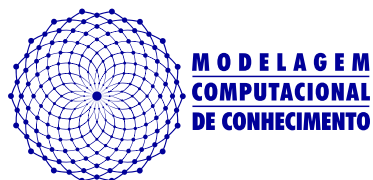


Universidade Federal de Alagoas  
Instituto de Computação



Dissertação de Mestrado

# **Um Conjunto de Indicadores para o Processo de Desenvolvimento de Software**

Davy de Medeiros Baía  
davybaia@gmail.com

Maceió  
2013

Davy de Medeiros Baía

# **Um Conjunto de Indicadores para o Processo de Desenvolvimento de Software**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Modelagem Computacional de Conhecimento do Instituto de Computação da Universidade Federal de Alagoas.

Orientador: Dr. Rodrigo de Barros  
Paes

Maceió  
2013

**Catálogo na fonte**  
**Universidade Federal de Alagoas**  
**Biblioteca Central**  
**Divisão de Tratamento Técnico**  
**Bibliotecária: Helena Cristina Pimentel do Vale**

B152u Baía, Davy de Medeiros.

Um Conjunto de Indicadores para o Processo de Desenvolvimento de Software / Davy de Medeiros Baía. – 2013  
69 f. : il.

Orientador: Rodrigo de Barros Paes  
Dissertação (Mestrado em Modelagem Computacional de Conhecimento) – Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2013.

Bibliografia: f. 67–69.

1. Software – Gerenciamento de Projeto. 2. Desenvolvimento de software – Indicadores. 3. GQ(I)M. I. Título.

CDU: 004.413



Membros da Comissão Julgadora da Dissertação de Mestrado de Davy de Medeiros Baía, intitulada: “Um Conjunto de Indicadores para o Processo de Desenvolvimento de Software”, apresentada ao Programa de Pós-Graduação em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas em 25 de fevereiro de 2013, às 08h30min, na sala de aula do Mestrado em Modelagem Computacional de Conhecimento.

### COMISSÃO JULGADORA

**Prof. Dr. Rodrigo de Barros Paes**

UFAL – Instituto de Computação

Orientador

**Prof. Dr. Evandro de Barros Costa**

UFAL – Instituto de Computação

Examinador

**Prof. Dr. Patrick Henrique da Silva Brito**

UFAL – Instituto de Computação

Examinador

**Prof. Dr. Hyggo Oliveira de Almeida**

UFCC – Departamento de Sistemas e Computação

Examinador

Maceió, fevereiro de 2013.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus! Sem fé e perseverança não chegamos em lugar algum. Posteriormente a todos que colaboraram de forma direta ou indireta, o orientador, os co-orientadores, que são os colegas acadêmicos, pais, irmãs, sobrinhos, namorada, enfim todos que colaboraram. Desde já agradeço.

## RESUMO

Projeto de desenvolvimento de software e seus resultados vêm obtendo maior atenção por motivos de não atingirem os resultados esperados. Algumas pesquisas revelam que custo e tempo não saem de acordo com o orçado ou planejado. O gerente de projeto tem um papel importante para obtenção desse resultado. Suas atribuições de alocação de recursos, ajuste de prioridades, monitoramento e controle da evolução do produto ou processo de desenvolvimento de software impactam diretamente na obtenção das metas a serem alcançadas. Tais atividades requerem um grande número de informações. Porém, saber quais informações são importantes para auxiliar o gerente de projeto ainda é um desafio. Esta dissertação propõe um conjunto de indicadores com o objetivo de auxiliar o gerente de projeto, utilizando a visão do PMBOK para acompanhar as dimensões de custo e tempo. Para isto, foi utilizada a abordagem GQ(I)M que possui todas as características necessárias para a criação dos indicadores. Além disso, o método de criação serve como um modelo para geração de novos indicadores. Para avaliar o conjunto de indicadores foi realizado um estudo de caso qualitativo e explanatório, fornecendo uma descrição detalhada de como o conjunto de indicadores auxiliou o gerente de projeto.

**Palavras-chave:** Gerenciamento de Projeto. Indicador. GQ(I)M. Gerenciamento de Tempo. Gerenciamento de Custo.

## **ABSTRACT**

The software development process and its results have received the Substantial attention from academia for the past years to the Mainly Regarding Lack of achievement of planned results. Recent researches Indicate That there is a large discrepancy Regarding to the current cost and time taken and the timetable. The project manager has an important role in Achieving planned results. It involves Activities such as allocating resources, setting Priorities, monitoring and controlling the product evolution process, Which has a straight impact in Achieving the expected goals. Despite the management of such a large amount Activities require of information, specifying what type of information is suitable to assist the project manager remains a challenge. The present dissertation Proposes a set of indicators to better assist the project manager by using PMBOK vision to Accompany the dimentions of cost and time. In order to accomplish the dissertation goals, it Has Been used the GQ (I) M (Goal question metric Indicator) approach, Which offers all features needed to create such indicators. Thus, to Evaluate the Proposed set of indicators, it Has Been executed the qualitative and explanatory case study, providing a detailed description of how the indicator has Helped September the project manager.

**Keywords:** Project Management. Indicator. GQ(I)M. Time Management. Cost Management.

## LISTA DE FIGURAS

3.1	Passo do GQ(IM)	22
3.2	Relação das dimensões	23
3.3	Etapas antes da efetivação da medição	31
3.4	Indicador - Distribuição de tipos de defeitos por projetos	31
3.5	Etapas do método de criação dos indicadores	34
4.1	Indicador 01 (I-1) - Total de Defeitos(bloqueio) x Pedidos	36
4.2	Indicador 02 (I-2) - Defeito por Gravidade x Estado do Defeito	37
4.3	Indicador 03 (I-3) - Atividades por pessoa x atividade por estado	38
4.4	Indicador 04 (I-4) - Esforço Planejado x Esforço Real	39
4.5	Indicador 05 (I-5) - Esforço Planejado da Tarefa x Esforço Real da Tarefa	40
4.6	Indicador 06 (I-6) - Tendências do item de escopo	41
4.7	Indicador 07 (I-7) - Produtividade	42
4.8	Indicador 08 (I-8) - Custo do Projeto	43
4.9	Indicador 09 (I-9) - Taxas de Chegada e Encerramento do Item de Escopo	44
5.1	Ambiente do estudo de caso	46
5.2	Ambiente do estudo de caso	49



## LISTA DE TABELAS

3.1	Questões Levantadas para a meta MT-01 . . . . .	24
3.2	Questões Levantadas para a meta MT-02 . . . . .	25
3.3	MT-01 Obter boa produtividade; . . . . .	26
3.4	MT-02 Cumprir os compromissos de prazo e custo assumidos . . . . .	26
3.5	Entidades e atributos identificados para a meta: MT-01 . . . . .	27
3.6	Entidades e atributos identificados para a meta: MT-02 . . . . .	28
3.7	Metas de medição estipuladas . . . . .	30
3.8	Questões quantificáveis e respectivos indicadores . . . . .	32
3.9	Elementos de dados identificados . . . . .	33
6.1	Trabalhos Relacionados . . . . .	52
A.1	Entrevista - Gerente A . . . . .	65
A.2	Entrevista - Gerente B . . . . .	66
A.3	Entrevista - Gerente E . . . . .	67
A.4	Entrevista - Líder de Equipe A . . . . .	68
A.5	Entrevista - Desenvolvedor A . . . . .	69

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>9</b>
1.1	Contextualização e Objetivos do Trabalho	9
1.2	Relevâncias da Pesquisa	10
1.3	Estrutura	11
<b>2</b>	<b>Engenharia de Software</b>	<b>12</b>
2.1	Processos de Engenharia de Software	12
2.2	Fábrica de Software	13
2.3	Métricas e Medidas de Software	14
2.4	Gerenciamento de projeto	16
<b>3</b>	<b>Método para Construção dos Indicadores</b>	<b>20</b>
3.1	Goal-Question-Metric	20
3.2	Goal-Driven Software Measurement	21
3.2.1	Identificar metas de negócio	22
3.2.2	Identificar o que se quer saber ou aprender	23
3.2.3	Identificar submetas	26
3.2.4	Identificar as entidades e os atributos	26
3.2.5	Efetivar os objetivos da medição	28
3.2.6	Identificar questões quantificáveis e respectivos indicadores	31
3.2.7	Identificando os elementos de dados	32
<b>4</b>	<b>Conjunto de Indicadores</b>	<b>35</b>
4.1	Defeitos(Bloqueio) x Pedido	36
4.2	Defeito por Gravidade x Estado do Defeito	37
4.3	Atividades por pessoa x atividades por estado	38
4.4	Esforço Planejado x Esforço Real (Projeto)	39
4.5	Esforço Planejado da Tarefa x Esforço Real da Tarefa	40
4.6	Tendências do item de escopo	41
4.7	Produtividade	41
4.8	Custo do Projeto	42
4.9	Taxas de Chegada e Encerramento do Item de Escopo	43
<b>5</b>	<b>Estudo de Caso</b>	<b>45</b>
5.1	Ambiente	45
5.2	Problema	46
5.3	Passo e explicação do método	47
5.4	Coleta de Dados	49
5.5	Análise de dados coletados	50
<b>6</b>	<b>Revisão</b>	<b>52</b>
6.1	Trabalhos objetivando Métricas	53
6.2	Trabalhos objetivando Indicadores	55
<b>7</b>	<b>Conclusão</b>	<b>58</b>

7.1	Limitações . . . . .	59
7.2	Trabalhos Futuros . . . . .	59
	<b>Referências</b>	<b>61</b>
<b>A</b>	<b>APÊNDICE - ENTREVISTAS</b>	<b>64</b>

# 1

## Introdução

### 1.1 Contextualização e Objetivos do Trabalho

O desenvolvimento de software em grandes organizações é uma atividade que envolve equipes que trabalham de forma colaborativa na resolução de problemas que muitas vezes não são triviais. Os membros destas equipes seguem processos de desenvolvimento definidos pela organização na qual trabalham. Estes processos são idealmente auxiliados por ferramentas automatizadas, assim fornecendo suporte a execução. Durante o projeto, com o auxílio dessas ferramentas, são gerados muitos dados que refletem o andamento do processo de desenvolvimento da solução. Estes dados ficam armazenados nas bases de dados de cada uma dessas ferramentas, porém o seu valor do ponto de vista de informação é pequeno se analisados isoladamente.

Grande parte das organizações possui o papel de gerente de projetos em sua equipe, onde algumas de suas atividades são: alocar recursos, ajustar as prioridades, monitorar e controlar a evolução e manter a equipe do projeto concentrada na meta a ser alcançada. Além disso, tais atividades envolvem geralmente uma grande quantidade de informações, principalmente no contexto dos grandes projetos de software. Nesses casos, o registro e a análise manual dos dados tornam-se inviáveis, tanto pelo risco de ocorrência de erros decorrentes de falhas humanas, quanto pelo custo envolvido. A existência de ferramentas automatizadas que proporcionem um suporte a essas atividades torna-se essencial.

Informações coletadas devem ser armazenadas de forma estruturada, preferencialmente em uma base de dados única. Para isso, é necessário que os aspectos de processo e de produto que serão submetidos à medição sejam agrupados em um modelo único, que especifique com precisão as entidades envolvidas, seus atributos e relacionamentos. É esse modelo que será utilizado como base para a análise das medidas coletadas e permitirá o acompanhamento e controle dos dados do projeto.

A Medição em um processo de software e seus produtos é realizada pela aplicação de determinadas métricas de software. Essas medidas quantitativas podem ser utilizadas como indicadores de um produto de software ou processo de desenvolvimento. Tais indicadores podem ser utilizados no gerenciamento de projetos em Fábricas de Software. Segundo o modelo de [Fernandes & Teixeira \(2004\)](#) uma Fábrica de software pode conter quatro escopos, sendo um deles o escopo de Fábrica de Programas. Embora os outros escopos tenham sua relevância em termo de complexidade e abrangência, concentramos os estudos no escopo de Fábrica de Programas, pois é nele que se concentram os processos das boas praticas do [PMBOK \(2004\)](#). Estas métricas, tais como tamanho, demanda, produtividade, quantidade de defeitos, tarefa planejada e realizada, são de interesse para gerentes de desenvolvimento de software, desenvolvedores e usuários. As métricas de software podem indicar sugestões para melhorar o processo de desenvolvimento de software.

Embora a implementação de um procedimento de mensuração traga benefícios em uma organização, não se constitui uma tarefa trivial, devido à quantidade e à complexidade dos fatores envolvidos no desenvolvimento de um software. Torna-se necessário selecionar apenas um subconjunto restrito, de forma a viabilizar as atividades de coleta e análise. Essa seleção desses indicadores a serem coletados deve ser cuidadosamente estudada e fortemente fundamentada, pois interfere diretamente no andamento e acompanhamento do projeto. Sem uma política bem definida para a definição desse conjunto de indicadores, a probabilidade de se escolher arbitrariamente as mais adequadas é remota.

Existem muitos trabalhos com propostas de utilizações de métricas e indicadores para processo de desenvolvimento de software, tais como ([Monteiro & de Oliveira, 2011](#); [McQuighan & Hammell, 2011](#); [Chandrasekaran & Kumar, 2012](#); [Islam et al., 2012](#); [Subramanyam & Krishnan, 2003](#); [Xie et al., 2009](#); [Paes, 2005](#)). Porém muitas empresas também desenvolveram seus próprios modelos de qualidade baseados em métricas de software. Além disso, alguns ambientes de desenvolvimento, como o Rational Insight, já incorporaram em suas características o apoio à aplicação de métricas de software. No entanto, nenhuma delas aborda a criação de um conjunto de indicadores gerados pelas métricas apropriadas para genericamente de projeto. Este trabalho possui 9 indicadores com abordagem nas áreas de custo e tempo seguindo as boas práticas do PMBOK.

## 1.2 Relevâncias da Pesquisa

Este trabalho propõe um conjunto de indicadores cujos objetivos principais são auxiliar o gerente de projetos a avaliar o andamento de projetos e a tomada de decisão nas áreas de custo e tempo. O conjunto de Indicadores é composto por métricas criadas a partir dos princípios bem conhecidos do paradigma GQM (Goal- Question-Metric), com informações obtidas nas bases de dados das ferramentas de apoio ao desenvolvimento. O conjunto

de indicadores foi criado para ser aplicado em projetos de desenvolvimento de software utilizando as boas praticas [PMBOK \(2004\)](#). Eles avaliam algumas dimensões pertencentes ao [PMBOK \(2004\)](#), tais como tempo e custo. Um gerente de Software pode usar o conjunto de indicadores como auxílio no acompanhamento do desenvolvimento de soluções. Finalmente, a criação e confecção de um conjunto de indicadores trazem como principais benefícios: i) Um modelo de aplicação para criação de um conjunto de indicadores. Como demosramos passo a passo como foi criado o conjunto de indicadores, poderemos utilizar esse processo como modelo para criação de novos indicadores; ii) Um conjunto de indicadores nas áreas de tempo e custo(PMBOK); iii) Auxílio ao gerente em suas atividades como: tomada de decisão, acompanhamento e planejamento do projeto;

### 1.3 Estrutura

No Capítulo 2 são apresentados conceitos relacionados à área de engenharia de software, processo de desenvolvimento de software, fábrica de software, métricas e medições e gerência de projetos. O resultado desse estudo foi utilizado como fundamentação para o trabalho.

No Capítulo 3, apresenta-se o método de construção do conjunto de indicadores. O método utilizado foi o GQ(I)M proposto por [Park et al. \(1996\)](#), com algumas adaptações. O Capítulo ?? descreve o conjunto de indicadores instanciado em uma base não real.

O Capítulo 5 descreve o estudo de caso aplicando o conjunto de indicadores em um ambiente real, de empresa, com a intenção de identificar a sua capacidade para auxiliar o gerente de projeto. Em seguida no Capítulo 6, foi realizada uma análise sobre o estado da arte descrevendo alguns trabalhos relacionados sobre os temas correlacionados. Finalmente no Capítulo 7, são apresentadas discussões e conclusões desse trabalho, assim como trabalhos futuros.

# 2

## Engenharia de Software

Segundo [Sommerville \(2011\)](#), Engenharia de Software é uma abordagem sistemática para a produção de software; analisando os custos, prazos e riscos, assim como as necessidades dos clientes e produtores do software. Um projeto de software é realizado para produzir algum produto. Considera-se que um projeto de software foi bem sucedido quando o software atende aos requisitos dos usuários e suas expectativas, produzindo resultados corretos de forma consistente, sendo considerado pelos usuários um produto útil e utilizável. ([Wongthongtham et al., 2009](#))

### 2.1 Processos de Engenharia de Software

O conceito de processo de engenharia de software incorpora um conjunto de práticas sequenciais que são funcionalmente coerentes e reutilizáveis para organização de engenharia de software, em sua implementação e gestão. É normalmente referido como o processo de software ou simplesmente o processo ([Wang & Bryant, 2002](#)).

Segundo [Sommerville \(2011\)](#) existem muitos processos de software diferentes, mas todos devem incluir quatro atividades fundamentais para a engenharia de software:

1. Especificação de Software. A funcionalidade do software e as restrições a seu funcionamento devem ser definidas.
2. Projeto e implementação de software. O software deve ser produzido para atender às demandas do cliente.
3. Validação de software. O software deve ser validado para garantir que atenda às demandas do cliente.
4. Evolução do software. O software deve evoluir para atender às necessidades de mudança dos clientes.

## 2.2 Fábrica de Software

Segundo [Cusumano \(1989\)](#), o termo Fábrica de Software foi usado entre 1960 e 1970, nos Estados Unidos da América e no Japão. No Brasil, o conceito de fábrica de software começou a ser aplicado no início da década de 90, por [Fernandes & Teixeira \(2004\)](#), sendo deles uma das primeiras obras literárias para tal segmento no Brasil. Segundo [Fernandes & Teixeira \(2004\)](#), uma fábrica de software pode ter vários domínios de atuação, desde um projeto de software completo, até a codificação de um programa de computador. Sendo que para a maioria das empresas de serviços tem como Fábrica de Software a *Fábrica de Programas*, dedicada exclusivamente para a codificação de programas.

O objetivo da Fábrica de Software deve ser a geração de produtos requeridos pelos usuários ou clientes, com o mínimo de defeitos possível e a um preço(ou custo) competitivo e compatível que forneça a margem necessária para os investimentos em manutenção e melhoria da fábrica ([Fernandes & Teixeira, 2004](#)). Os autores enumeram uma série de atributos básicos a uma estrutura fabril para software, alguns deles são:

- deve haver um processo definido e padrão para o desenvolvimento do produto de software;
- entrada para fábrica( a ordem de serviço ou solicitação de serviço) deve ser padronizada;
- as estimativas de prazo e custo devem ser baseados na capacidade real de atendimento da fábrica a determinada demanda;
- deve haver métodos padrões de estimativas baseados em históricos;
- a fábrica deve ter, de preferência, tempos padrões de atendimento já estabelecidos de acordo com o domínio da aplicação, da plataforma tecnológica e do tamanho da demanda (programa e/ou projeto);
- os perfis de recursos humanos devem ser controlados e estar alinhados ao tipo de demanda (natureza e complexidade) da fábrica;
- a fábrica deve ter rigorosos controle dos recursos em termos de sua alocação, disponibilidade, necessidade futura e produtividade (esta deve ser medida);
- a fábrica deve ter um processo para o planejamento e controle da produtividade;
- a fábrica deve ter o controle do *status* das múltiplas demandas em seu processo e permitir rastreamento dessas demandas;
- a fábrica deve controlar todos os itens de *software* (documentos, métodos, procedimentos, ferramentas e códigos), criando uma biblioteca de itens;



- a fábrica deve ter o absoluto controle do andamento da execução de cada demanda;
- os produtos de *software* devem ser construídos de acordo com métodos, técnicas e ferramentas padronizadas;
- a fábrica deve ter mecanismo de apuração, apropriação e controle de custos;
- a fábrica deve ter mecanismo de medições de atributos de sua operação, tais como: tempos médios de atendimentos, densidade de defeitos dos produtos, eficiência de remoção de defeitos, exatidão das estimativas, e assim sucessivamente;
- a fábrica tem que melhorar seus processos de forma contínua, visando ao aumento de sua produtividade e à redução de seus custos de operação.

Vistos esses requisitos, [Fernandes & Teixeira \(2004\)](#) definem a Fábrica de Software como um processo estruturado, controlado e melhorado de forma contínua, considerando abordagens de engenharia industrial, orientado para o atendimento a múltiplas demandas de natureza e escopo distintas, visando à geração de produtos de Softwares, conforme os requerimentos documentados dos usuários e/ou cliente, da forma mais produtiva e econômica possível.

## 2.3 Métricas e Medidas de Software

De acordo com [Bhatti \(2005\)](#), as métricas de software têm se provado uma técnica eficaz para estimar, avaliar e melhorar a qualidade do software e da produtividade, ou seja, um programa de métricas de software fornece assistência para avaliar, controlar e identificar ações de melhoria para atingir as metas de qualidade. As métricas de software são de interesse por algumas razões:

- As medidas quantitativas podem ser utilizados como indicadores de um produto software ou processo de desenvolvimento. Estes indicadores, tais como tamanho, produto, qualidade, qualidade do processo são de interesse para gerentes de desenvolvimento de software, desenvolvedores e usuários.
- As métricas de software podem indicar sugestões para melhorar o processo de desenvolvimento de software.

Para [Baumert & McWhinney \(1992\)](#) organizações que possuem um programa de métricas implantado com sucesso apresentam os seguintes benefícios:

- Gerentes mais engajados no projeto.
- Maior capacidade para quantificar decisões a serem tomadas.
- Melhor planejamento, controle e monitoramento de projetos.

- Melhor entendimento do processo e ambiente de desenvolvimento
- Identificação de áreas de potencial melhoria no processo, bem como uma métrica objetiva do esforço necessário para melhoria.
- Comunicação entre membros da organização otimizada.

A medição em um processo de desenvolvimento de software e seus produtos é realizada através da aplicação de determinadas métricas de software. Uma medição é composta normalmente por várias métricas, resultando novamente em várias medições por métricas. Alguns categorias e exemplos de métricas podem ser encontrados na literatura, alguns exemplos são (van Solingen & Berghout, 1999; Fenton, 1996; Grady, 1992). Os tipos mais comuns de métricas são descritos abaixo.

### **Métricas de produto e processo**

Uma métrica de produto é uma medida de um produto intermediário ou final de desenvolvimento de software e, portanto, aborda a saída de uma atividade de desenvolvimento de software. Métricas de processo mede as características do processo de desenvolvimento como um todo.

### **Métricas objetivas e subjetivas**

Métricas objetivas são medidas tomadas somente no processo ou produto, são atributos de contagem ou características de uma forma objetiva, por exemplo, o número de linhas de código ou número de falhas descobertas. Essas métricas têm um ponto de partida fundamental, um zero natural. Métricas subjetivas são medidas de um processo ou produto que envolvem o julgamento humano, subjetivo. Alguns exemplos de métricas subjetivas são: complexidade ou grau de conformidade com normas de codificação. Essas medições são classificações de observações humanas.

### **Métricas diretas e indiretas**

Uma métrica direta é uma medida de um processo ou característica do produto que não depende da medição de qualquer outra característica. Exemplos disso são: o número de falhas em um produto ou número de horas gastas durante determinado processo. Uma métrica indireta, por outro lado, é uma medida de um processo ou característica do produto que envolve a medição de mais de uma características, tal como, relação de produção e defeito, o quanto produziu de código sem erro. Uma métrica indireta sempre contém um cálculo de pelo menos duas outras métricas.

Coletar medidas e desenvolver métricas são atividades atribuídas no processo de desenvolvimento de software de modo que indicadores sejam obtidos. Um indicador é uma métrica

ou combinação de métricas que fornece profundidade na visão do processo de software, projeto de software ou produto. Um indicador fornece profundidade de visão suficiente para permitir ao gerente do projeto ou engenheiro de software ajustar o processo, projeto ou produto (PRESSMAN, 2006) de acordo com suas necessidades.

## 2.4 Gerenciamento de projeto

Um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo (PMBOK, 2004). Ele surge para resolver um problema concreto, transformando idéias em ações.

A definição do termo “Projeto” é uma informação importante para o Gerenciamento de Projetos, uma vez que, se confundem o que se pode ou não ser chamado de projeto. As atividades rotineiras e de ciclo contínuo não podem ser chamadas de projeto. Um projeto é definido por seus objetivos e limites de recursos e tempo. Os objetivos de um projeto devem ser claros e possíveis de ser atingidos. Todo projeto tem início, meio e fim. Alguns projetos são definidos pela data de início, outros pela data de entrega, devendo está última data atender ao prazo de entrega, do(s) produto(s) e/ou serviço(s), negociados com os stakeholders (F, 2008).

Além desses fatores, um projeto também envolve recursos humanos e materiais que deverão ser otimizados para obtenção do produto e/ou serviço final com a qualidade esperada pelo contratante. Para que os projetos produzam o resultado esperado, eles precisam ser gerenciados desde a definição de seu escopo, até a entrega final do próprio projeto.

A importância do Gerenciamento de Projetos já vem sendo reconhecida, por diversas empresas de todos os setores da economia, como fator essencial para o sucesso de suas iniciativas. Iniciativas como o desenvolvimento de novos produtos, implantação de novas tecnologias, ampliação e criação de novas unidades organizacionais, quando conduzidas sob forma de projetos, produzem melhores resultados, por serem mais bem gerenciadas.

O gerenciamento minimiza a possibilidade de insucesso de um projeto, pois é feito um planejamento de todas as entregas do projeto, onde em cima desse planejamento, serão feitos monitoramentos e controles das atividades de execução do projeto, minimizando erros que pudessem acarretar num aumento no custo e no prazo de entrega do projeto.

O gerenciamento de projetos é a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos (PMBOK, 2004).

Habitualmente, os gerentes de projetos costumam se apoiar nos três pilares do gerenciamento de projetos: escopo, tempo e custo do projeto. Mas pouco se fala da qualidade. A qualidade do projeto é afetada pelo balanceamento desses três fatores. Projetos de alta qualidade entregam o produto, serviço ou resultado solicitado dentro do escopo, no prazo e dentro do orçamento. A relação entre esses fatores ocorre de tal forma que, se algum dos três

fatores mudar, pelo menos outro fator provavelmente será afetado. Os gerentes de projetos também gerenciam projetos em resposta a incertezas. Um risco do projeto é um evento ou condição incerta que, se ocorrer, terá um efeito negativo em pelo menos um dos objetivos do projeto (PMBOK, 2004).

A meta do gerenciamento de projetos, segundo o PMBOK, é conseguir exceder as necessidades e expectativas dos stakeholders, mas para isto deve haver um balanceamento entre as várias demandas concorrentes em relação:

- Escopo, tempo, custo e qualidade (objetivos do projeto);
- Stakeholders com necessidades e expectativas diferenciadas; e
- Requisitos identificados (necessidades) e requisitos não identificados (expectativas).

O PMBOK® (Project Management Body of Knowledge) é um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos, publicado desde 1987 que contém a metodologia do PMI (Project Management Institute) para Gerenciamento de Projetos.

O seu principal objetivo é fornecer uma visão geral de conhecimento em gerenciamento de projetos, sendo esse conjunto de conhecimento reconhecido como boas práticas em projetos, aplicáveis à maioria dos projetos, onde a aplicação correta dessas habilidades, ferramentas e técnicas podem elevar as chances de sucesso em vários tipos de projetos diferentes. Para isso, propõe nove áreas de conhecimento para o gerenciamento de projetos, com a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto, a fim de atender ao propósito para o qual ele está sendo executado. As nove áreas propostas são: integração, escopo, tempo, custo, qualidade, recursos humanos, comunicações, riscos e aquisições. Detalharemos as mais significativas para esse trabalho.

### **Gerenciamento do escopo do projeto**

Inclui os processos necessários para garantir que o projeto inclua todo o trabalho necessário, e somente ele, para terminar o projeto com sucesso. A principal preocupação compreende em definir e controlar o que está ou não incluído no projeto. O escopo descreve todos os produtos de um projeto, os serviços necessários para realizá-los e os resultados finais esperados. Descreve também o que é preciso fazer para se alcançar os objetivos com os recursos e funções especificadas. O escopo do projeto refere-se ao trabalho que precisa ser realizado para entregar um produto, serviço ou resultado com as características e funções especificadas.

O termo escopo pode referir-se a escopo do produto ou escopo do projeto, sendo o primeiro referente às características e funções que descrevam um produto, serviço ou resultado, e o segundo refere-se ao trabalho que precisa ser realizado para entregar um produto, serviço ou resultado com as características e funções especificadas.

### **Gerenciamento de tempo do projeto**

Objetivo é garantir o término do projeto no prazo de terminado. O gerenciamento de tempo é uma habilidade importante de qualquer gerente de projeto. É bastante comum o custo real ultrapassar o custo orçado (realizado no processo de planejamento do projeto) e um dos possíveis motivos para que essa situação venha a ocorrer, é a má gerência do tempo, podendo vir a causar prejuízos financeiros e até mesmo inviabilizar o projeto.

### **Gerenciamento de custos do projeto**

O gerenciamento de custos tem como objetivo elaborar e controlar o orçamento do projeto. O gerenciamento de custos trata principalmente do custo dos recursos necessários para terminar as atividades do cronograma. No entanto, o gerenciamento de custos do projeto também deve considerar o efeito das decisões do projeto sobre o custo de utilização, manutenção e suporte do produto, serviço ou resultado do projeto.

### **Gerenciamento da qualidade do projeto**

O gerenciamento da qualidade, objetiva garantir que o projeto satisfará as exigências para as quais foi contratado. É função do gerente de projetos equilibrar adequadamente o custo, o prazo, o escopo e a qualidade do projeto, para satisfazer as expectativas e as necessidades dos interessados no projeto. Portanto o gerenciamento da qualidade do projeto deve abordar o gerenciamento do projeto e do produto do projeto. Enquanto o gerenciamento da qualidade do projeto se aplica a todos os projetos, as medidas e técnicas de qualidade do produto são específicas do tipo particular de produto produzido pelo projeto. Não atender aos requisitos de qualidade pode trazer sérias consequências negativas para algumas ou todas as partes interessadas no projeto. Por exemplo, atender às necessidades do cliente sobrecarregando a equipe do projeto pode trazer consequências negativas na forma de esgotamento dos funcionários, erros sem motivo aparente ou retrabalho; Atender aos objetivos de cronograma do projeto apressando as inspeções de qualidade planejadas pode trazer consequências negativas quando os erros não são detectados.

### **Gerenciamento de recursos humanos do projeto**

Inclui os processos que organizam e gerenciam a equipe do projeto. A equipe do projeto é composta de pessoas com habilidades específicas e responsabilidades. Os membros de equipe devem estar envolvidos em boa parte do processo de planejamento do projeto, bem como em tomadas de decisões. O reconhecimento do trabalho da equipe, por parte do gerente de projetos, reflete positivamente no resultado do projeto, consequência disso, é o aumento do comprometimento para com o projeto. O tipo e o número de membros da equipe do projeto muitas vezes podem mudar conforme o projeto se desenvolve.

**Gerenciamento de riscos do projeto**

O gerenciamento de riscos do projeto inclui os processos que tratam da realização de identificação, análise, respostas, monitoramento e controle e planejamento do gerenciamento de riscos em um projeto; a maioria desses processos é atualizada durante todo o projeto. Os objetivos do gerenciamento de riscos do projeto são aumentar a probabilidade e o impacto dos eventos positivos e diminuir a probabilidade e o impacto dos eventos adversos ao projeto. O objetivo é maximizar os resultados de ocorrências positivas e minimizar as consequências de ocorrências negativas.

# 3

## Método para Construção dos Indicadores

Como apresentado no Capítulo 2, alguns fatores são determinantes para o sucesso ou fracasso de uma fábrica de software. Um desses fatores são as decisões tomadas pelo gerente de projetos. Para auxiliar nessas decisões, um acompanhamento adequado com um conjunto de indicadores pode contribuir para os objetivos que se pretende atingir. Boa parte dos projetos de software não atinge os objetivos (custos, prazos, satisfação do cliente), ou seja, os projetos fracassam por falta de um bom gerenciamento de projetos (Sant'Anna, 2004). A escolha dos indicadores implica na seleção de um conjunto de métricas que deve ser mensurado ao longo do processo. Para auxiliar na seleção das métricas que irão compor o conjunto de indicadores, este trabalho seguiu a abordagem GQ(I)M (Park et al., 1996), que é uma extensão do GQM, por ser bem difundida e se adequar bem ao trabalho proposto.

### 3.1 Goal-Question-Metric

O GQM (*Goal-Question-Metric*) é um paradigma bastante difundido para guiar a escolha das medidas mais adequadas para um processo de mensuração Basili (1994). Uma de suas principais características é a abordagem top-down para a definição das medidas. A utilização da abordagem de cima para baixo auxilia na melhor compreensão da utilização de medições no contexto de desenvolvimento de software. O paradigma GQM preconiza que o processo de mensuração não deve ser guiado pelas medidas, mas pelos objetivos que se pretende atingir com sua coleta. Assim as medições só devem ser realizadas se estiverem fortemente fundamentadas por metas claramente definidas por van Solingen & Berghout (1999). O ponto de partida para a escolha das medidas é, portanto, a definição das metas que irão guiar o restante do processo. De forma resumida, o GQM descreve três passos que devem ser seguidos (van Solingen & Berghout, 1999):

- Definir um conjunto de metas que se pretende atingir quanto ao processo e aos produtos com as atividades de medição. As metas geralmente estão diretamente rela-

cionadas a questões de relevância fundamental para a organização, como a satisfação dos usuários, a qualidade dos produtos e a produtividade.

- Gerar uma série de perguntas que traduzam essas metas em aspectos quantitativos, e que possam ser alvos de medição. A ideia é gerar perguntas que, se respondidas, ajudariam a organização a atingir as metas estabelecidas.
- Especificar as medidas (métricas) que precisariam ser coletadas de forma a obter as informações necessárias para responder às perguntas geradas.

## 3.2 Goal-Driven Software Measurement

Uma extensão do paradigma GQM proposta em 1996 pelo SEI (Software Engineering Institute), na Universidade de Carnegie Mellon é o Goal-Driven Software Measurement (Park et al., 1996). No GQM, as medidas necessárias são deduzidas diretamente a partir das perguntas. No Goal-Driven Software Measurement são introduzidos os indicadores, que constituem um nível intermediário entre perguntas e métricas. Essa versão adaptada do GQM é frequentemente denominada GQ(I)M – Goal-Question-Indicator-Metric.

A sua proposta é transformar os princípios do GQM em um processo completo de definição de política de medição, seguindo etapas de atividades que produzem no final um conjunto de medidas customizado para as necessidades de uma organização. A estrutura do processo é composta de dez etapas, que devem ser seguidas de maneira ordenada:

1. Identificar metas de negócio;
2. Identificar o que se quer saber ou aprender;
3. Identificar submetas;
4. Identificar as entidades e os atributos relacionados com as suas submetas;
5. Efetivar objetivos da medição;
6. Identificar questões quantificáveis e respectivos indicadores que serão usado para ajudar a atingir os objetivos de medição;
7. Identificar os elementos de dados que serão coletados para construção dos indicadores que ajudam a responder às perguntas;
8. Definir as medidas a serem usadas e fazer definições operacionais;
9. Identificar as ações para implementar as medidas;
10. Preparar um plano para implementar as medidas.

A Figura 3.1 demonstra as sete primeiras etapas nas quais iremos nos concentrar.



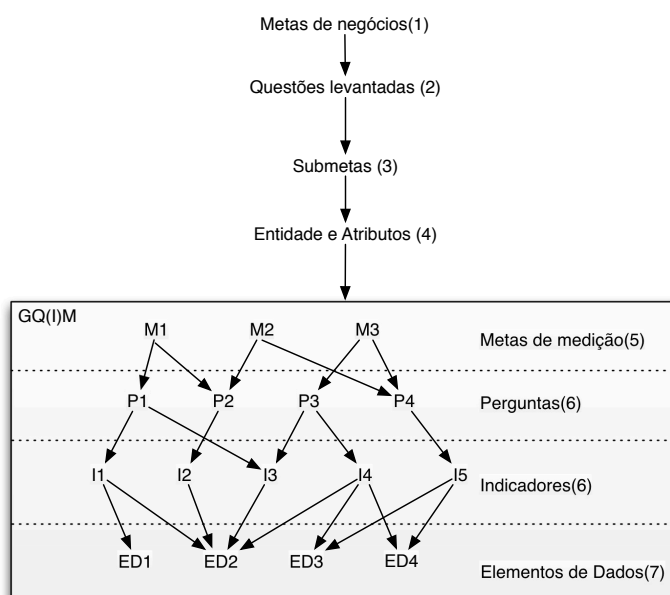


Figure 3.1: Passo do GQ(I)M

### 3.2.1 Identificar metas de negócio

Um estudo publicado na Harvard Business Review indica que projetos de TI são muito mais arriscados do que se pensa: entre os mais de 1.400 projetos pesquisados, uma média de 27% extrapolou o orçamento. 16% dos projetos poderiam ser chamados de Cisnes Negros: excederam o custo em mais de 200% e tiveram atrasos de mais de 70%, um nível de fracasso grande o suficiente para levar empresas à falência (Budzier, 2012). Outro estudo denominado Chaos Report revela a evolução da taxa de sucessos e de falhas dos projetos em TI. Em média, os projetos de software extrapolam o custo em 45%, o cronograma em 63% e apenas 63% da funcionalidade é entregue (Rubinstein, 2007). Fica evidenciado através desses estudos que as dimensões de custo, prazo e risco representam preocupações constantes na gerência dos processos e/ou projeto.

Neste contexto, com o objetivo de criar um conjunto de indicadores para processo de desenvolvimento de software, como auxílio às atividades do gerente de projeto, nessa primeira etapa identificamos algumas questões inerentes a essas preocupações:

1. Qualidade dos produtos;
2. Cumprimento de prazos para entrega dos produtos;
3. Custo do processo.

Existe uma correlação entre cada uma dessas questões com as dimensões de custo, tempo e risco. Por exemplo, a produtividade de software pode ser definida como a realização de uma unidade de tarefa dividido pelo tempo necessário para realizá-las. Uma técnica comum

para a medição de produtividade em projetos de software é a contagem de pontos de função (Sommerville, 2011). Com essa técnica, a produtividade é medida verificando-se quantas horas são necessários para se executar (implementar) um ponto de função. Sendo assim, intuitivamente, quanto maior a produtividade, mais unidades de trabalho são realizadas em um intervalo de tempo. Podemos verificar também quais são as unidades de trabalhos que não estão concluídas por um defeito que impossibilite sua finalização. A mitigação ou a identificação dos tipos de defeitos podem aumentar a qualidade (preocupação 1), reduzir custo (preocupação 3) e auxiliar no cumprimento dos prazos para entrega de produtos (preocupação 2). Ou seja, tanto produtividade como cumprimento de prazo e custo estão diretamente relacionada as preocupações 1, 2 e 3. A Figura 3.2 representa a relação destas questões evidenciada pelo PMBOK (2004). Por estas razões, neste trabalho, propõe-se a utilização destes fatores para a construção do conjunto de indicadores. Com isso as metas de negócio, utilizadas como base para a construção do conjunto de indicadores foram definidas como:

- Obter boa produtividade;
- Cumprir os compromissos de prazo e custo assumidos.

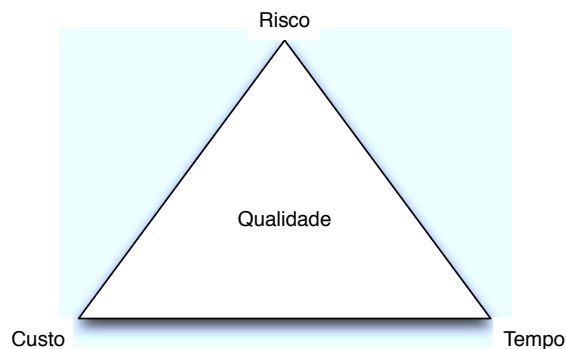


Figure 3.2: Relação das dimensões

Identificaremos estes fatores como metas MT-01 e MT-02 respectivamente. A escolha se justifica pela preocupação de obter uma boa produtividade seguindo tempo estimado e custo orçado. Como mostrado na Figura 3.2 essas metas estão relacionada aos pilares de custo, tempo e risco e sua resultante esperada é a qualidade.

### 3.2.2 Identificar o que se quer saber ou aprender

As metas precisam ser refinadas para que se tornem objetivos mensuráveis. Para iniciar esse refinamento se faz necessária a identificação do que se deseja saber ou aprender sobre o processo de desenvolvimento de software para que se possa compreender, prever ou melhorar

as atividades relacionadas a cada um dos objetivos escolhidos. Esse refinamento é feito através de agrupamento de questões por tema. Para cada uma dessas questões e temas relacionados nas Tabelas 3.1 e 3.2 existe uma relação com as obtenções de metas mencionadas na Seção 3.2.1. Por exemplo, a resposta para a quantidade de defeitos do tipo bloqueio que estão atrasando outros itens de trabalho pode ajudar a identificar se existe esse tipo de defeito e quantos são. Com a identificação, o gerente de projeto poderá começar uma iniciativa para que esse defeito não se torne um obstáculo para a obtenção das metas. As respostas a este conjunto de perguntas podem refletir sobre o andamento do projeto e auxiliar na obtenção das metas. Estas respostas também funcionam como uma ferramenta para tomada de decisão do gerente de projeto. Para a Tabela 3.1 foram geradas perguntas que irão servir de base para a obtenção da primeira meta (MT-01) - Obter boa produtividade.

Table 3.1: Questões Levantadas para a meta MT-01

Tema	Questões
Defeito	Q1.1 Qual é o número de defeitos (bloqueio) que estão atrasando outros itens de trabalho?
	Q1.2 Quais são os números de defeitos por tipos?
	Q1.3 Qual é a gravidade dos defeitos?
Tarefas Escopo	Q1.4 Qual a quantidade de esforço planejado e realizado?
	Q1.5 Qual a quantidade de atividades por pessoa?
	Q1.6 Qual é a estimativa de horas de trabalho restante para uma atividade?
	Q1.7 Quantidade de atividades e seu estado atual por pessoa?
	Q1.8 Qual a relação de quantidade de criação e encerramento de atividades?

Com essas questões, podemos descobrir informações relevantes para obtenção da MT-01. Tais perguntas podem ser respondidas pelos indicadores. Detalharemos o objetivo das perguntas da Tabela 3.1 a seguir:

- Q1.1** - Identificar quais são os defeitos que precisam ser acompanhados de forma mais cuidadosa;
- Q1.2** - Identificar quais são os tipos de defeitos que são inseridos no processo. Com isso, mitigar e evitar algum tipo de defeito específico;
- Q1.3** - Identificar qual é a gravidade do defeito. Com isso, mitigar os defeitos de gravidade que possam comprometer o projeto;
- Q1.4** - Identificar os esforços planejados e realizado. Com isso, verificar se o planejamento está de acordo com o planejado;

- Q1.5** - Identificar o esforço por pessoa. Com isso, podem ser atribuídas mais ou menos atividades por pessoa;
- Q1.6** - Identificar qual é a estimativa de horas restantes para uma atividade. Com isso, pode-se alocar mais ou menos recursos para o cumprimento de prazo ou custo planejados;
- Q1.7** - Identificar as atividades e seu estado. Com isso, mitigar ou controlar atividades que se encontram em determinado estado;
- Q1.8** - Identificar como está a criação e o encerramento de atividades. Com isso, equilibrar a quantidade de criações com encerramentos.

Para a Tabela 3.2 foram geradas perguntas que irão servir de base para a obtenção da segunda meta (MT-02) - Cumprir os compromissos de prazo e custo assumidos.

Table 3.2: Questões Levantadas para a meta MT-02

Tema	Questões
Tempo	Q2.1 Qual a quantidade de trabalho que falta ser realizado em relação ao tempo estimado?
	Q2.2 As atividades realizadas estão de acordo com o planejado?
Custo	Q2.3 Qual é o custo realizado?
	Q2.4 Os custos realizados estão de acordo com o planejado?

Com essas questões podemos descobrir informações relevantes para obtenção da (MT-02). Tais perguntas podem ser respondidas pelos indicadores. Detalharemos as perguntas da Tabela 3.2 a seguir:

- Q2.1** - Identificar a relação de quantidade de trabalho com tempo estimado. Com isso, fazer ajuste para cumprir prazo e custo orçado;
- Q2.2** - Identificar se as atividades estão com sua proporção de conclusão correta. Com isso, equilibra-se sua conclusão com o tempo estimado;
- Q2.3** - Identificar o custo realizado do projeto. Com isso, acompanhar o custo.
- Q2.4** - Identificar se o custo realizado do projeto está de acordo com o planejado. Com isso, acompanhar o custo realizado com o planejado.

As Tabelas 3.1 e 3.2 mostram as questões levantadas para as metas de negócio agrupando por temas. Algumas dessas perguntas dependem da definição do processo da empresa, como por exemplo: esforço e produtividade podem ser medidos por linhas de códigos ou por pontos de função; Estado da atividade reflete em que momento a atividade se encontra como, por exemplo, aberta, fechada, atribuída, resolvida. Na próxima etapa, vamos transformar as metas em itens específicos (submetas).

### 3.2.3 Identificar submetas

O objetivo da identificação das submetas é decompor as metas de negócio em submetas, tornando-as mais específicas. Essa decomposição resultará em submetas que poderão ser abordadas por ações gerenciais e técnicas. Por exemplo, para alcançarmos uma boa produtividade, uma das submetas utilizadas é a diminuição de defeitos gerados no processo de desenvolvimento. O resultado do cumprimento desta submeta resultará na diminuição de horas gastas para correção de defeitos. Com isso, essas horas poderão ser atribuídas em novas atividades ou simplesmente diminuir o custo final do produto, aumentando a produtividade ou diminuindo o custo final de produção.

As Tabelas 3.3 e 3.4 listam as metas em submetas de acordo com o agrupamento por temas já realizado.

Table 3.3: MT-01 Obter boa produtividade;

Submeta	Descrição
MT-01.01	Diminuir o número de defeitos durante todo o processo de desenvolvimento.
MT-01.02	Aumentar a confiabilidade do produto final.
MT-01.03	Diminuir o impacto dos defeitos na produtividade.
MT-01.04	Conhecer o tamanho de cada produto e o esforço necessário para produzi-los.
MT-01.05	Aumentar e/ou controlar os ganhos de produtividade.
MT-01.06	Distribuir uniformemente as atividades com as pessoas.
MT-01.07	Conhecer o estado das atividades.
MT-01.08	Planejar novas iterações, tendo em vista a tendência.

Table 3.4: MT-02 Cumprir os compromissos de prazo e custo assumidos

Submeta	Descrição
MT-02.01	Aumentar a precisão das estimativas de esforço.
MT-02.02	Aumentar a adequação dos planejamentos dos projetos.
MT-02.03	Controlar o andamento de cada tarefa do projeto.
MT-02.04	Controlar o custo de cada tarefa do projeto.

Com a lista das submetas definidas, a próxima etapa do GQ(I)M consiste em identificar as entidades e atributos envolvidos.

### 3.2.4 Identificar as entidades e os atributos

A identificação das entidades e atributos relacionados com as suas metas gerenciáveis definirá o cenário para a formulação de objetivos bem estabelecidos de medição que serão

utilizados posteriormente junto com a lista de questões. Segundo [Sant'Anna \(2004\)](#) uma entidade é um objeto (como uma pessoa ou um quarto) ou um evento (como uma viagem ou um projeto de desenvolvimento de um software) e um atributo é uma característica ou propriedade de uma entidade. Exemplos de atributos são tamanho de um quarto, o tempo de uma viagem ou o custo de um projeto de desenvolvimento de um software. Para cada pergunta identificamos a sua entidade e em seguida listamos os seus atributos.

O objetivo do atributo é contribuir para a obtenção de respostas às perguntas formuladas e ou estabelecer o contexto para a interpretação de tais respostas. Como exemplo, poderemos citar a distribuição por tipo de defeito, com sua classificação por tipo, podemos identificar quais são os defeitos do tipo bloqueio e quantificá-los, assim respondendo a pergunta relacionada a este atributo. As Tabelas 3.5 e 3.6 completa a identificação das entidades e atributos.

Table 3.5: Entidades e atributos identificados para a meta: MT-01

Pergunta	Entidade(s)	Atributo(s)
Q1.1 Qual é o número de defeitos (bloqueio) que estão atrasando outros itens de trabalho?	Conjunto de defeitos	Distribuição dos defeitos, por tipo;
Q1.2 Quais são os números de defeitos por tipos?	Conjunto de defeitos	Distribuição dos defeitos, por tipo;
Q1.3 Qual é a gravidade dos defeitos?	Conjunto de defeitos	Distribuição dos defeitos, por gravidade;
Q1.4 Quantidade de esforço planejado e realizado?	Estimativa de esforço	Esforço estimado; Esforço realizado;
Q1.5 Quantidade de atividade por pessoa?	Estimativa de esforço	Atividade por pessoa; Estado da atividade;
Q1.6 Qual é a estimativa de horas de trabalho restante para uma atividade?	Estimativa de esforço	Esforço estimado por atividade; Esforço realizado por atividade;
Q1.7 Quantidade de atividade e seu estado atual por pessoa?	Planejamento e acompanhamento	Atividade por estado; Atividade por pessoa;
Q1.8 Qual a relação de quantidade de criação e encerramento de atividades?	Planejamento e acompanhamento dos projetos	Atividade por estado;

Table 3.6: Entidades e atributos identificados para a meta: MT-02

Pergunta	Entidade(s)	Atributo(s)
Q2.1 Qual a quantidade de trabalho que falta ser realizado em relação ao tempo estimado?	Planejamento e acompanhamento dos projetos	Trabalho estimado; Trabalho realizado;
Q2.2 As atividades realizadas estão de acordo com o planejado?	Planejamento e acompanhamento dos projetos	Trabalho estimado; Trabalho realizado;
Q2.3 Qual é o custo realizado?	Planejamento e acompanhamento dos projetos	Custo realizado;
Q2.4 Os custos realizados estão de acordo com o planejado?	Planejamento e acompanhamento dos projetos	Custo realizado; Custo planejado;

Uma vez identificadas as metas de negócio e pontuando o que se deseja saber ou aprender, juntamente com as submetas e as questões relacionadas a cada uma delas, a próxima etapa do GQ(I)M consiste em identificar as entidades e atributos envolvidos para obtenção da medição listada nas Tabelas 3.5 e 3.6. Elaborando assim uma base para a efetivação dos objetivos de medição, descrita na seção seguinte.

### 3.2.5 Efetivar os objetivos da medição

Nas etapas anteriores, o foco principal foi identificar as metas de negócio e sua perspectiva que influencia no alcance dessas metas para ser aplicado de forma eficaz (Basili, 1994; Park et al., 1996; van Solingen & Berghout, 1999). A utilização desses dados levantados até o momento servirá de base para definição das metas do processo de medição, formando inicialmente a estrutura de metas, perguntas e medidas que modelam a base do paradigma GQ(I)M completando o paradigma, adicionando um INDICADOR. Nesta etapa iremos efetivar as questões e preocupação sinalizadas nas etapas anteriores em metas de medição claramente definidas.

Essa efetivação consiste na especificação de quatro componentes:

1. Um objeto de interesse, que pode ser um produto, processo, recurso, agente, artefato, medida ou ambiente. Também pode ser um conjunto de outras entidades ou objetos.
2. Uma proposta, que pode ser compreender, prever, planejar, controlar, comparar ou melhorar algum aspecto do objeto.
3. Uma perspectiva, que identifica quem está interessado nos resultados das medições (desenvolvedor, gerente de projeto, cliente, etc.).

4. Uma descrição do ambiente, que fornece um contexto para que os resultados das medições possam ser interpretados. Normalmente refere-se a aspectos do projeto que está sendo medido.

Foram geradas as 04 metas de medição de acordo com as etapas anteriores, apresentadas na Tabela 3.7. Esta tabela apresenta a formalização das metas em termos do objeto de interesse, proposta e perspectiva. A descrição do ambiente foi omitida por estar voltada para situações em que a política de medição tem como foco uma organização genérica, o que a torna incompatível nesse momento. No Capítulos 5 iremos tratar do ambiente. Para permitir um mapeamento, a última coluna da tabela mostra a relação de submetas às quais cada meta de medição está associada.



Table 3.7: Metas de medição estipuladas

Nº	Objeto de Interesse	Proposta	Perspectivas	Submetas relacionadas
1	Defeitos	Analisar a distribuição dos defeitos de natureza diversa, injetados e detectados no processo de desenvolvimento, com o objetivo de identificar oportunidades de melhoria no processo quanto à redução de defeitos.	Formar uma base de dados históricos para as atividades de melhoria de processo, identificando os pontos mais problemáticos quanto à inserção e detecção de defeitos ao longo do processo.	MT-01.01 MT-01.02 MT-01.03
2	Esforço	Analisar o esforço gasto em cada atividade do processo, com o objetivo de conhecê-lo e identificar formas de obter maior produtividade com menor esforço.	Fornecer embasamento quantitativo para a elaboração de estimativas pelo gerente de projeto.	MT-01.05 MT-01.06 MT-02.01 MT-01.04
3	Planejamento	Analisar a relação de estimado x realizados com o objetivo de conhecer sua acurácia e identificar oportunidades para melhorá-la.	Permitir um melhor conhecimento dos erros das estimativas pelos gerentes de projetos, possibilitando o dimensionamento dos fatores de contingência, e monitorar a previsibilidade do processo de desenvolvimento.	MT-02.01 MT-02.02 MT-02.04 MT-01.08
4	Acompanhamento	Analisar a fração já concluída dos projetos, com o objetivo de compará-la com as estimativas feitas e com os compromissos assumidos.	Prover aos gerentes de projeto um controle quantitativo sobre o andamento dos projetos.	MT-02.02 MT-02.03 MT-01.07

A Figura 3.3 descreve os passos anteriores para chegar até a efetivação da medição. Com isso, teremos de forma mais clara como chegamos nessa definição.

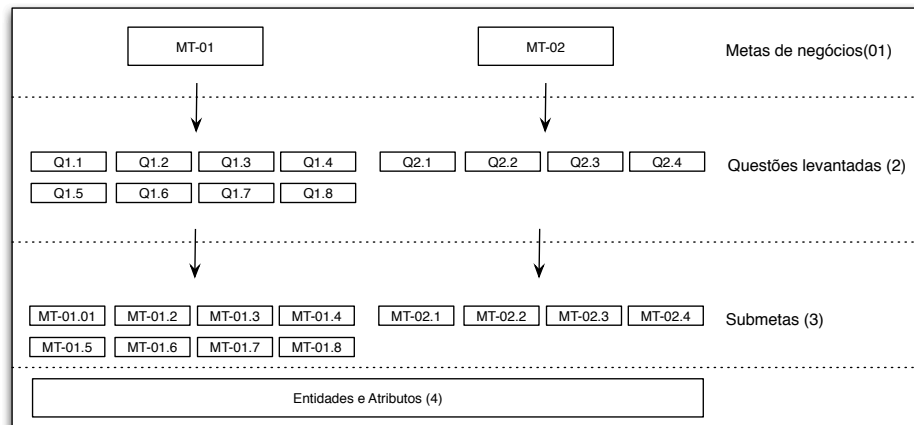


Figure 3.3: Etapas antes da efetivação da medição

### 3.2.6 Identificar questões quantificáveis e respectivos indicadores

Até o momento foram definidas as metas de medições que servirão como base para a utilização do paradigma GQ(I)M. Ao completar a primeira etapa, a próxima etapa é formular questões *quantificáveis* e *indicadores* que suportam as perguntas. Por exemplo, para a primeira meta de medição, referente à análise da distribuição de tipos de defeitos, as seguintes perguntas foram elaboradas:

- Qual é a quantidade de tipos de defeitos observada em cada produto desenvolvido?
- Qual tem sido a evolução dos tipos de defeitos ao longo dos projetos de desenvolvimento?

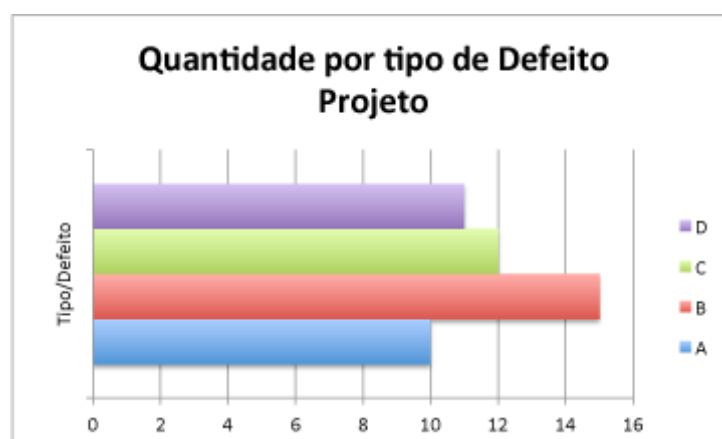


Figure 3.4: Indicador - Distribuição de tipos de defeitos por projetos

A utilização de um *Indicador* serve para exibir um ou mais resultados de medições com o objetivo de responder os questionamentos (questões quantificáveis) que foram criados para todas as metas de medições (Baumert & McWhinney, 1992; Park et al., 1996). A Tabela 6.1 exibe as questões elaboradas para cada uma das metas enumeradas na seção anterior, e associa a cada grupo de questões um conjunto de indicadores. Nela, cada indicador é apenas referenciado por seu código; sua descrição encontra-se no próximo capítulo.

Table 3.8: Questões quantificáveis e respectivos indicadores

Meta	Pergunta	Indicador
MT-01-GQ1	• Qual é a quantidade de defeitos aberto no projeto?	I-1
	• Qual tem sido a evolução dos tipos de defeitos ao longo dos projetos de desenvolvimento?	I-2
	• A quantidade de defeitos está sob controle, ou representa algum risco para a conclusão tempestiva do projeto?	
MT-01-GQ2	• Qual o esforço gasto em cada atividade?	I-3
	• Qual a distribuição de produção por pessoa?	I-4
	• Qual e a relação de atividade por pessoa e estado da atividade?	I-5
		I-7
MT-02-GQ1	• Os esforços foram subestimados?	I-4
	• Qual é a tendência do item de escopo?	I-6
	• Os custos foram subestimados?	I-7
		I-8
MT-02-GQ2	• Qual a fração já concluída do projeto?	I-4
	• Qual o esforço já gasto até então?	I-8
	• Como está a relação estimado x realizado?	I-9
	• Como esta a taxa de chagada e encerramento do escopo?	

A partir da lista dos indicadores, é possível definir quais as medidas que devem ser coletadas para possibilitar a construção de cada um, identificando os elementos de dados que serão coletados para construção dos indicadores que ajudam a responder às perguntas.

### 3.2.7 Identificando os elementos de dados

Para concluir a identificação de dados, basta fazer uma lista de todos os elementos de dados que são necessário recolher para construção dos indicadores sugerido na Seção 3.2.6. Dando continuidade ao exemplo anterior (Figura 3.4), é necessário a coleta das seguintes informações:

- Projeto ao qual pertence o defeito;
- Classificação do defeito quanto ao tipo;
- Data em que o defeito foi identificado.

Essa identificação se faz necessária, pois mapeia os elementos de dados para construção desse indicador. Do mesmo modo fazendo o mesmo para cada um dos indicadores criados, obtém-se um conjunto completo dos elementos de dados que devem ser coletados ao longo do processo de desenvolvimento. Até o momento, devido ao trabalho feito anteriormente, pode-se afirmar que cada um desses elementos corresponde a pelo menos uma proposta objetiva de medição, que por sua vez está alinhada com as metas de negócio da organização. Sendo este, o principal benefício trazido pelo Goal-Driven Software Measurement: a possibilidade de mapear cada um dos elementos de dados com as metas que motivaram sua coleta. A Tabela 3.9 apresenta todos os elementos de dados que são necessários para a construção dos indicadores enumerados na Seção 3.2.6.

Table 3.9: Elementos de dados identificados

Entidade	Grupo	Elemento de dado	Indicadores
Conjunto de Defeito	ED-1	Estado do defeito;	I-1
		Classificação quanto ao tipo do defeito;	I-2
Esforço	ED-2	Atividade por pessoa;	I-3
		Estado da atividade; Número de atividade; Produtividade por pessoa;	I-7
Planejamento e Acompanhamento	ED-3	Tamanho estimado do esforço por tarefa;	I-4
		Tamanho de esforço realizado por tarefa;	I-5
	ED-4	Distribuição de incertezas;	I-6
		Custo estimado; Custo realizado; Quantidade de itens que chegam; Quantidade de itens que encerram;	I-8 I-9

Para cada elemento de dado, a Tabela 3.9 informa na sua ultima coluna a relação dos indicadores para os quais contribui. As outras etapas do processo GQ(I)M foram omitida por estar voltada para situações em que a política de medição tem como foco uma organização específica, o que a torna incompatível com a natureza deste trabalho. A Figura 3.5 ilustra todas as etapas desse capítulo.

O próximo capítulo descreve o conjunto de indicadores como resultado desse método. Esses indicadores foram instanciados em uma base não real, sendo utilizados como uma fonte de dados para nosso estudo de caso do Capítulo 5.

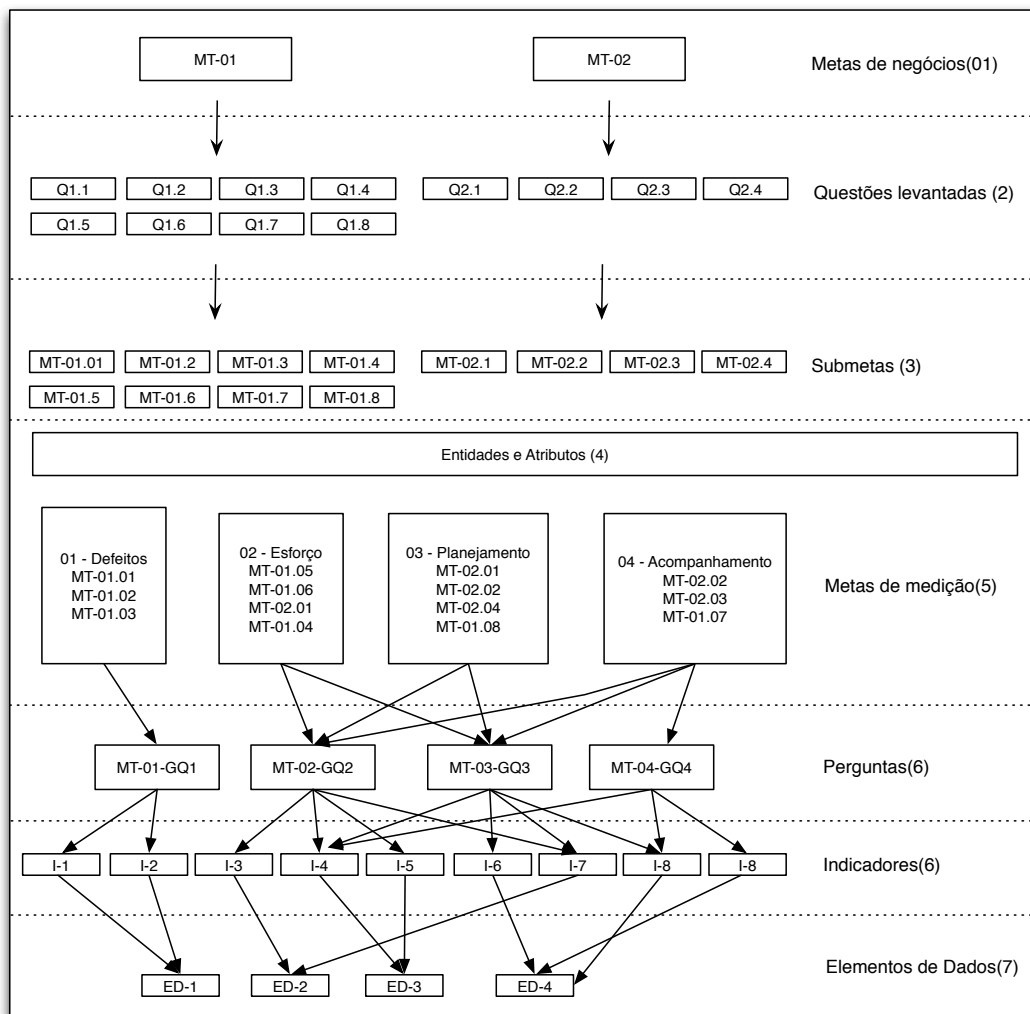


Figure 3.5: Etapas do método de criação dos indicadores

# 4

## Conjunto de Indicadores

A proposta dessa dissertação é utilizar um conjunto de indicadores para auxiliar o gerente de projeto em suas atividades. Neste contexto, o conjunto de indicadores se torna uma ferramenta para gestão de projetos, tornando eficaz na obtenção de evidências e ou informações, fornecendo diferentes dimensões do processo de desenvolvimento de software. Com isso, o gerente de projeto pode tomar decisões para alcançar as metas propostas. Entretanto escolher quais indicadores utilizar não é uma tarefa trivial, sendo este objeto de estudo. Nesse contexto, esta dissertação propõe um conjunto de indicadores para o processo de desenvolvimento de software nas áreas de tempo e custo seguindo as boas praticas do PMBOK. Para a criação dos indicadores foi utilizada a técnica de GQ(I)M demonstrando no Capítulo 3 com algumas adaptações, sendo também uma contribuição deste trabalho. Portanto, o método serviu como uma base para geração desses indicadores. Além disso, foi utilizada a expertise de dois engenheiros de software com experiências em gerência de projetos, um com oito anos e o outro com quatro anos. Os engenheiros de software auxiliaram nas etapas do método utilizado para a criação dos indicadores.

Na Seção 3.2.6 foi elaborado um conjunto de perguntas e são identificados vários indicadores que contribuem para a resposta a essas perguntas. Neste Capítulo iremos descrever e discutir os indicadores criados com as seguintes informações:

- Descrição – breve descrição da informação contida no indicador;
- Ilustração – forma de apresentação do indicador;
- Aplicabilidade – descrição da contribuição trazida pelas informações contidas no indicador;
- Variações possíveis – descrição das diferentes formas de construção do indicador;
- Perguntas - lista o grupo de perguntas que podem ser respondida pelo indicador.

Serão apresentados os 09 indicadores propostos na Seção 3.2.6. As seções 4.1 a 4.9 apresentam a documentação de cada um deles.

## 4.1 Defeitos(Bloqueio) x Pedido

Distribuição por tempo(nesse caso é semanal) do total de pedidos que foi registrado e a quantidade de defeitos do tipo bloqueio. Com isso podemos saber, por exemplo, quantos são os defeitos que estão impactando em outros pedidos. A Figura 5.2 ilustra um exemplo desse indicador.

Defeito(Bloqueio) x Pedido

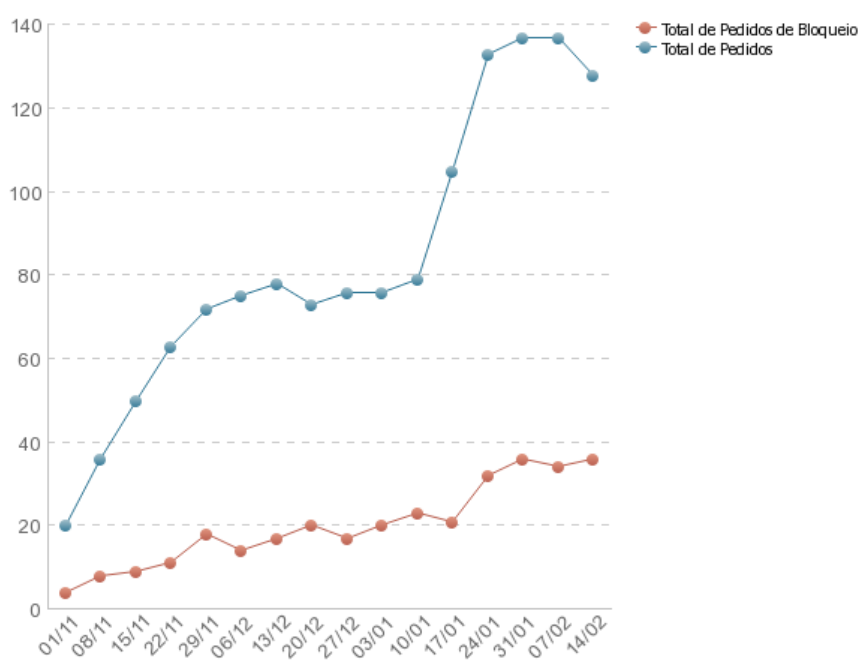


Figure 4.1: Indicador 01 (I-1) - Total de Defeitos(bloqueio) x Pedidos

**Aplicabilidade** O conhecimento da quantidade de defeitos que são de alta prioridade, pois estão impedindo o fechamento de uma determinada solicitação. A mitigação e o acompanhamento desses defeitos pode ser útil para guiar atividades de melhoria de processo que visam à redução dos defeitos e à detecção precoce dos mesmos. Além disso, a linha do total de defeitos(bloqueio) deve fazer uma curva para baixo. Uma curva para cima indicaria que o trabalho em um número crescente de requisitos está sendo atrasado.

**Variações possíveis** Esse indicador pode sofrer alterações, como:

1. Identificar Todos os defeitos registrados;
2. Identificar Todos os defeitos registrado por tipo;

3. Mudar a dimensão tempo para mês ou dia.

Para cada uma das situações acima, há ainda duas opções: considerar defeitos pertencentes a um projeto individual, ou considerar um conjunto de projetos.

**Perguntas** - MT-0.1-GQ1.

## 4.2 Defeito por Gravidade x Estado do Defeito

Defeito tem que ser acompanhado, pois pode atrasar o andamento do projeto ou refletir a qualidade do processo e ou produto. A Figura 4.2 ilustra a relação de quantidade de defeitos por gravidade e para cada gravidade a quantidade por estado.

### Defeitos por Gravidade x Estado do Defeito

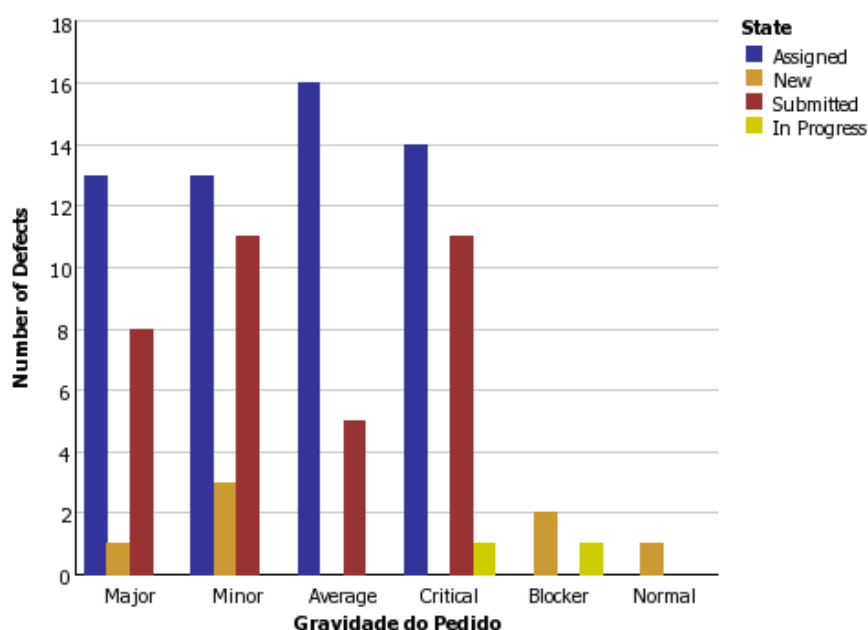


Figure 4.2: Indicador 02 (I-2) - Defeito por Gravidade x Estado do Defeito

**Aplicabilidade** Durante o ciclo de um projeto, o número de defeitos com gravidades de alto nível deve diminuir. As maiorias dos defeitos devem ser resolvidos ou encerrados. Um número substancial de defeitos novos ou em andamento pode indicar um problema com o produto/processo ou ser resultado de maior atividade de teste.

**Variações possíveis** Considerar todos os defeitos registrados (resolvidos ou encerrado) e ainda duas alternativas: considerar defeitos pertencentes a um projeto individual, ou considerar um conjunto de projetos.



Perguntas - MT-0.1-GQ1.

### 4.3 Atividades por pessoa x atividades por estado

A relação atividade e estado por pessoa demonstra o quanto estamos distribuindo nossa atividade assim como qual a sobrecarga no processo. A Figura 4.3 ilustra a distribuição de atividades por pessoa e mostra a quantidade de atividades que está em certo estado.

#### Atividades por Pessoa

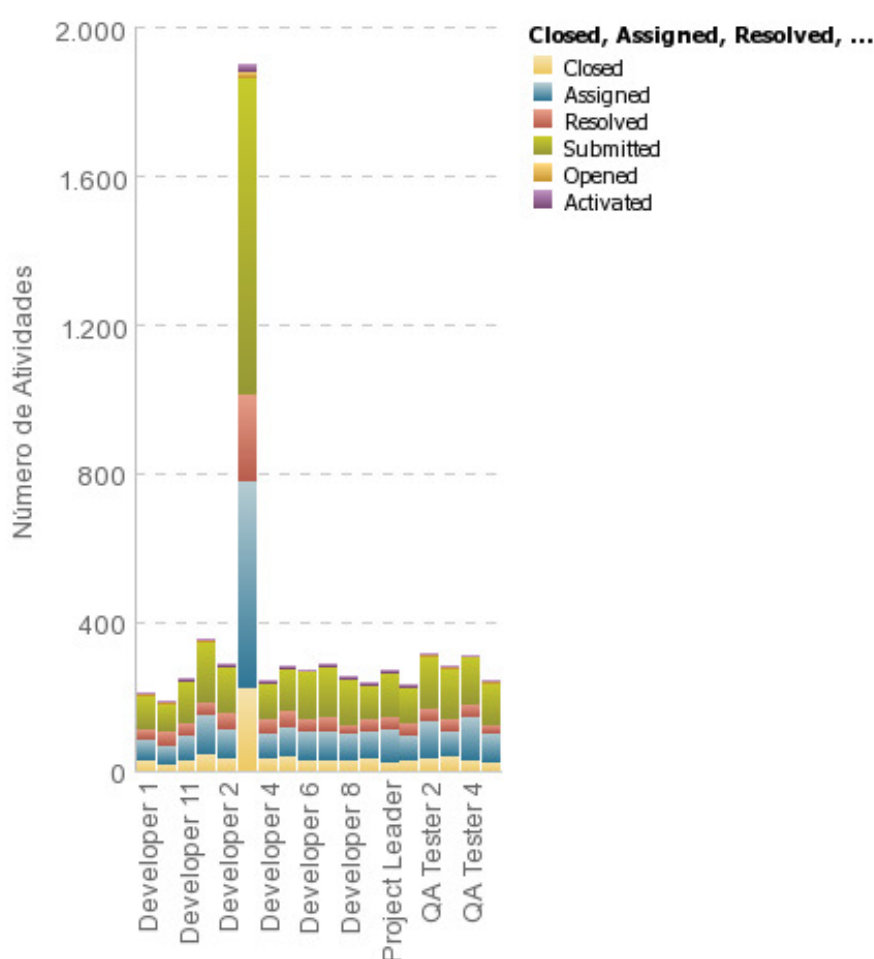


Figure 4.3: Indicador 03 (I-3) - Atividades por pessoa x atividade por estado

**Aplicabilidade** Como as atividades representam geralmente uma tarefa ou outra unidade de trabalho relacionada, o número por pessoa dá uma ideia geral do trabalho que cada pessoa está fazendo e o estado atual das atividades. Essas atividades não tem o fator de complexidade ou a quantia real de trabalho necessária por qualquer atividade. Esse indicador serve para acompanhar quais atividades estão relacionadas para qual pessoa e qual o estado dela.

**Variações possíveis** O indicador pode ser construído para representar a atividades sob diferentes aspectos:

1. Atividades total (considerando toda a atividade sem o estado da mesma);
2. Atividade por tamanho (considera-se o tamanho de cada atividade);
3. Atividade por projeto e conjunto de projetos. A atividade pode ser calculada em função do tamanho do produto (em Pontos de Função) ou do tamanho de algum de seus artefatos (em alguma unidade de medida previamente definida).

**Perguntas** - MT-0.1-GQ2.

## 4.4 Esforço Planejado x Esforço Real (Projeto)

Esse indicador ilustra como estamos perto do que foi planejado para o esforço. A Figura 4.4 faz uma comparação do que foi planejado(esforço) por o que foi realizada(esforço). A unidade de medida utilizada é quantidade de linhas de códigos, porém essa medida pode ser alterada de acordo com a definição do gerente de projeto.

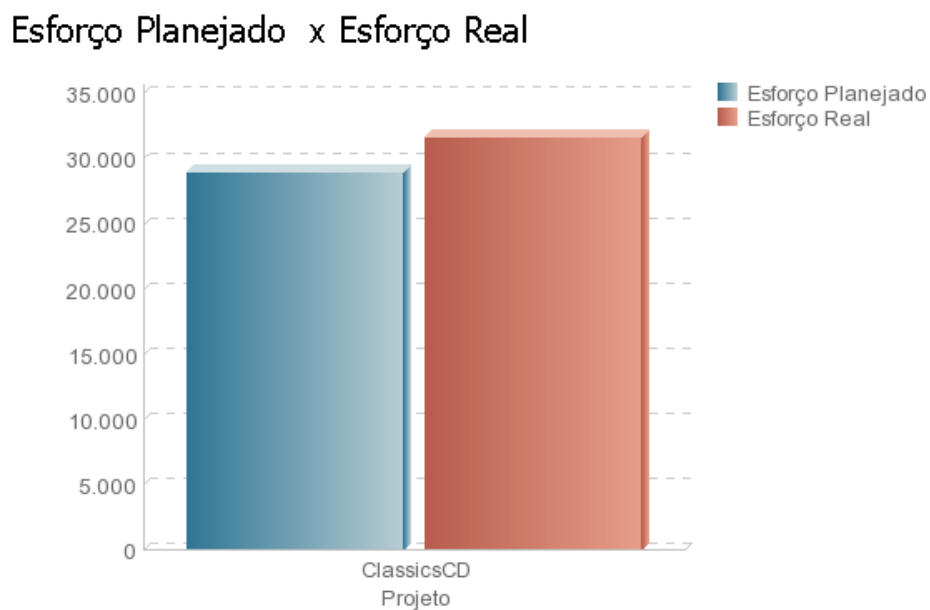


Figure 4.4: Indicador 04 (I-4) - Esforço Planejado x Esforço Real

**Aplicabilidade** São mostradas duas métricas de esforço por projeto. Esforço planejado é o valor do plano mais recente. Esforço real é a quantidade de trabalho realmente executado. O esforço real não deve diferir substancialmente do plano, uma grande diferença pode indicar que os planos subestimaram a quantidade de trabalho ou que o trabalho é mais difícil do que o previsto.

**Variações possíveis** O indicador pode ser construído para representar o Esforço Planejado por linha base (considerando o esforço planejado da primeira linha de base salva ou assim em diante). Além disso, a produtividade pode ser calculada em função do tamanho do produto (em Pontos de Função) ou do tamanho de algum de seus artefatos (em alguma unidade de medida previamente definida).

**Perguntas** - MT-0.1-GQ2, MT-0.2-GQ1, MT-0.2-GQ2.

## 4.5 Esforço Planejado da Tarefa x Esforço Real da Tarefa

No indicador I-4 fizemos uma comparação macro. Entretanto podemos querer acompanhar de maneira mais detalhada. Na Figura 4.5 fazemos o acompanhamento do que foi planejado para uma tarefa e o que foi realizado da mesma.

Esforço Planejado da Tarefa X Esforço Real da Tarefa

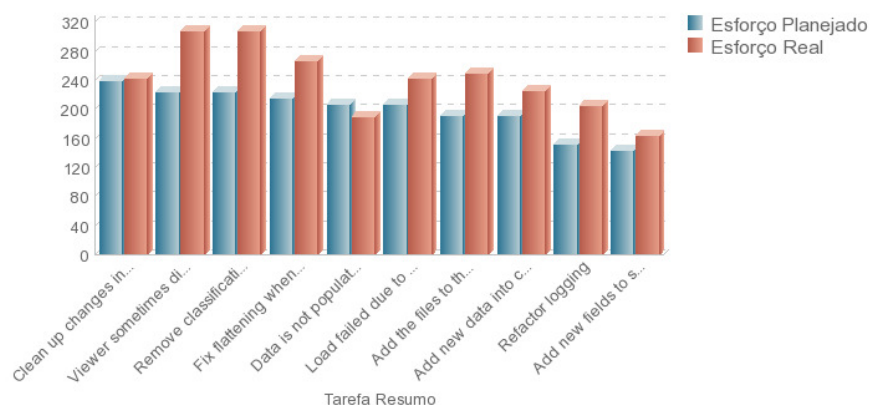


Figure 4.5: Indicador 05 (I-5) - Esforço Planejado da Tarefa x Esforço Real da Tarefa

**Aplicabilidade** Este relatório compara os esforços planejado e real das tarefas. Embora geralmente haja diferenças, grandes discrepâncias podem indicar planejamento impreciso, mudanças no escopo ou nos requisitos do projeto, complexidade maior do que a esperada, ou algum outro fator.

**Variações possíveis** O indicador pode ser construído para representar o esforço das tarefas incluindo também o esforço planejado por linha base (considerando o esforço planejado da primeira linha de base salva ou assim em diante). Além disso, a produtividade pode ser calculada em função do tamanho do produto (em Pontos de Função) ou do tamanho de algum de seus artefatos (em alguma unidade de medida previamente definida).

**Perguntas** - MT-0.1-GQ2.

## 4.6 Tendências do item de escopo

Saber a tendência se faz necessário para acompanhar o planejamento, por exemplo, muitos pedidos de mudança podem mudar o planejamento. A Figura 4.6 mostra o número de itens de escopo que podem causar um impacto no planejamento.

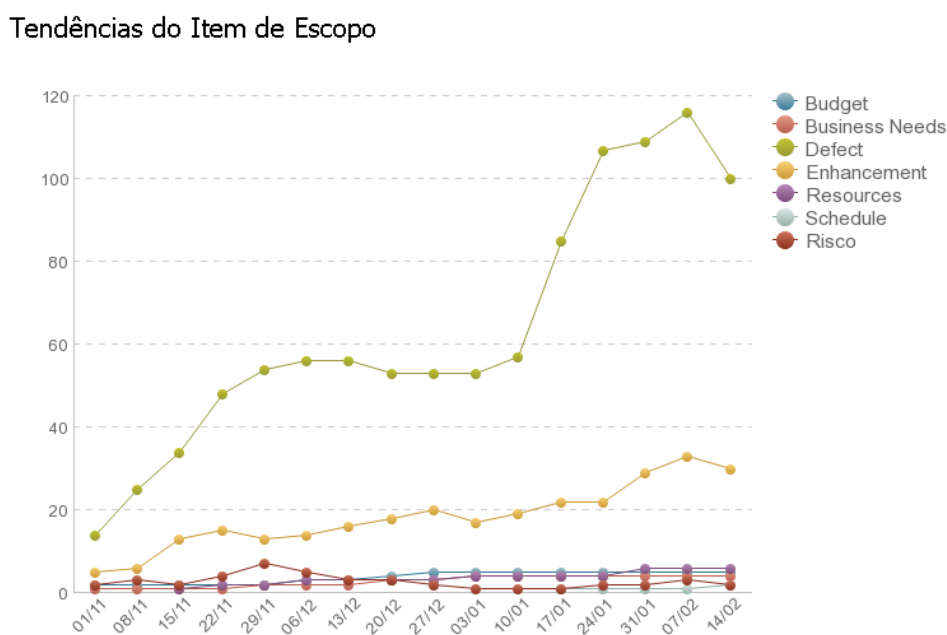


Figure 4.6: Indicador 06 (I-6) - Tendências do item de escopo

**Aplicabilidade** Os itens de escopo são as incertezas de um projeto, podendo afetar o planejamento do projeto. Estão incluídos riscos, problemas, pedidos de mudança, defeitos e aprimoramentos. Em geral, o número de itens de escopo pendentes aumenta no início de um projeto e mostra uma tendência descendente com o progresso do projeto. Se a tendência dos itens de escopo pendentes não baixar, a conclusão do projeto está em risco.

**Variações possíveis** Podem sofrer alteração no tipo de item de escopo e também podemos selecionar uma um projeto específico ou vários projetos.

**Perguntas** - MT-0.2-GQ1.

## 4.7 Produtividade

É importante entender as contribuições de código relativas que as pessoas fazem em um projeto. Nesse gráfico da Figura 4.7 é utilizando a métrica de contribuição por linha de código.

### Produtividade

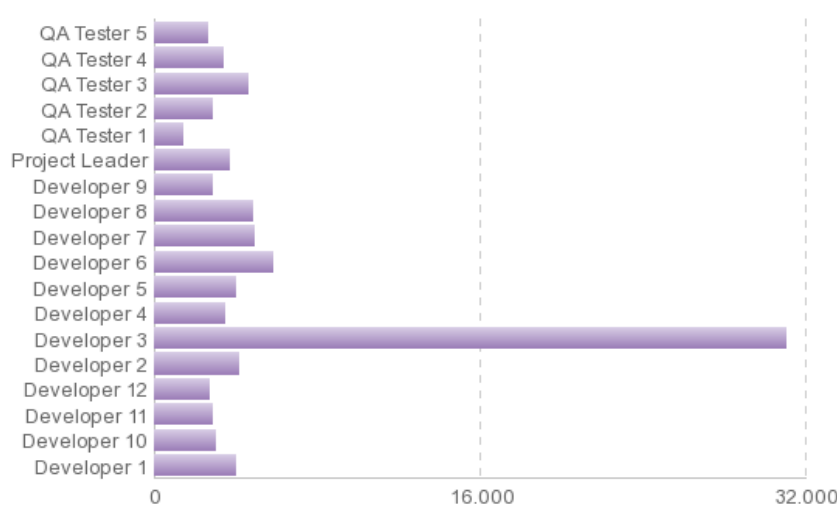


Figure 4.7: Indicador 07 (I-7) - Produtividade

**Aplicabilidade** Geralmente o trabalho deve ser distribuído proporcionalmente através da equipe. Se uma pessoa receber significativamente mais ou menos trabalhos do que os outros, provavelmente há um problema de planejamento de projeto. Porém, esse gráfico não leva em conta fatores como outras responsabilidades, dificuldade relativa do trabalho, período de férias, etc. Os dados deste gráfico geralmente não devem ser usados para fins de avaliação do funcionário.

**Variações possíveis** O indicador pode ser construído para representar a produtividade total (considerando todo o esforço dedicado aos projetos) ou em um projeto específico ou um atividade específica. Além disso, a produtividade pode ser calculada em função do tamanho do produto (em Pontos de Função) ou do tamanho de algum de seus artefatos (em alguma unidade de medida previamente definida). Um exemplo de produtividade em função de um artefato seria a quantidade de linhas de código produzida por pessoa-mês (nesse caso, o artefato seria o código-fonte e a unidade de medida seria o número de linhas de código).

**Perguntas** - MT-0.2-GQ1.

## 4.8 Custo do Projeto

As informações de custo real são essenciais para controlar o progresso de um projeto e se vamos precisar alavancar novos recursos caso passe o custo planejado. A Figura 4.8 mostra ambos os custos planejado e real.

## Custo do Projeto

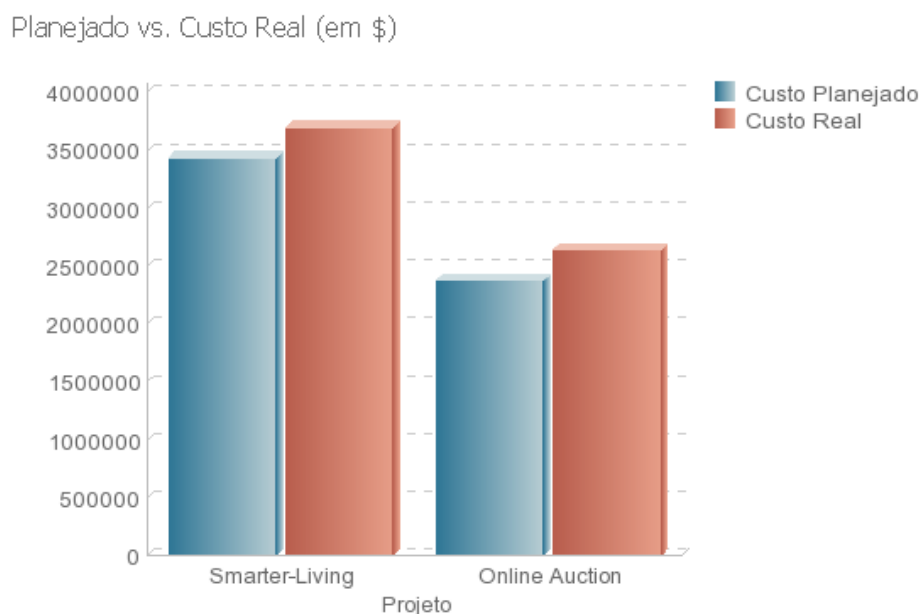


Figure 4.8: Indicador 08 (I-8) - Custo do Projeto

**Aplicabilidade** Embora seja raro que um projeto cumpra exatamente seu orçamento planejado, é necessário entender a diferença substancial entre eles. Precisamos acompanhar se essa diferença está com uma discrepância grande.

**Variações possíveis** O indicador pode ser construído para representar o custo da Planejado por linha base (considerando o esforço planejado da primeira linha de base salva assim por diante).

**Perguntas** - MT-0.2-GQ1, MT-0.2-GQ1.

## 4.9 Taxas de Chegada e Encerramento do Item de Escopo

Saber como se comporta a relação de chegada de item e seu encerramento se faz necessário para saber como anda a produção, por exemplo. A Figura 4.9 ilustra quantos itens de escopo chegaram e foram encerrados em um intervalo de tempo.

Taxas de Chegada e Encerramento do Item de Escopo

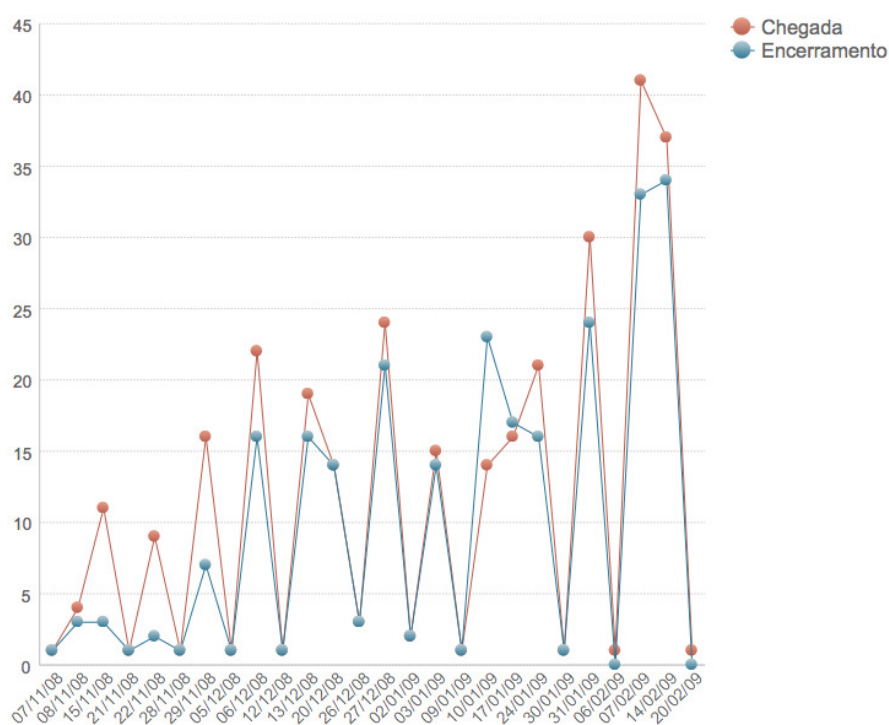


Figure 4.9: Indicador 09 (I-9) - Taxas de Chegada e Encerramento do Item de Escopo

**Aplicabilidade** No início de um projeto, a linha de chegada pode exceder a linha de encerramento. Com a estabilização do projeto, a linha de encerramento deve igualar-se à linha de chegada. Um grande intervalo entre a linha de chegada e a linha de encerramento significa que vários itens de escopo estão pendentes, o que não é desejável, especialmente perto do fim do ciclo do projeto.

**Variações possíveis** Há ainda duas alternativas: considerar escopos pertencentes a um projeto individual, ou considerar um conjunto de projetos.

**Perguntas** - MT-0.2-GQ2.

No próximo Capítulo iremos descrever um estudo de caso em que utilizamos esses indicadores em um ambiente real.

# 5

## Estudo de Caso

O conjunto de indicadores do Capítulo 4 proposto nesta dissertação foi usado no contexto de um estudo de caso em um ambiente real. Embora exista uma tendência na engenharia de software para a realização de estudos empíricos, abordagens que utilizam estudos de caso estão em constante crescimento (Runeson & Host, 2008). Ainda segundo Runeson & Host (2008), na engenharia de software, o método de estudo de caso é adequado em diversas situações como, por exemplo, quando os objetos de estudo são fenômenos contemporâneos, ou ainda quando tais objetos são difíceis de serem analisados de forma isolada. Como o estudo foi realizado em um ambiente real e, portanto, com o envolvimento de diversas pessoas no contexto de um processo de desenvolvimento de software real de uma organização, a execução de um experimento controlado se tornaria inviável.

O estudo de caso seguirá uma abordagem qualitativa, pois fornece uma descrição rica e profunda para verificar se o conjunto de indicadores realmente auxilia as atividades do gerente de projeto. Runeson & Host (2008) propõem uma estrutura para o relato de estudos de caso em engenharia de software. A estrutura basicamente é composta por: ambiente, problema, coleta de dados e análise de dados coletados. A seguir, tal estrutura é adaptada e utilizada para o relato do estudo de caso realizado nesta dissertação.

### 5.1 Ambiente

A empresa em estudo realiza atividades relacionadas ao desenvolvimento de software. Como o estudo envolve a utilização de informações confidenciais da organização, omitiremos o seu nome, bem como informações que possam identificá-la. Tal organização foi escolhida, pois ela atende a todos os requisitos para a utilização dos indicadores estudados e, portanto, a utilização das ferramentas que auxiliem no processo de desenvolvimento de software torna-se essencial. Como por exemplo, IBM Rational Team Concert (RTC) que auxilia na gestão de item de trabalho, controle de versão e integração contínua.



Durante o projeto, com o auxílio desse tipo de ferramenta, é gerada uma grande quantidade de dados que refletem o andamento do processo de desenvolvimento da solução. Estes dados ficam armazenados nas bases de dados de cada uma dessas ferramentas. Os dados coletados devem ser armazenados de forma estruturada, preferencialmente em uma base de dados única e que especifique com precisão as entidades envolvidas, seus atributos e relacionamentos. Essa base servirá para a aplicação dos indicadores e permitirá o acompanhamento e controle dos dados do projeto. Para a unificação das bases de dados foi utilizada a ferramenta IBM Cognos que dá suporte aos processos de extração de bases distintas, tratamento de dados e carga para o modelo único citado. A Figura 5.1 ilustra esse ambiente. Para o servidor de relatório e painéis de controle dos indicadores foi utilizado o IBM Rational Insight.

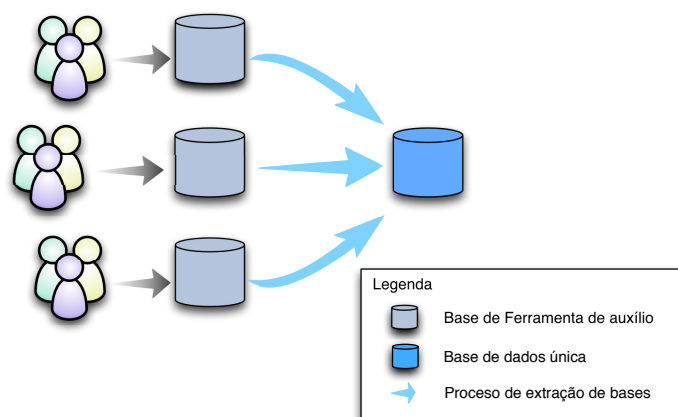


Figure 5.1: Ambiente do estudo de caso

## 5.2 Problema

Grande parte das organizações possui o papel do gerente de projeto. A empresa estudada também possui esse papel. Dentre as diversas atividades do gerente de projeto, podemos citar a avaliação e controle do desempenho dos projetos assim como a tomada de decisão. Para a realização de tais atividades, é necessária a coleta e uso adequado de informações que auxiliem nas tomadas de decisões no sentido de atender as expectativas do projeto. Nesse contexto este trabalho propõe um conjunto de indicadores para auxiliar o gerente de projetos em suas atividades nas áreas de tempo e custo. Com isso, surge outra preocupação que é o risco que envolve o tempo e o custo, pois esses são dois dos principais fatores relacionados nas pesquisas (Budzier, 2012; Jones, 2006) que tendem a levar projetos ao fracasso.

### 5.3 Passo e explicação do método

No capítulo 3 foi apresentado o método utilizado para a criação do conjunto de indicadores. Nesta subseção iremos detalhar como esse método foi utilizado. Como o método se baseia na abordagem GQ(IM) (Park et al., 1996) seguiremos as sete etapas descritas no Capítulo 3.

Na primeira etapa identificamos se as metas propostas na Seção 3.2.1 são as mesmas metas da empresa. De fato, com as pesquisas citadas e com declarações dos gerentes de projetos essas duas dimensões de custo e tempo são importantes para o acompanhamento do projeto. Com isso, obter boa produtividade e cumprir os compromissos de prazo e custo são metas da empresa. Porém, deixamos claro que caso eles queiram acrescentar outras metas poderão utilizar nosso trabalho como referência de inserção de novos indicadores. Entretanto, nosso estudo não se concentrou em outras metas além das supracitadas.

Com as metas alinhadas, o próximo passo era descobrir o que o gerente de projetos gostaria de saber sobre o projeto utilizando os indicadores analisados. Para isso, repassamos cada uma das questões que se encontram na Seção 3.2.2 ao gerente de projetos. Em cada uma das tabelas da Seção 3.2.2 estão perguntas relacionadas ao que desejamos saber ou aprender sobre o projeto. Para cada meta existe uma tabela agrupando as perguntas por tema. A escolha dessas perguntas se deu pelo fato de que as suas repostas podem ajudar na obtenção das metas desejadas. As perguntas geraram questionamentos dos gerentes de projetos como, por exemplo, a forma com que tais perguntas ajudariam na obtenção das metas. Para isso foi explicado que as respostas são os demonstrativos do andamento do projeto e que com essas informações o gerente de projetos pode tomar suas decisões. Com isso, a chance de acertar na tomada de decisão para alcançar as metas são maiores, pois possuem um maior número de informações precisas sobre o andamento do projeto auxiliando assim o gerente de projeto.

No terceiro passo, descrito na Seção 3.2.3, explicamos cada uma das submetas. O entendimento dessas submetas se faz necessário, pois é nelas que serão abordadas as ações gerenciais e técnicas como, por exemplo: MET-01.01 – Diminuir o número de defeitos durante todo o processo de desenvolvimento. Se a Meta-01 é “obter boa produtividade”, então queremos saber como é a distribuição dos defeitos e qual é sua gravidade ou como o gerente de projetos pode tomar alguma ação para diminuir a quantidade de defeito e, portanto, aumentando a produtividade.

Após a explicação das submetas, demonstramos como as perguntas da Seção 3.2.2 pode ajudar a identificar as entidades e os atributos relacionados com as suas submetas (Seção 3.2.4). Um exemplo disso seria a identificação de conjuntos de defeitos, defeitos por tipo e gravidade como entidades e atributos, respectivamente, sendo que tais entidades e atributos façam parte da pergunta de como estão distribuídos os defeitos e qual é a gravidade desse defeito. Para isso foi repassado, juntamente com o gerente de projetos, como são identificados os defeitos e como se classifica os mesmos no processo da empresa. Foi levantada a hipótese

de que a mudança na classificação dos defeitos poderia afetar o indicador. Deixamos claro que a mudança da classificação não afetaria o indicador. Essa mudança adicionaria um novo tipo de defeito ao relatório. Com a identificação das entidades e atributos passamos para o próximo passo que era a efetivação dos objetivos de medição.

Na Seção 3.2.5, Tabela 3.7, temos quatro objetos de interesse: defeito, esforço, planejamento e acompanhamento. Cada objeto de interesse possui uma proposta e uma perspectiva e está relacionado a uma submeta. Explicamos cada um desses objetos de interesse e fizemos um apontamento com as submetas para identificar como surgiram as propostas e perspectivas. Por exemplo, no objeto de interesse Defeito temos uma proposta de analisar a distribuição de defeito com a perspectiva de identificar os pontos que merecem uma maior atenção, estando relacionado com as MET-01.01, MET-01.02 e MET-01.03. Com isso, o objetivo de medição ficou mais detalhado e a equipe entende de forma mais clara o propósito da realização das medições. Após esclarecer sobre os objetivos da realização de medições, aplicamos os conjuntos de indicadores do Capítulo 4. O resultado dessa aplicação será omitido por não termos acesso a base completa, por motivos que os indicadores refletem o estado do projeto em termos de custo e prazo. Portanto, a empresa em estudo não disponibilizou a base completa e sim uma base simplificada dos seus projetos. Entretanto, o objetivo do estudo de caso é fazer uma análise qualitativa. Pois, queremos identificar se o conjunto de indicadores auxiliam o gerente de projeto. Portanto, vamos nos concentrar em realizar entrevista, para coletar informações, sobre a aplicação dos indicadores. Essa será nossa coleta de dados. Porém no Capítulo 4 já possui uma ilustração e descrição de todos os indicadores utilizados. Como o estudo de caso segue uma linha explicativa e qualitativa, vamos focar no resultado da utilização dos indicadores. A Figura 5.2 descreve todos os passos do estudo de caso até chegar a coleta de dados (entrevistas).

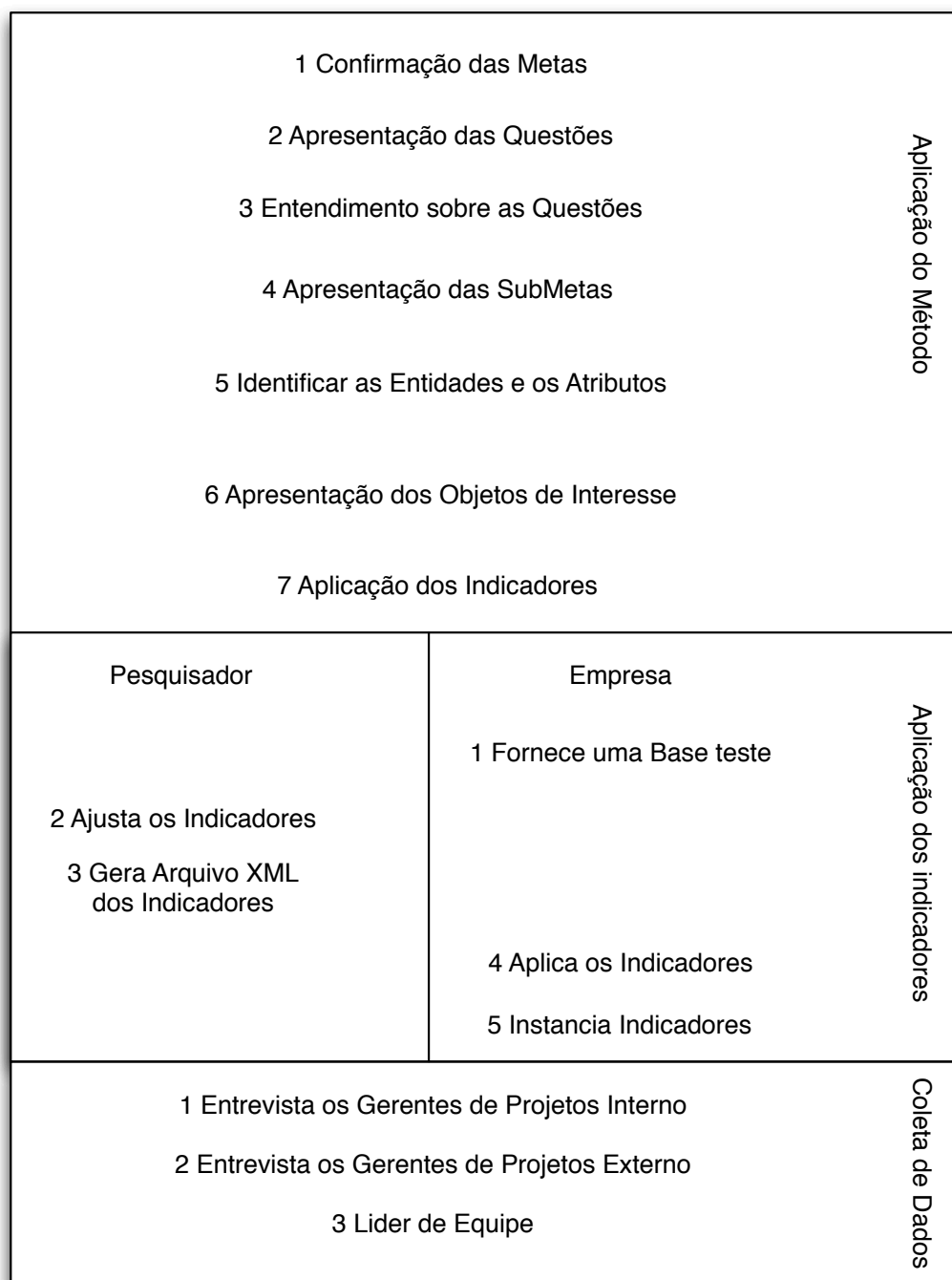


Figure 5.2: Ambiente do estudo de caso

## 5.4 Coleta de Dados

A coleta de dados foi realizada através de entrevistas de primeiro grau, no qual o pesquisador tem contato direto com os entrevistados e ambiente. Foram realizadas entrevistas com os gerentes de projetos, bem como com indivíduos que realizam outros papéis dentro do processo. Para manter a triangulação sugerida por [Runeson & Host \(2008\)](#), em

posse dos indicadores que também foram analisados, fizemos uma entrevista com um gerente de projetos externo e que não possuía vínculo com a empresa. Ao todo foram 5 entrevistas, 2 gerente de projetos internos, 1 gerente externo, 1 líder de equipe e 1 desenvolvedor. Essas entrevistas, que são nossos dados, estão no Anexo A. O objetivo da entrevista era saber se de posse de tais indicadores possuímos uma visão do projeto. Com isso, auxiliar o gerente de projeto a tomar decisões para alcançar as metas. O modelo de entrevista utilizado foi o semiestruturado, no qual as perguntas não são necessariamente feitas na mesma ordem em que são listadas. Ou seja, o desenvolvimento da conversa na entrevista pode decidir qual a ordem em que as diferentes questões são tratadas. Além disso, entrevistas semiestruturadas permitem improvisação e exploração dos objetos estudados. As perguntas abordadas foram as mesmas da Tabela 6.1 do Capítulo 3, além das seguintes perguntas acrescidas:

- Os indicadores ajudaram na tomada de decisão?
- Sem os indicadores as decisões podem ser tomadas?
- Como você avalia o conjunto de indicadores?

Vale ressaltar que as perguntas utilizadas dependiam do papel do entrevistado na empresa. Só no papel do gerente de projetos que utilizamos todas as perguntas.

## 5.5 Análise de dados coletados

Após a coleta de dados foram identificados alguns pontos relevantes à utilização do conjunto de indicadores proposto. Os pontos relevantes são descritos a seguir.

Sobre as perguntas da Tabela 6.1, todos os entrevistados, em posse dos indicadores, conseguiram responder de forma parecida para não dizer igual. Por exemplo, na pergunta relacionada à quantidade de defeitos abertos no projeto, o indicador I-1 da Seção 4.1 demonstra a quantidade de defeitos do tipo bloqueio. Este tipo de defeito se caracteriza por impedir a continuidade de algum outro pedido. Esse indicador demonstra quantos pedidos estão chegando e quantos têm defeitos do tipo bloqueio. Já o indicador I-2 da Seção 4.2 demonstra a quantidade de defeito por estado e gravidade. Com esses indicadores podemos saber a quantidade de defeitos abertos no projeto e quais deles devem ser acompanhados cuidadosamente para que o projeto acabe no tempo previsto. Dessa forma, em posse dos indicadores, o gerente soube responder a pergunta. Esse exemplo serve de modelo para todas as outras perguntas na qual o gerente soube responder. Isso se dá pelo fato do gerente estar com as informações necessárias, que são os indicadores, para responder às perguntas. Entretanto o gerente de projetos externos fez alguns questionamentos. Por exemplo, o que a empresa considera como defeito e como o classifica. De fato não tinha como ele saber, pois cada empresa adota uma forma de classificar esse defeito, bem como o processo de desenvolvimento de software. Outro exemplo seria a medida da produção, ficando a critério do cliente escolher qual adotar,

por exemplo, por ponto de função ou linha de código. Porém, o foco deste trabalho não é forma de medir produtividade e sim a utilização do conjunto de indicadores com auxílio à tomada de decisões. Além dessas perguntas os gerentes de projetos responderam a outras três perguntas fora do escopo da Tabela 6.1 do Capítulo 3. As perguntas são:

### **Os indicadores ajudaram na tomada de decisão?**

De acordo com as respostas dadas pelos gerentes de projetos, a utilização dos indicadores ajudaram nas tomadas de decisões. Entretanto, um dos entrevistados ressaltou que os indicadores por si só não servem de base para a tomada de decisões. De fato o conjunto de indicadores serve como auxílio para as tomadas de decisão, pois conseguem agregar um grande número de informações que reflete o andamento do projeto.

### **Sem os indicadores as decisões podem ser tomadas?**

As decisões são tomadas sem os indicadores, porém a falta de informação leva a decisões menos eficazes. Os indicadores geram uma vantagem de se ter várias informações reunidas e modeladas. Com isso, o controle e monitoramento se torna uma tarefa mais simples, pois a necessidade de se garimpar as informações em repositórios diferentes, e que torna a tarefa árdua e custosa, é eliminada.

### **Como você avalia o conjunto de indicadores?**

Os gerentes de projetos da organização analisada levantaram alguns pontos importantes para a avaliação do conjunto de indicadores. Os pontos positivos levantados foram: informação centralizada, foco em áreas de interesses (risco, tempo, custo), maior controle e monitoramento do projeto, aplicáveis em quaisquer projetos de desenvolvimento de software da empresa. Os pontos negativos citados foram: informações macro, não apontam a solução e não são focadas em nenhuma certificação.

O gerente externo avaliou como pontos positivos os mesmos dos gerentes de projetos da organização analisada, porém não citou se o conjunto de indicadores é “aplicável a quaisquer projetos de desenvolvimento de software da empresa”. Entretanto, foi deixado claro que esses indicadores servem com modelo para projetos da sua empresa caso possam sofrer alterações como, por exemplo, a produtividade ser medida por pontos de função. Como ponto negativo foi salientado a não identificação de algumas classificações como, por exemplo, as classificações do defeito. Como sugestão o gerente pediu para deixar claro nos indicadores como são extraídos tais defeitos, assim como para os outros indicadores.

# 6

## Revisão

Até o momento, a maioria dos trabalhos estudados no contexto do processo de desenvolvimento de software e criação de indicadores se basearam em uma abordagem diferente da proposta por este trabalho (Monteiro & de Oliveira, 2011; McQuighan & Hammell, 2011; Chandrasekaran & Kumar, 2012; Islam et al., 2012; Subramanyam & Krishnan, 2003; Xie et al., 2009). Por exemplo, Monteiro & de Oliveira (2011) propõem uma lista de indicadores para apoiar na análise de desempenho do processo de desenvolvimento de Software. Porém, seu intuito é auxiliar os gestores na realização da análise de desempenho de processo do Capability Maturity Model Integration for Development (CMMI-DEV). Em outro contexto, Islam et al. (2012) apresenta uma revisão sistemática da literatura sobre avaliação e mensuração de melhoria de processo de software(SPI), com objetivo de identificar e caracterizar as estratégias de avaliações e métricas utilizadas para avaliar o impacto em diferentes iniciativas SPI. A Tabela 6.1 elenca os trabalhos relacionados com suas características.

Table 6.1: Trabalhos Relacionados

Autor	Abordagem	Objetivo	Forma
Islam et al. (2012)	SPI	Visão Geral	Métricas
Subramanyam & Krishnan (2003)	CK	Produtividade e De-feito	Métricas
Monteiro & de Oliveira (2011)	CMMI-DEV	Nível 4	Indicadores
McQuighan & Hammell (2011)	PMI	Escopo	Indicadores
Chandrasekaran & Kumar (2012)	PMI	AVA/OTB	Indicadores
Trabalho Proposto	PMI	Custo e Tempo	Indicadores

Para cada trabalho relacionado, verificamos a existência de duas diretrizes: as que estudam métricas e outras que usam essas métricas para construir indicadores, como forma de atingir o objetivo.

## 6.1 Trabalhos objetivando Métricas

Os objetivos desses trabalhos são de identificar as métricas e/ou comprovar os benefícios de sua utilização no processo de desenvolvimento de software. Neste sentido, [Islam et al. \(2012\)](#) apresentou uma revisão sistemática da literatura sobre avaliação e mensuração de melhoria de processo de software (SPI). O objetivo é identificar e caracterizar as estratégias de avaliações e métricas utilizadas para avaliar o impacto em diferentes iniciativas SPI. Tal revisão sistemática incluiu 148 artigos publicados entre 1991 a 2008.

[Islam et al. \(2012\)](#) justifica que atividades realizadas através de pessoas, possibilitam o comportamento e desempenho indesejável ou inesperado das atividades, sendo assim, a utilização de métodos de avaliações para o processo de desenvolvimento de software se torna fundamental. Para isso, o processo de desenvolvimento de software precisa ser avaliado e aperfeiçoado. Melhoria de Processo de Software (SPI) envolve a avaliação e aprimoramento do processo através de praticas do desenvolvimento de software. Ainda segundo [Islam et al. \(2012\)](#), a medição do processo de software é um esforço essencial para alcançar o desempenho esperado. Com isso, é necessário obter a capacidade para assegurar que o processo atenda os artefatos e os seus requisitos de qualidade especificados.

De acordo com [Islam et al. \(2012\)](#), as informações obtidas pela medição e a avaliação dos efeitos, proporcionam pelo menos duas vantagens, quais sejam: primeiro, transparece o resultado, motivando e justificando o esforço colocado em um iniciativa. Além disso, permite a avaliação de estratégias e táticas. Ao mesmo tempo, é difícil estabelecer e implementar um programa de medição que forneça informações relevantes e válidas para tomadas de decisões. Foi identificado por [Islam et al. \(2012\)](#) que o objetivo da melhoria do processo de software é aumentar a qualidade do produto, mas também reduzir o tempo e custos de produção.

Como resultado do trabalho de [Islam et al. \(2012\)](#), foram identificados que publicações em relação aos métodos de pesquisa concentravam-se na maior parte em estudo de caso (45 por cento) e o restante é dividido em relatórios da indústria (36 por cento), experiências (5 por cento), pesquisas (4 por cento), pesquisa-ação (1 por cento), e uma combinação de pesquisa-ação e experimentação (1 por cento). Também foi observado que a falta de uma adequada descrição da metodologia de pesquisa aplicada, impediu uma categorização (8 por cento). Já na distribuição das iniciativas do SPI a predominância é em “Framework” (61 por cento), seguido por “Práticas” (20 por cento) e “Ferramentas” (6 por cento).

Em relação as estratégias de avaliações identificadas, a “pré-pós comparação” é predominante. Nesta, se compara o momento anterior a implantação e os resultados alcançados com



a implantação de um SPI. Porém algumas dessas estratégias de avaliação foram genéricas e em diferentes organizações aplicavam o método para medir indicadores de sucesso com base em suas necessidades e contextos. Assim, era necessário criar um linha base para aplicar o “Pré-Pós Comparação” com o objetivo de atingir os indicadores de sucesso. Em relação a análise estatística e controle estatísticos, foi identificado a utilização de estatística descritiva, onde os dados são resumidos numericamente ou graficamente. Uma boa parte do trabalhos identificados por [Islam et al. \(2012\)](#), relata a importância do custo-benefício da implantação de uma iniciativa de acompanhamento do processo do desenvolvimento de software.

Outro objetivo da pesquisa de [Islam et al. \(2012\)](#) consiste em identificar as métricas utilizadas e os indicadores de sucesso nas avaliações SPI. Foi identificado “Qualidade do processo” (39 por cento), “precisão de estimativas” (38 por cento), “Produtividade” (35 por cento) e “Qualidade do produto”(32 por cento). Sobre perspectiva de medição, foi identificado que “Projeto” (66 por cento) é predominante, seguido pelo “Projeto e Produto” (20 por cento) e por “Projeto, Produto e Organização” (5 por cento). Esse números mostram que a medição e avaliação em nível de Projeto é a abordagem mais comum.

Nesse contexto, o trabalho apresentado nesta dissertação tem propriedades como levantadas por [Islam et al. \(2012\)](#). Esta dissertação tem como método de pesquisa estudo de caso, apontando com a maior participação por [Islam et al. \(2012\)](#). Já em termo de iniciativa, temos a proposta de boas Práticas do [PMBOK \(2004\)](#). [Islam et al. \(2012\)](#) também cita “Práticas” como uma iniciativa, porém, não discrimina quais são as práticas. Em relação as estratégias de avaliação, verifica-se a “Pré-Pós Comparação”, citada por [Islam et al. \(2012\)](#) como predominante. De igual modo, em nosso estudo de caso realizamos uma entrevista antes e depois da implantação dos indicadores. Esse tipo de pesquisa, qualitativa, tenta identificar quais foram os benefícios da implantação dos indicadores, através de perguntas abertas. O principal objetivo é saber a opinião do entrevistado. Uma diferença identificada do nosso trabalho e dos trabalhos relacionados na revisão de literatura proposta por [Islam et al. \(2012\)](#), é que o nosso objetivo é auxiliar o gerente de projeto. Além disso, através da implantação dos indicadores podemos ter resultado também na melhoria do processo de software. Pois, estamos extraindo informações do processo de desenvolvimento de software, com intuito de melhorar metas que estão relacionadas com o processo.

Outra pesquisa com o objetivo de identificar métricas para processo de desenvolvimento de software foi a de [Subramanyam & Krishnan \(2003\)](#). Esta pesquisa estuda a relação entre um subconjunto de métricas de Chidamber e Kemerer (CK) e a qualidade de software Orientado a Objeto (OO) medido em termos de defeitos, especificamente aqueles relatados por clientes e os identificados durante os testes de aceitação do cliente.

[Subramanyam & Krishnan \(2003\)](#) oferecem provas da associação entre um subconjunto de métricas CK e defeitos. Para isso, embora o conjunto original de métricas CK tenham seis métricas, foram utilizadas apenas três para formar o subconjunto. São elas: WMC, CBO, e DIT. A falta de aplicabilidade das outras métricas, para o modelo, e a potencial dificuldade

no cálculo dessas medidas, levou [Subramanyam & Krishnan \(2003\)](#) a excluí-las. Para comprovar os benefícios do subconjunto proposto, foram coletados dados de uma empresa de desenvolvimento de software que utiliza duas linguagens populares de programação (Java e C++) Orientado a Objeto. Após a aplicação do conjunto de métricas em um ambiente real, [Subramanyam & Krishnan \(2003\)](#) evidenciam algumas contribuições do seu trabalho. A primeira contribuição do trabalho é o efeito de métricas CK sobre defeitos, após o controle do tamanho do software. Outra contribuição é a validação da associação entre um subconjunto de métricas CK e defeitos em dois ambientes de linguagem atual, ou seja, Java e C++. Por fim, os recém-citados autores se valem da regressão linear ponderada para evidenciar o efeito da interação de algumas dessas medidas em defeitos de software.

[Subramanyam & Krishnan \(2003\)](#) concluíram que após o controle do tamanho, algumas das medidas do pacote de CK, como por exemplo, métricas de complexidade de projeto, explicariam significativamente a variância em defeitos. Além disso, descobriram que os efeitos dessas métricas sobre defeitos variam entre as amostras de duas linguagens de programação Java e C++. [Subramanyam & Krishnan \(2003\)](#) sustentam que esses resultados têm implicações importantes para a concepção de produtos de alta qualidade de software utilizando a abordagem OO.

Neste contexto, existem diferenças do trabalho do [Subramanyam & Krishnan \(2003\)](#) para o trabalho proposto. Primeiro, [Subramanyam & Krishnan \(2003\)](#) propõem um subconjunto de métricas para defeitos, enquanto esta dissertação tem métricas de defeito como parte de alguns indicadores, conforme observado em 4.1 , 4.2, 4.6. Além disso é específico para projeto orientado a objeto. Entretanto, nossos indicadores que tem como foco defeito, podem servir como informações adicionais para o trabalho de [Subramanyam & Krishnan \(2003\)](#). Além dos trabalhos relacionados com o objetivo de identificar métricas, foram estudados trabalhos com o objetivo de criar indicadores a partir das métricas.

## 6.2 Trabalhos objetivando Indicadores

Os objetivos desses trabalhos são de criar e/ou identificar indicadores e comprovar os benefícios de sua utilização no processo de desenvolvimento de software. Como exemplo, [Monteiro & de Oliveira \(2011\)](#) apresentam uma lista de indicadores para apoiar a análise de desempenho do processo. Tais indicadores têm como objetivo apoiar os gestores a realizar uma análise de desempenho de processo do Capability Maturity Model Integration for Development (CMMI-DEV). [Monteiro & de Oliveira \(2011\)](#) estabelecem a definição e o uso do catálogo de indicadores. Entretanto, especifica completamente apenas dois indicadores: Densidade de Defeitos e Valor Agregado.

O objetivo do trabalho de [Monteiro & de Oliveira \(2011\)](#) é ajudar as organizações a definir as medidas que possam ser efetivamente utilizadas no desempenho do processo organiza-

cional (OPP). Para tanto, foi realizada uma pesquisa na literatura visando definir um catálogo de medidas organizadas por indicadores. O catálogo proposto suporta a análise de OPP para CMMI-DEV, processos de engenharia, contemplando as principais atividades técnicas de projetos de desenvolvimento de software.

Para definição do catálogo, [Monteiro & de Oliveira \(2011\)](#) decidiram usar métricas que foram definidas e implementadas em organizações propostas por diferentes autores. Para isso, foi realizada uma ampla revisão bibliográfica objetivando identificar as métricas existentes e quais são utilizadas na análise de desempenho do processo. Em seguida, foram analisadas e selecionadas as métricas mais relevantes para compor os indicadores. Em suma, o trabalho de [Monteiro & de Oliveira \(2011\)](#) objetiva fornecer um catálogo como forma de apoio adequado durante a realização do processo desempenho organizacional (OPP) como pratica definida no CMMI-DEV. Por fim, [Monteiro & de Oliveira \(2011\)](#) estabelecem a construção de um catálogo de indicadores para apoiar as organizações na seleção, especificação e utilização de métricas durante a aplicação do CMMI-DEV.

Neste contexto, existem diferenças do trabalho do [Monteiro & de Oliveira \(2011\)](#) para o trabalho proposto. Primeiro, a abordagem utilizada é voltada para o nível 3 do CMMI-DEV. Já esta dissertação tem a abordagem das boas praticas do PMBOK (2004). Além disso, esta dissertação não necessita de um nível de maturidade muito elevado conforme proposto por [Monteiro & de Oliveira \(2011\)](#). Entretanto, assim como o trabalho de [Monteiro & de Oliveira \(2011\)](#) tem o objetivo de alcançar o nível 4 de maturidade, esta dissertação tem o objetivo de auxiliar o gerente de projeto. Em comum, os dois trabalhos objetivam apoiar as organizações na seleção, especificação e utilização de indicadores para aprimorar o processo de desenvolvimento de software.

[McQuighan & Hammell \(2011\)](#) propôs a utilização de inteligência computacional para facilitar o acompanhamento do escopo, com o objetivo de fornecer informações úteis para o gerente de projeto. Para [McQuighan & Hammell \(2011\)](#) a utilização de indicadores adicionais, além de custo e tempo, podem identificar problemas de projeto. Portanto, propõe adicionar escopo como forma de verificar o status do projeto. Para isso, faz uso de ferramenta da inteligência computacional. Como projeto de software tem propriedade de imprecisão e indefinição, [McQuighan & Hammell \(2011\)](#) utiliza a lógica fuzzy para auxiliar o monitoramento de escopo. Assim, o objetivo da ferramenta de inteligência computacional é identificar em tempo hábil o ponto ou os pontos onde o escopo não conseguiu ser compreendido pelos participantes do projeto e, portanto, tem que ser refinado. Com a identificação precisa de um problema, os gerentes de projeto poderão implementar conceitos que suportem as correções, assim evitando ou diminuindo problemas futuros. Assim, uma sensibilização para os problemas de escopo deve contribuir para oportunas ações corretivas, aumentando desta forma o sucesso do projeto.

Neste contexto, existem diferenças do trabalho do [McQuighan & Hammell \(2011\)](#) para o trabalho proposto. O autor mencionado, utiliza a ferramenta de inteligência computacional

para acompanhar escopo enquanto este trabalho não faz uso de nenhuma ferramenta de inteligência computacional. Foram utilizados os dados do processo, agrupando os mesmos, de forma a gerar informações necessárias para tomada de decisão do gerente de projeto. Além disso, [McQuighan & Hammell \(2011\)](#) utiliza apenas escopo como área de concentração, ao passo que esta dissertação utiliza tempo e custo. Entretanto, este trabalho tem como objetivo em comum identificar pontos falhos, em tempo hábil, para que o gerente de projeto consiga implementar conceitos que suporte correção.

No trabalho de [Chandrasekaran & Kumar \(2012\)](#) propõe-se a utilização da ferramenta de Análise de Valor Agregado (EVA) com o objetivo de propor, ao gerente de projeto, conhecer o status e prever desempenho futuro do projeto. Com isso, o gerente de projeto pode prever o alvo a ser atingindo ao longo da linha de base (OTB) e também a estimativa no término (EAC) dos projetos em andamento, com base na análise de valor agregado (EVA), utilizando técnicas de regressão logística. Para este fim, foi feito uso de métricas de estimativas progressivas de projetos, tais como, o custo de linha de base, valor planejado, valor agregado e Custo Real.

Como resultado, o trabalho de [Chandrasekaran & Kumar \(2012\)](#) pretende obter uma melhor previsão do custo do projeto e diminuição do risco de custo do projeto superfaturado e, portanto, podendo ser benéfico para o planejamento de ações preventivas. [Chandrasekaran & Kumar \(2012\)](#) fazem uso de técnica de regressão logística para prever o tempo estimado de conclusão (EAC) de um projeto de desenvolvimento. Com isso, o objetivo principal é identificar qualquer desvio, o mais cedo possível, de tal forma que o gerente de projeto possa ter tempo suficiente para avaliar o desvio do projeto e tomar ações corretivas, se necessário. Para isso, [Chandrasekaran & Kumar \(2012\)](#) valem-se da Análise de Valor Agregado (EVA) em toda fase de um projeto, desde a iniciação até o encerramento, para monitorar, medir, elaborar relatórios e prever o desempenho do projeto de software. Assim, a Análise de Valor Agregado requer três métricas abaixo expostas:

- Valor Planejado (PV) é o custo orçado para o Trabalho Planejado (BCWS)
- Valor Agregado (EV) é o custo orçado do Trabalho Realizado (COTR)
- Custo real (AC) é o total incorrido na realização o Trabalho Planejado (CRTR).

Este trabalho pretende desenvolver um modelo preventivo em relação ao Alvo Inicial e Estimativa na conclusão de projetos de software.

Neste contexto, existem diferenças do trabalho do [Chandrasekaran & Kumar \(2012\)](#) para o trabalho proposto. Primeiro, utiliza ferramenta de Análise de Valor Agregado para conhecer o status e prever o desempenho futuro do projeto. Nosso trabalho não faz uso de Valor Agregado. Podemos analisar em trabalhos futuros a incorporação dos indicadores proposto por [Chandrasekaran & Kumar \(2012\)](#) em nosso trabalho. Entretanto, temos em comum o valor planejado e custo real em uns de nossos indicadores.

# 7

## Conclusão

Alguns fatores são determinantes para o sucesso ou fracasso de um projeto. Um desses fatores são as decisões tomadas pelo gerente de projetos para atingir as metas. Para auxiliar nessas decisões e nas tarefas do gerente de projeto, um acompanhamento adequado com um conjunto de indicadores, pode contribuir para alcançar essas metas. Um conjunto de indicadores contém informações importantes sobre o andamento do projeto. Para isso, sua criação deve seguir um método estruturado e definido. Neste contexto, foram utilizadas as dimensões de tempo e custo. No Capítulo 3 são citados alguns estudos que indicam o quanto essas dimensões contribuem para o fracasso do projeto.

Para criar o conjunto de indicadores foi utilizado o método GQ(IM) proposto por [Park et al. \(1996\)](#), com algumas adaptações. Esse método adaptado descrito no Capítulo 3 pode ser utilizado para criação de novos indicadores, sendo uma das contribuições desse trabalho. Para isso, foram documentados todos os passos da criação dos indicadores. Com isso, formando uma estrutura que pode ser seguida para criação de novos indicadores.

O conjunto de indicadores foi aplicado em um estudo de caso que teve uma abordagem qualitativa. Com indícios positivos que o conjunto de indicadores pode auxiliar o gerente de projeto em suas atividades. Os indicadores propostos satisfazem importantes requisitos para as dimensões de custo e tempo, e os risco inerentes a ambas. As entrevistas realizadas no estudo de caso indicam positivamente que o conjunto de indicadores contém informações relevantes para auxiliar o gerente de projeto a tomar decisões. Os gerentes de projetos relataram que os indicadores auxiliam na tomada de decisão. Porém, alguns pontos foram levantados sobre algumas informações dos indicadores. Como por exemplo, identificação do processo de classificação dos defeitos. Nosso trabalho não propõe definição do processo e nem como classificar os defeitos. Entretanto, os indicadores só podem ser utilizados se os dados coletados no processo de desenvolvimento foram armazenados de forma estruturada, preferencialmente em uma base de dados única.

A realização do estudo de caso serviu como uma primeira avaliação da utilidade e usabilidade do conjunto de indicadores. Pode-se concluir, então, que o estudo de caso serviu como

indicação positiva de que o conjunto de indicadores por ser estudado mais profundamente em outros contextos de forma a poder ser refinado e amadurecido.

## 7.1 Limitações

Existem algumas limitações para este trabalho. Para ser aplicado, se faz necessário que o processo de desenvolvimento de software possua sua estrutura já definida para custo e tempo e que se utilizem ferramentas que auxiliem no processo. Essas ferramentas irão oferecer dados para aplicação dos indicadores. Os dados coletados devem ser armazenados de forma estruturada, preferencialmente em uma base de dados única e que especifique com precisão as entidades envolvidas, seus atributos e relacionamentos. Essa base servirá para a aplicação dos indicadores e permitirá o acompanhamento e controle dos dados do projeto. Para isso, precisa utilizar uma ferramenta que ofereça suporte aos processos de extração de bases distintas, tratamento de dados e carga para o modelo único.

## 7.2 Trabalhos Futuros

O trabalho apresentado nesta dissertação pode ser continuado com a realização das seguintes atividades:

- Realização de outros estudos de caso. Atualmente, o conjunto de indicadores já está sendo usado em outro estudo como as mesmas características.
- Definição dos conjuntos de soluções que acompanham os indicadores para cumprimento das metas.
- O conjunto de indicadores como base de conhecimento onde sistemas multi-agentes usando normas possam monitorar e equilibrar o ambiente para o cumprimento das metas, por exemplo, atividade que devem ter maior prioridade pode usar o indicador de estado de atividade por pessoa como base de conhecimento para o agente normativo. Segue um exemplo da norma:

(Norma 1): Se o numero de atividade em aberto for maior X% do total de atividades, o Desenvolvedor X é obrigado a reduzir para X%

- (Recompensas): Se o desenvolvedor reduzir para a margem aceitável
  - \* (Recompensa 1.1): Ele é premiado com .....
- (Punições): Se o desenvolvedor não reduzir
  - \* (Punição 1.1): Sua reputação é diminuída dado que ele não está contribuindo;

Com isso pode ser criada uma pontuação para os desenvolvedores com finalidade de atingir as metas estabelecidas. Assim os agentes teriam um papel de recompensar ou penalizar os desenvolvedores. Sendo este mais um auxílio para o gerente de projeto. De fato, este trabalho já está em andamento porém sem nenhum estudo de caso.

## REFERÊNCIAS

- Basili, V. (1994), 'Goal question metric paradigm', *Encyclopedia of Software Engineering*. URL <http://www.cs.umd.edu/users/basili/publications/technical/T87.pdf>.
- Baumert, J. H. & McWhinney, M. S. (1992), Software measures and the capability maturity model, Technical report. URL <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA257238&Location=U2&doc=GetTRDoc.pdf>.
- Bhatti, S. N. (2005), 'Why quality? iso 9126 software quality metrics (functionality) support by uml suite', *ACM SIGSOFT Software Engineering Notes* **30**(2), 1–5. URL <http://portal.acm.org/citation.cfm?doid=1050849.1050860>.
- Budzier, A. (2012), 'Why your it project may be riskier than you think'. URL [http://www.researchgate.net/publication/225070625\\_Why\\_your\\_IT\\_project\\_might\\_be\\_riskier\\_than\\_you\\_think/file/d912f4fc5f0ec7434f.pdf](http://www.researchgate.net/publication/225070625_Why_your_IT_project_might_be_riskier_than_you_think/file/d912f4fc5f0ec7434f.pdf).
- Chandrasekaran, R. & Kumar, R. V. (2012), 'Article: Application of logistic regression to predict over target baseline of software projects', *International Journal of Computer Applications* **44**(10), 1–6. Published by Foundation of Computer Science, New York, USA.
- Cusumano, M. A. (1989), 'The software factory: A historical interpretation', *IEEE Softw.* **6**(2), 23–30. URL <http://dx.doi.org/10.1109/MS.1989.1430446>.
- F, B. V. A. S. P. A. C. F. J. S. L. S. (2008), *Fundamentos de gerenciamento de projeto*, FGV.
- Fenton, N. E. (1996), *Software Metrics: A Rigorous and Practical Approach*, 2nd ed., International Thomson Computer Press, Boston, MA, USA.
- Fernandes, A. A. & Teixeira, D. S. (2004), *Fábrica de Software*, Atlas.
- Grady, R. B. (1992), *Practical software metrics for project management and process improvement*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Islam, A. K. M. M., Gorschek, T., Unterkalmsteiner, M., Feldt, R., Permadi, R. B. & Cheng, C. K. (2012), 'Evaluation and measurement of software process improvement - a systematic literature review', *IEEE Transactions on Software Engineering* **38**(2), 398–424.
- Jones, C. (2006), 'Social and technical reasons for software project failures', *CrossTalk* pp. 4–9. URL [http://nas.uhcl.edu/Boetticher/SWEN5430/WhySoftwareProjectsFail\\_CapersJones.pdf](http://nas.uhcl.edu/Boetticher/SWEN5430/WhySoftwareProjectsFail_CapersJones.pdf).



- McQuighan, J. M. & Hammell, R. J. (2011), Scope as a leading indicator for managing software development, *in* 'Proceedings of the 2011 Ninth International Conference on Software Engineering Research, Management and Applications', SERA '11, IEEE Computer Society, Washington, DC, USA, pp. 235–241. URL <http://dx.doi.org/10.1109/SERA.2011.25>.
- Monteiro, L. F. S. & de Oliveira, K. M. (2011), 'Defining a catalog of indicators to support process performance analysis', *J. Softw. Maint. Evol.* **23**(6), 395–422. URL <http://dx.doi.org/10.1002/smr.482>.
- Paes, R. (2005), 'Agentes e métricas no processo de desenvolvimento de software em equipes distribuídas', *Journal of Computer Science*. URL <http://www.dcc.ufla.br/infocomp/artigos/v4.2/art07.pdf>.
- Park, R. E., Goethert, W. B. & Florac, W. A. (1996), Goal-driven software measurement. a guidebook., Technical report. URL <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA313946&Location=U2&doc=GetTRDoc.pdf>.
- PMBOK (2004), *A Guide To The Project Management Body Of Knowledge - PMBOK Guides*, Project Management Institute.
- PRESSMAN, R. (2006), *ENGENHARIA DE SOFTWARE*, MCGRAW HILL.
- Rubinstein, D. (2007), 'Standish group report: There's less development chaos today', *Software Development Times*. URL [http://pdd.citsolutions.edu.au/Clients/DOGPM/documentation/Standish\\_Group\\_Chaos\\_Article\\_2006.pdf](http://pdd.citsolutions.edu.au/Clients/DOGPM/documentation/Standish_Group_Chaos_Article_2006.pdf).
- Runeson, P. & Host, M. (2008), 'Guidelines for conducting and reporting case study research in software engineering', *Empirical Software Engineering* **14**(2), 131–164. URL <http://www.springerlink.com/index/10.1007/s10664-008-9102-8>.
- Sant'Anna, C. N. (2004), *Manutenibilidade e reusabilidade de software orientado a aspectos: Um framework de avaliação*, Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro.
- Sommerville, I. (2011), *Engenharia de Software*, Pearson Prentice Hall.
- Subramanyam, R. & Krishnan, M. S. (2003), 'Empirical analysis of ck metrics for object-oriented design complexity: Implications for software defects', *IEEE Trans. Softw. Eng.* **29**(4), 297–310. URL <http://dx.doi.org/10.1109/TSE.2003.1191795>.
- van Solingen, R. & Berghout, E. (1999), 'The goal/question/metric method'. URL [http://scholar.google.com/scholar?q=related:1B6lKAp47MoJ:scholar.google.com/&hl=en&num=30&as\\_sdt=0,5](http://scholar.google.com/scholar?q=related:1B6lKAp47MoJ:scholar.google.com/&hl=en&num=30&as_sdt=0,5).

- Wang, Y. & Bryant, A. (2002), 'Process-based software engineering: Building the infrastructures - editors' introduction', *Annals of Software Engineering* **14**, 9–37. URL <http://links.isiglobalnet2.com/gateway/Gateway.cgi?GWVersion=2&SrcAuth=mekentosj&SrcApp=Papers&DestLinkType=FullRecord&DestApp=WOS&KeyUT=000178531400001>.
- Wongthongtham, P., Chang, E., Dillon, T. & Sommerville, I. (2009), 'Development of a software engineering ontology for multisite software development', *IEEE Transactions on Knowledge and Data Engineering* **21**(8), 1205–1217. URL <http://links.isiglobalnet2.com/gateway/Gateway.cgi?GWVersion=2&SrcAuth=mekentosj&SrcApp=Papers&DestLinkType=FullRecord&DestApp=WOS&KeyUT=000267122600009>.
- Xie, T., Thummalapenta, S., Lo, D. & Liu, C. (2009), 'Data mining for software engineering', *IEEE Computer*. URL [http://dbgroup.cs.tsinghua.edu.cn/wangyi/refs/kdd06/docs/t\\_3.pdf](http://dbgroup.cs.tsinghua.edu.cn/wangyi/refs/kdd06/docs/t_3.pdf).

# **Appendix A**

## **APÊNDICE - ENTREVISTAS**

Seguem as entrevista realizadas com os Gerentes de Projetos interno e externo, Líder de Equipe e Desenvolvedor. O objetivo da entrevista, era saber em pose do indicadores, os entrevistados, conseguia responder as perguntas. As respostas possuíam duas opções, sim ou não. Quando o entrevistado respondia que com os indicadores NÃO tinha a resposta da pergunta era questionado o por que. As tabelas a seguir lista as respostas de cada uma entrevista.

Table A.1: Entrevista - Gerente A

Meta	Pergunta
MT-01	• Qual é a quantidade de defeitos aberto no projeto?
Resposta	Sim (X), Não ( ) e Por quê? Com o indicador temos um visão de como esta os defeitos.
MT-01	• Qual tem sido a evolução dos tipos de defeitos ao longo dos projetos de desenvolvimento?
Resposta	Sim (X), Não ( ) e Por quê? Pois temos o números dos defeitos por tipos.
MT-01	• A quantidade de defeitos está sob controle, ou representa algum risco para a conclusão tempestiva do projeto?
Resposta	Sim (X), Não ( ) e Por quê? Com os indicadores 1 e 2 tempos um visão do defeitos que estão bloqueando alguns tarefas.
MT-01	• Qual o esforço gasto em cada atividade?
Resposta	Sim (X), Não ( ) e Por quê? Temos por atividade e por projeto.
MT-01	• Qual a distribuição de produção por pessoa?
Resposta	Sim ( ), Não (X) e Por quê? Temos a produção por linha de código, mas não sei o quanto de fato essa linha de código agrega ao projeto.
MT-01	• Qual e a relação de atividade por pessoa e estado da atividade?
Resposta	Sim (X), Não ( ) e Por quê? Com isso verificamos se existe um numero de atividades não desejada em um estado por pessoa.
MT-02	• Os esforços foram subestimados?
Resposta	Sim ( ), Não (X) e Por quê? Precisamos avaliar se atividades foram atribuídas as pessoas corretas.
MT-02	• Qual é a tendência do item de escopo?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Os custos foram subestimados?
Resposta	Sim (X), Não ( ) e Por quê? Com os indicadores temos os custos planejados e realizados.
MT-02	• Qual a fração já concluída do projeto?
Resposta	Sim (X), Não ( ) e Por quê? Temos informações da conclusão das atividades.
MT-02	• Qual o esforço já gasto até então?
Resposta	Sim (X), Não ( ) e Por quê? Temos a informação do custo.
MT-02	• Como está a relação estimado x realizado?
Resposta	Sim (X), Não ( ) e Por quê? Se temos o planejado e o realizado então sabemos a relação.
MT-02	• Como esta a taxa de chagada e encerramento do escopo?
Resposta	Sim (X), Não ( ) e Por quê? Temos o numero de criação e encerramento.

Table A.2: Entrevista - Gerente B

Meta	Pergunta
MT-01	• Qual é a quantidade de defeitos aberto no projeto?
Resposta	Sim (X), Não ( ) e Por quê? Temos no I-2.
MT-01	• Qual tem sido a evolução dos tipos de defeitos ao longo dos projetos de desenvolvimento?
Resposta	Sim ( ), Não (X) e Por quê? Pois não temos por datas.
MT-01	• A quantidade de defeitos está sob controle, ou representa algum risco para a conclusão tempestiva do projeto?
Resposta	Sim (X), Não ( ) e Por quê? Indicadores I-1 e I-2.
MT-01	• Qual o esforço gasto em cada atividade?
Resposta	Sim (X), Não ( ) e Por quê? Indicadores I-5
MT-01	• Qual a distribuição de produção por pessoa?
Resposta	Sim (X), Não ( ) e Por quê? Indicador I-7.
MT-01	• Qual e a relação de atividade por pessoa e estado da atividade?
Resposta	Sim (X), Não ( ) e Por quê? Indicador I-3.
MT-02	• Os esforços foram subestimados?
Resposta	Sim ( ), Não (X) e Por quê? Teria que analisar o final de cada atividade para saber se o planejado foi o realizado.
MT-02	• Qual é a tendência do item de escopo?
Resposta	Sim (X), Não ( ) e Por quê? Indicador I-6.
MT-02	• Os custos foram subestimados?
Resposta	Sim ( ), Não (X) e Por quê? Teria que analisar o final de cada atividade para saber se o planejado foi o realizado.
MT-02	• Qual a fração já concluída do projeto?
Resposta	Sim (X), Não ( ) e Por quê? Indicador I-4, porém tem que fazer o calculo.
MT-02	• Qual o esforço já gasto até então?
Resposta	Sim (X), Não ( ) e Por quê? Indicador I-4.
MT-02	• Como está a relação estimado x realizado?
Resposta	Sim (X), Não ( ) e Por quê? Indicador I-4 e I-8.
MT-02	• Como esta a taxa de chagada e encerramento do escopo?
Resposta	Sim (X), Não ( ) e Por quê? Indicador I-9.

Table A.3: Entrevista - Gerente E

Meta	Pergunta
MT-01	• Qual é a quantidade de defeitos aberto no projeto?
Resposta	Sim (X), Não ( ) e Por quê?
MT-01	• Qual tem sido a evolução dos tipos de defeitos ao longo dos projetos de desenvolvimento?
Resposta	Sim ( ), Não (X) e Por quê? Não tem uma dimensão tempo.
MT-01	• A quantidade de defeitos está sob controle, ou representa algum risco para a conclusão tempestiva do projeto?
Resposta	Sim (X), Não ( ) e Por quê? Temos um conjunto de indicadores bons sobre defeito. Porém precisamos saber como estão classificados.
MT-01	• Qual o esforço gasto em cada atividade?
Resposta	Sim (X), Não ( ) e Por quê?
MT-01	• Qual a distribuição de produção por pessoa?
Resposta	Sim (X), Não ( ) e Por quê? Temos o indicador 4.7. Porém se mudar a estratégia para PF?
MT-01	• Qual e a relação de atividade por pessoa e estado da atividade?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Os esforços foram subestimados?
Resposta	Sim ( ), Não (X) e Por quê? Só saberemos na entrega do projeto. Pois podemos cumprir os esforço planejado e não entregar o produto esperado pelo cliente.
MT-02	• Qual é a tendência do item de escopo?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Os custos foram subestimados?
Resposta	Sim ( ), Não (X) e Por quê? Só saberemos na entrega do projeto. Pois podemos cumprir os custos planejado e não entregar o produto esperado pelo cliente.
MT-02	• Qual a fração já concluída do projeto?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Qual o esforço já gasto até então?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Como está a relação estimado x realizado?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Como esta a taxa de chegada e encerramento do escopo?
Resposta	Sim (X), Não ( ) e Por quê?

Table A.4: Entrevista - Líder de Equipe A

Meta	Pergunta
MT-01	• Qual é a quantidade de defeitos aberto no projeto?
Resposta	Sim (X), Não ( ) e Por quê?
MT-01	• Qual tem sido a evolução dos tipos de defeitos ao longo dos projetos de desenvolvimento?
Resposta	Sim ( ), Não (X) e Por quê? Evolução só pode ser vista com uma dimensão tempo.
MT-01	• A quantidade de defeitos está sob controle, ou representa algum risco para a conclusão tempestiva do projeto?
Resposta	Sim (X), Não ( ) e Por quê?
MT-01	• Qual o esforço gasto em cada atividade?
Resposta	Sim (X), Não ( ) e Por quê?
MT-01	• Qual a distribuição de produção por pessoa?
Resposta	Sim (X), Não ( ) e Por quê?
MT-01	• Qual e a relação de atividade por pessoa e estado da atividade?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Os esforços foram subestimados?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Qual é a tendência do item de escopo?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Os custos foram subestimados?
Resposta	Sim ( ), Não (X) e Por quê? Depende do projeto. Existe projeto que o Cliente pede alteração no meio do projeto.
MT-02	• Qual a fração já concluída do projeto?
Resposta	Sim ( ), Não (X) e Por quê? Falta o calculo.
MT-02	• Qual o esforço já gasto até então?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Como está a relação estimado x realizado?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Como esta a taxa de chegada e encerramento do escopo?
Resposta	Sim (X), Não ( ) e Por quê?

Table A.5: Entrevista - Desenvolvedor A

Meta	Pergunta
MT-01	• Qual é a quantidade de defeitos aberto no projeto?
Resposta	Sim (X), Não ( ) e Por quê?
MT-01	• Qual tem sido a evolução dos tipos de defeitos ao longo dos projetos de desenvolvimento?
Resposta	Sim (X), Não ( ) e Por quê?
MT-01	• A quantidade de defeitos está sob controle, ou representa algum risco para a conclusão tempestiva do projeto?
Resposta	Sim ( ), Não (X) e Por quê? Não sabe identificar riscos.
MT-01	• Qual o esforço gasto em cada atividade?
Resposta	Sim (X), Não ( ) e Por quê?
MT-01	• Qual a distribuição de produção por pessoa?
Resposta	Sim (X), Não ( ) e Por quê?
MT-01	• Qual e a relação de atividade por pessoa e estado da atividade?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Os esforços foram subestimados?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Qual é a tendência do item de escopo?
Resposta	Sim ( ), Não (X) e Por quê? Pode ser do momento.
MT-02	• Os custos foram subestimados?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Qual a fração já concluída do projeto?
Resposta	Sim ( ), Não (X) e Por quê? Não tem o calculo.
MT-02	• Qual o esforço já gasto até então?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Como está a relação estimado x realizado?
Resposta	Sim (X), Não ( ) e Por quê?
MT-02	• Como esta a taxa de chagada e encerramento do escopo?
Resposta	Sim ( ), Não (X) e Por quê? Não falta o calculo.



Este trabalho foi redigido em  $\text{\LaTeX}$  utilizando uma modificação do estilo IC-UFAL. As referências bibliográficas foram preparadas no JabRef e administradas pelo  $\text{\BIBTeX}$  com o estilo LaCCAN. O texto utiliza fonte Fourier-GUTenberg e os elementos matemáticos a família tipográfica Euler Virtual Math, ambas em corpo de 12 pontos.