custo por Iteração e o de Custo por Caso de Uso. Com estes gráficos é possível identificar o caso de uso, ou a iteração, ou a fase está com problemas. A Figura 5.7 – Exemplo: Gráfico de custo por fases, exemplifica o gráfico de Custo por Fase, gerado pela ferramenta.

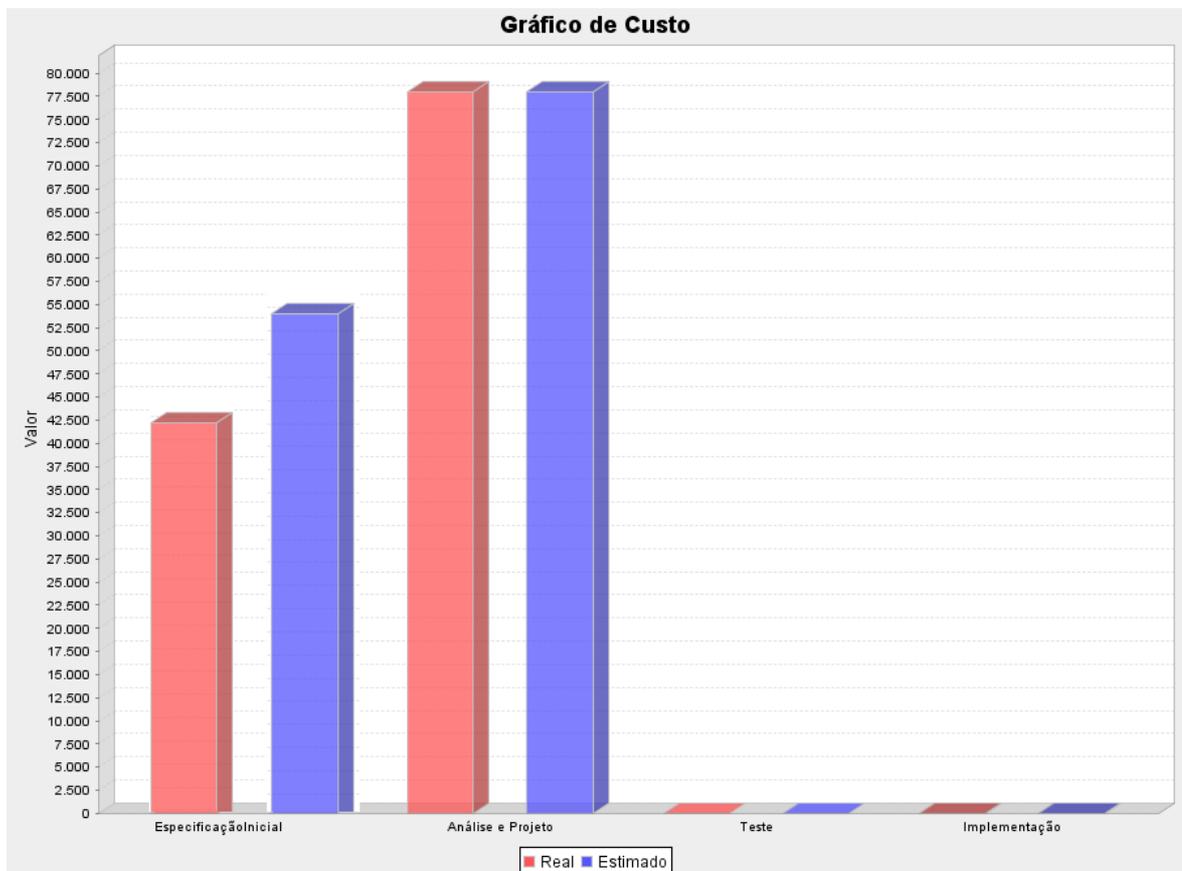


Figura 5.7 – Exemplo: Gráfico de custo por fases

## 5.2 Considerações Finais

A realização do estudo de caso que aplicou o processo Inspector, utilizando a ferramenta Inspector Panel, a um projeto real foi muito importante para identificar os pontos de melhorias. Um ponto a ser destacado foi à facilidade de identificar casos de usos que estão com problemas no seu desenvolvimento, através da visão de funcionalidade. O Inspector Panel apresenta, de forma bastante clara os casos de usos problemáticos e a fase onde as atividades necessitam de uma maior atenção da equipe.

Outro ponto de destaque, observado pela equipe do projeto, foi à flexibilidade da ferramenta na definição das fases do projeto, na definição das iterações e na definição do peso (complexidade) dos casos de uso. Com o uso da ferramenta, o gerente de projeto,

---

pode identificar de forma rápida o progresso dos casos de uso ( $\mu_{\text{caso de uso}}$ ) e o progresso do projeto ( $\mu_{\text{sistema}}$ ) a cada inspeção realizada, bem como identificar e acompanhar as equipes através das métricas  $\mu_{\text{concluído}}$ ,  $\mu_{\text{atraso médio}}$  e  $\mu_{\text{novas atividades}}$ .

A equipe do projeto identificou alguns pontos de melhorias, entre eles destacam-se:

- Melhoria no gráfico de *Gantt* gerado pela ferramenta – o padrão esperado e sugerido pela equipe do projeto, deve se aproximar aos gráficos de *gantt* gerados por diversas ferramentas já existentes, como o Microsoft Project [6] ou GanttProject [21].
- Melhoria no gráfico de progresso por casos de uso – a ferramenta disponibiliza apenas um gráfico com todos os casos de uso, caso um projeto tenha muitos casos de uso o gráfico ficaria com muitas informações, ficando impossível analisar o progresso de um caso de uso específico.
- Sugeriu a criação de um filtro antes da geração do gráfico de progresso por casos de uso, o filtro tem como objetivo especificar e gerar apenas informações de progresso dos casos de uso selecionado.

---

## Capítulo 6 Conclusão

---

Este capítulo apresenta as dificuldades enfrentadas na realização deste trabalho, as considerações finais sobre o trabalho desenvolvido nesta monografia, e os trabalhos futuros que podem ser realizados na área de pesquisa na qual este trabalho foi centrado.

### 6.1 Contribuições

O processo de avaliação de progresso, o Inspector, representa uma nova tecnologia que visa inserir facilidades para o acompanhamento mais próximo do projeto por parte do gerente de projetos. Apesar de ser o mais abrangente possível, algumas limitações foram impostas ao mesmo a fim de garantir sua efetividade: utiliza a UML (*Unified Modeling Language*) como linguagem de modelagem, focaliza projetos de software orientados a objetos baseados em casos de uso, apesar de possuir propriedades que independem de paradigma, não focaliza aspectos não funcionais e cognitivos, entre outras.

O trabalho descrito nesta monografia teve por objetivo a especificação e o desenvolvimento de uma ferramenta web, focalizado na automatização do processo Inspector, visando os aspectos gerenciais para facilitar o acompanhamento de projetos de software iterativos e incrementais, utilizando métricas estabelecidas no processo. Essa ferramenta poderá ser utilizada por pequenas, médias e grandes organizações desenvolvedoras de softwares, com o objetivo de avaliar o progresso dos projetos. Neste sentido, as principais contribuições deste trabalho são:

- A elicitação de requisitos para uma ferramenta de avaliação de progresso de projetos de software.
- Ferramenta web, facilitando sua utilização num ambiente distribuído.
- Automatização na coleta e processamento dos dados.

- 
- Evitar falhas e erros durante a coleta e processamento dos dados.
  - Criação de base histórica para a empresa.
  - Geração de indicadores visuais (gráficos).
  - Maior segurança.
  - Acompanhamento das atividades e dos custos do projeto.
  - Internacionalização do sistema (I18N).

## **6.2 Dificuldades Enfrentadas**

Durante a realização deste trabalho algumas dificuldades foram encontradas. Dentre elas podemos destacar: todos os riscos associados a um projeto de desenvolvimento de software; o fornecimento, a divulgação de informações relativas ao projeto xP; e aceitação da equipe de desenvolvimento do projeto xP sobre a visão de desempenho.

A visão de desempenho tem como objetivo apresentar a realização das atividades planejadas no plano de projeto durante o tempo. Ela verifica através de dados dispostos em gráficos, se as equipes de desenvolvimento têm realizado as tarefas propostas de maneira condizente, ou se atrasos estão ocorrendo. Por sua vez, a visão de funcionalidade apresenta o progresso dos requisitos funcionais no sistema, ou seja, quantos destes requisitos têm sido incorporados no sistema, e se esse progresso está adequado para o cronograma proposto.

## **6.3 Trabalhos Futuros**

Esta seção apresenta alguns trabalhos futuros a serem desenvolvidos e relacionados ao contexto desta monografia:

- Definir níveis de acesso ao sistema: a primeira versão da ferramenta, Inspector Panel, não trata dos níveis de acesso e permissões dos usuários do sistema.

- 
- Exportação e importação de dados: a ferramenta deve permitir a exportação de dados no formato XML. O XML é um padrão aberto, independentes das aplicações, dos sistemas operativos, etc. Trata-se de uma linguagem de marcação, que apesar da sua simplicidade, permite criar estruturas complexas; É extremamente flexível, possibilitando a representação de dados estruturados e/ou semi-estruturados, além de facilitar a comunicação entre sistemas.
  - Agregar parâmetros para avaliação da qualidade nos artefatos produzidos.
  - Integração entre as ferramentas: Microsoft Project [6] e o Inspector Panel.

---

## Referências Bibliográficas

---

- [1] KERZNER, H. Gestão de Projetos: As Melhores Práticas. Porto Alegre: Bookman, 2002.
- [2] MENEZES, L. C. M. Gestão de Projetos. 2ª ed., São Paulo: Atlas, 2003.
- [3] MAXIMIANO, A. C. A. Administração de Projetos: Como Transformar Idéias em Resultados. 3ª ed., São Paulo: Atlas, 1997.
- [4] PRADO, D. Gerenciamento de Projetos nas Organizações. 2ª ed., Belo Horizonte: Editora de Desenvolvimento Gerencial, 2003.
- [5] ABNT – Associação Brasileira de Normas Técnicas. NBR ISO/IEC 12207 – Tecnologia da Informação – Processos de Ciclo de Vida de Software. Rio de Janeiro: ABNT, 1998.
- [6] VARGAS, R. V. Microsoft Project 2000: Transformando Projetos em Resultados de Negócios. 3ª ed., Rio de Janeiro: Brasport, 2000.
- [7] CLELAND, D. I.; IRELAND, L. R. Gerência de Projetos. Rio de Janeiro: Reichmann & Afonso Editores, 2002.
- [8] HUMPHREY, W. Managing the Software Process. Addison-Wesley, 1990.
- [9] PMBOK – A Guide to the Project Management Body of Knowledge. 3ª ed., Pennsylvania: PMI, 2004.
- [10] MPS.BR – Associação para Promoção da Excelência do Software Brasileiro – SOFTEX. MPS.BR – Guia Geral, versão 1.2, 2008. Disponível em: [www.softex.br](http://www.softex.br). Último acesso em 06/2008.

- 
- [11] MENESES, J. B. Inspector: Um Processo de Avaliação de Progresso para Projetos de Software. Dissertação de Mestrado, CIn/UFPE, Pernambuco, 2001. Disponível em: <http://www.cin.ufpe.br/~inspector> . Último acesso em 10/2008.
- [12] GERALDA, P. Gerenciamento de Projetos. Parte da dissertação de Mestrado. Disponível em: <http://php.cin.ufpe.br/~pmk/hp/publicacoes/dissertacao/capitulo3PaulaTorreao.pdf>.  
Último acesso em 06/2008.
- [13] MORO, R. D.; NARDI, J. C.; FALBO, R. A. ControlPro: Uma Ferramenta de Acompanhamento de Projetos Integrada a um Ambiente de Desenvolvimento de Software. 19º Simpósio Brasileiro de Engenharia de Software - 2005. Disponível em: [www.sbbd-sbes2005.ufu.br/arquivos/ControlPro.pdf](http://www.sbbd-sbes2005.ufu.br/arquivos/ControlPro.pdf) . Último acesso em 02/2008.
- [14] VAVASSORI, F. B.; SOUZA, E. W.; FIAMONCINI, J. C. Ferramenta Case para o Gerenciamento de Projetos e Métricas de Software. XV Simpósio Brasileiro de Engenharia – 2001. Disponível em: [www.lbd.dcc.ufmg.br:8080/colecoes/sbes/2001/027.pdf](http://www.lbd.dcc.ufmg.br:8080/colecoes/sbes/2001/027.pdf) . Último acesso em 02/2008.
- [15] SBRAGIA, R.; MAXIMIANO, A. C. A.; KRUGLIANSKAS, I. O Gerente de Projetos: seu Papel e Habilidades. Revista da Administração, Vol. 21 (3), p. 24 – 31, Julho/ Setembro 1986.
- [16] CADOZO, E. L.; VILLIERS, DJ. Project Planning Best Practices. The Rational Edge. Agosto, 2003.
- [17] ERIKSSON, H. Business Modeling with UML: Business Patterns at Work. John Wiley & Sons. New York.
- [18] RUP – Rational Unified Process. Disponível em: [www.wthreex.com/rup](http://www.wthreex.com/rup) . Último acesso em 11/2008.
- [19] ECLIPSE – Disponível em: <http://www.eclipse.org/>. Último acesso em 02/2008.

- 
- [20] ARGOUML – Disponível em: <http://argouml.tigris.org/>. Último acesso em 02/2008.
- [21] GANTTPROJECT – Disponível em: <http://ganttproject.biz/>. Último acesso em 02/2008.
- [22] BUGZILLA – Disponível em: <http://www.bugzilla.org/>. Último acesso em 02/2008.
- [23] JUDE – Disponível em: <http://jude.change-vision.com/jude-web/index.html>. Último acesso em 02/2008.
- [24] SUN – Disponível em: <http://br.sun.com/>. Último acesso em 02/2008.
- [25] HIBERNATE – Disponível em: <http://www.hibernate.org/> . Último acesso em 02/2008.
- [26] SPRING – Disponível em: <http://www.springframework.org/>. Último acesso em 02/2008.
- [27] POSTGRESQL – Disponível em: <http://www.postgresql.org.br/>. Último acesso em 02/2008.

### Anexo A – Cronograma do Projeto Inspector Panel

<u>Atividades</u>	<u>Início</u>	<u>Fim</u>
<b><u>Concepção</u></b>	04/01/2008	08/03/2008
<b>Iteração 01</b>	04/01/2008	08/03/2008
<b>Gerenciamento de projeto</b>	25/02/2008	08/03/2008
<b>Plano de iteração (01) - Concepção</b>	25/02/2008	26/02/2008
<b>WBS</b>	26/02/2008	27/02/2008
<b>Plano de gerenciamento de risco</b>	03/03/2008	05/03/2008
<b>Lista de riscos</b>	05/03/2008	07/03/2008
<b>Documento de transposição de fases</b>	07/03/2008	08/03/2008
<b>Cronograma do projeto</b>	27/02/2008	28/02/2008
<b>Requisitos</b>	19/02/2008	23/02/2008
<b>Glossário</b>	19/02/2008	20/02/2008
<b>Documento de visão</b>	20/02/2008	22/02/2008
<b>Modelo de caso de uso</b>	22/02/2008	23/02/2008
<b>Análise e design</b>	07/02/2008	08/02/2008
<b>Protótipo</b>	07/02/2008	08/02/2008
<b>Modelo entidade relacionamento</b>	07/02/2008	08/02/2008
<b>Ambiente</b>	22/01/2008	23/01/2008

---

<b>Documento de ferramentas</b>	22/01/2008	23/01/2008
<b>Teste</b>	04/01/2008	19/01/2008
<b>Plano de teste</b>	18/01/2008	19/01/2008
<b>Protótipo validado</b>	04/01/2008	05/01/2008
<b><u>Elaboração</u></b>	18/03/2008	22/03/2008
<b>Iteração 01</b>	18/03/2008	22/03/2008
<b>Gerenciamento de projetos</b>	20/03/2008	22/03/2008
<b>Plano de iteração (01) - Elaboração</b>	20/03/2008	21/03/2008
<b>Plano de desenvolvimento de software</b>	21/03/2008	22/03/2008
<b>Requisitos</b>	20/03/2008	22/03/2008
<b>Modelo de caso de uso</b>	20/03/2008	21/03/2008
<b>Especificação de UC - Validar colaborador</b>	21/03/2008	22/03/2008
<b>Especificação de UC - Sair do sistema</b>	21/03/2008	22/03/2008
<b>Análise e design</b>	20/03/2008	21/03/2008
<b>Documento de arquitetura de software</b>	20/03/2008	21/03/2008
<b>Implementação</b>	18/03/2008	21/03/2008
<b>UC - Validar colaborador</b>	18/03/2008	20/03/2008
<b>UC - Sair do sistema</b>	20/03/2008	21/03/2008
<b><u>Construção</u></b>	04/03/2008	20/05/2008
<b>Iteração 01</b>	04/03/2008	13/05/2008
<b>Gerenciamento de projetos</b>	04/03/2008	05/03/2008
<b>Plano de iteração (01) - Construção</b>	04/03/2008	05/03/2008

---

---

<b>Requisitos</b>	05/03/2008	02/04/2008
<b>Especificação de UC - Consultar recurso</b>	05/03/2008	06/03/2008
<b>Especificação de UC - Cadastrar recurso</b>	06/03/2008	07/03/2008
<b>Especificação de UC - Alterar recurso</b>	07/03/2008	10/03/2008
<b>Especificação de UC - Excluir recurso</b>	10/03/2008	11/03/2008
<b>Especificação de UC - Consultar colaborador</b>	11/03/2008	12/03/2008
<b>Especificação de UC - Cadastrar colaborador</b>	12/03/2008	13/03/2008
<b>Especificação de UC - Alterar colaborador</b>	13/03/2008	14/03/2008
<b>Especificação de UC - Excluir colaborador</b>	14/03/2008	17/03/2008
<b>Especificação de UC - Consultar projeto</b>	17/03/2008	18/03/2008
<b>Especificação de UC - Cadastrar projeto</b>	18/03/2008	19/03/2008
<b>Especificação de UC - Alterar projeto</b>	19/03/2008	20/03/2008
<b>Especificação de UC - Excluir projeto</b>	20/03/2008	21/03/2008
<b>Especificação de UC - Consultar fase</b>	21/03/2008	24/03/2008
<b>Especificação de UC - Cadastrar fase</b>	24/03/2008	25/03/2008
<b>Especificação de UC - Alterar fase</b>	25/03/2008	26/03/2008
<b>Especificação de UC - Excluir fase</b>	26/03/2008	27/03/2008
<b>Especificação de UC - Consultar caso de uso</b>	27/03/2008	28/03/2008
<b>Especificação de UC - Cadastrar caso de uso</b>	28/03/2008	31/03/2008

---

---

<b>Especificação de UC - Alterar caso de uso</b>	31/03/2008	01/04/2008
<b>Especificação de UC - Excluir caso de uso</b>	01/04/2008	02/04/2008
<b>Implementação</b>	01/05/2008	13/05/2008
<b>UC - Cadastrar recurso</b>	01/05/2008	02/05/2008
<b>UC - Alterar recurso</b>	02/05/2008	03/05/2008
<b>UC - Excluir recurso</b>	02/05/2008	03/05/2008
<b>UC - Consultar recurso</b>	05/05/2008	06/05/2008
<b>UC - Cadastrar colaborador</b>	05/05/2008	06/05/2008
<b>UC - Alterar colaborador</b>	06/05/2008	07/05/2008
<b>UC - Excluir colaborador</b>	06/05/2008	07/05/2008
<b>UC - Consultar colaborador</b>	05/05/2008	06/05/2008
<b>UC - Cadastrar projeto</b>	07/05/2008	08/05/2008
<b>UC - Alterar projeto</b>	08/05/2008	09/05/2008
<b>UC - Excluir projeto</b>	08/05/2008	09/05/2008
<b>UC - Consultar projeto</b>	07/05/2008	08/05/2008
<b>UC - Cadastrar fase</b>	08/05/2008	09/05/2008
<b>UC - Alterar fase</b>	09/05/2008	10/05/2008
<b>UC - Excluir fase</b>	09/05/2008	10/05/2008
<b>UC - Consultar fase</b>	08/05/2008	09/05/2008
<b>UC - Cadastrar caso de uso</b>	09/05/2008	10/05/2008
<b>UC - Alterar caso de uso</b>	12/05/2008	13/05/2008
<b>UC - Excluir caso de uso</b>	12/05/2008	13/05/2008
<b>UC - Consultar caso de uso</b>	09/05/2008	10/05/2008

---

---

<b>Iteração 02</b>	15/05/2008	20/05/2008
<b>Gerenciamento de projetos</b>	15/05/2008	16/05/2008
<b>Plano de iteração (02) - Construção</b>	15/05/2008	16/05/2008
<b>Plano de gerenciamento de mudança</b>	15/05/2008	16/05/2008
<b>Planilha de gerenciamento de mudança</b>	15/05/2008	16/05/2008
<b>Requisitos</b>	15/05/2008	16/05/2008
<b>Especificação de UC - Consultar atividade</b>	15/05/2008	16/05/2008
<b>Especificação de UC - Cadastrar atividade</b>	15/05/2008	16/05/2008
<b>Especificação de UC - Alterar atividade</b>	15/05/2008	16/05/2008
<b>Especificação de UC - Excluir atividade</b>	15/05/2008	16/05/2008
<b>Implementação</b>	16/05/2008	20/05/2008
<b>UC - Cadastrar atividade</b>	16/05/2008	17/05/2008
<b>UC - Alterar atividade</b>	19/05/2008	20/05/2008
<b>UC - Excluir atividade</b>	19/05/2008	20/05/2008
<b>UC - Consultar atividade</b>	16/05/2008	17/05/2008
<b><u>Transição</u></b>	11/06/2008	12/06/2008

---