

Definição de Indicadores para a Gerência de Requisitos

Claudia Hazan^{1 2}
claudinhah@yahoo.com

Julio Cesar Sampaio do Prado Leite²
www.inf.puc-rio.br/~julio

¹ SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS (SERPRO)
SGAN Quadra 601 - Módulo V - SUPCT
CEP: 70836-900 -Brasília – DF
Brasil

² DEPARTAMENTO DE INFORMÁTICA PUC-Rio
Rua Marquês de São Vicente 225, Gávea
CEP: 22453-900 – Rio de Janeiro - RJ
Brasil

Resumo

A implementação de boas práticas de gerência de requisitos de software é uma prioridade na implantação de melhoria do processo de software. Os indicadores de desempenho são formas de representação quantificáveis de características de produtos e processos, sendo utilizados para acompanhar e melhorar os resultados ao longo do tempo. Este trabalho tem como propósito definir indicadores para apoiar um gerenciamento efetivo de requisitos funcionais, baseado em dados. O objetivo é promover um acompanhamento da rastreabilidade e estabilidade dos requisitos, fornecendo subsídios para a realização de ações corretivas.

Palavras-chave: Gerência de Software, Gerência de Requisitos, Indicadores, Rastreabilidade, Estabilidade, Qualidade de *Software*

1. Introdução

A medição de software tem evoluído dentro da disciplina Engenharia de Software. No passado, muitas organizações de software tratavam as medições como um trabalho adicional. No entanto, em organizações maduras a medição é implementada como uma disciplina pró-ativa, e os indicadores constituem uma ferramenta estratégica [15]. A implantação de um processo de medição pode ser vista como o primeiro passo na implantação de um processo de melhoria de software, pois as medições e as análises constituem a base da Gerência pela Qualidade Total (*TQM -Total Quality Management*) [25].

Os requisitos de software evoluem devido às mudanças no ambiente do sistema e conforme os clientes desenvolvem um melhor entendimento de suas necessidades reais. Assim, torna-se fundamental gerenciar efetivamente as mudanças de requisitos funcionais, os quais incluem: novos requisitos, requisitos alterados e requisitos excluídos. Os indicadores constituem uma fundação para decisões gerenciais racionais [13]. Segundo De Marco [4], o gerenciamento eficaz dos parâmetros quantitativos de projeto resulta em planejamento e controle mais eficazes. Leite em [28] ressalta que as métricas são essenciais para uma comunicação objetiva e precisa. Portanto, é fundamental para garantir qualidade, que os processos sejam baseados em dados quantitativos. Este trabalho propõe indicadores de desempenho para apoiar o processo de gerência de requisitos, baseando-se em informações quantitativas.

Os indicadores de requisitos definidos são classificados em dois grupos: **Indicadores de Estabilidade** e **Indicadores de Rastreabilidade**. O Indicador de Estabilidade analisa o grau de mudanças para a *baseline* dos requisitos do *software* e o impacto destas mudanças no esforço de desenvolvimento. A Rastreabilidade é necessária para controlar: as mudanças nos requisitos, o processo de desenvolvimento e os riscos. O Indicador de Rastreabilidade mede: a aderência dos produtos de *software* (especificações, desenho e código) com os requisitos deles nos vários níveis de detalhamento do produto; o percentual de requisitos que podem ser rastreados das especificações alocadas para o próximo nível mais baixo de especificação, descendo até o nível de listagens de código.

2. Gerência de Requisitos

Leite em [28] ressalta que o processo de construção de software será cada vez mais baseado no conceito de evolução [29]. Entende-se que o próprio processo de definição de requisitos gera um *feedback* que acaba modificando os próprios requisitos. Portanto é falsa a idéia de congelar requisitos. Concordamos com Lehman [33] que caracteriza o processo de *software* como: “*a complex, multi loop multilevel feedback system*”.

Os requisitos devem estar associados a idéia de *baseline*. Como os requisitos não podem ser congelados, a *baseline* também evolui. A definição da *baseline* de requisitos [31] [30] explica que essa evolução se dá em dois eixos, um no que diz respeito a pontos de referência do modelo de processo de *software* e outro eixo no que diz respeito a progressão do processo de software no que se refere a mudança de nível de abstração.

A Gerência de Requisitos tem como objetivo principal controlar a evolução dos requisitos, seja por constatação de novas necessidades, seja por constatação de deficiências nos requisitos registrados até o momento [23].

A **tabela I** apresenta o conjunto de atividades de um processo de gerência de requisitos. Estas atividades visam apoiar a identificação, controle e rastreamento dos requisitos, bem como o tratamento das mudanças nos requisitos. As principais preocupações da gerência de requisitos são: Gerenciar mudanças nos requisitos acordados; Gerenciar os relacionamentos entre os requisitos; Gerenciar as dependências entre o documento de requisitos e outros documentos produzidos ao longo do processo [11].

Segundo El Emam [5], os principais problemas relacionados à gerência de requisitos são os seguintes:

- Dificuldades de elicitar claramente as mudanças nos requisitos;
- Falta de habilidade para chegar a um consenso sobre as mudanças chave para os *stakeholders*;
- Falta de habilidade para manter o documento de requisitos consistente;
- Falta de habilidade para estimar adequadamente os recursos necessários para implementar as mudanças nos requisitos.

Tabela I: Atividades de um Processo de Gerência de Requisitos [21]

ATIVIDADES	DESCRIÇÃO
<i>Receber as solicitações de alteração de requisitos</i>	O grupo de engenharia de requisitos recebe as solicitações de alteração de requisitos, ou por formulário padronizado, ou por meio de um sistema de solicitação de demandas.
<i>Registrar novos requisitos</i>	Novos requisitos também devem ser recebidos formalmente, seja por formulário padronizado, ou por meio de controle sistemático.
<i>Analisar impacto da mudança de requisitos</i>	Uma análise criteriosa deve ser conduzida para avaliar o impacto do requisito a ser incluído, alterado ou excluído sobre cada um dos seus requisitos relacionados, os quais podem ser identificados por meio das matrizes de rastreabilidade. Caso o impacto seja significativo, os requisitos (analisado e relacionado) devem ser revistos.
<i>Elaborar relatório de impacto</i>	Deve ser mantido um histórico de alterações para cada requisito, permitindo uma visão cronológica das principais mudanças nos requisitos.
<i>Notificar os envolvidos</i>	Os envolvidos são um conjunto de pessoas para as quais pode haver um impacto devido à alteração de requisitos (alteração, inclusão ou exclusão de requisitos) e devem ser notificados.
<i>Coletar métricas</i>	As métricas devem ser utilizadas e coletadas periodicamente para o acompanhamento das atividades de Gerência de Requisitos.

Além disso, a Gerência de Requisitos é fundamental para o tratamento de riscos dos projetos de software. Segundo Jones [10], o principal risco que atinge 80 % dos projetos de *software* é o da **Evolução de Requisitos**. Este risco é definido como:

- Novos requisitos ou modificações significativas nos requisitos existentes que são feitas após o conjunto básico de requisitos acordado pelos clientes e desenvolvedores;

- Falhas para antecipar mudanças de requisitos (previsão de aumento ou mudança do escopo do projeto) e fazer planos para lidar com estes.

Portanto, como os requisitos são voláteis, a própria natureza do processo leva ao que Jones classifica de risco. Segundo Jones [10], os principais problemas associados a este risco são os seguintes: atritos entre a equipe de desenvolvimento gerentes e usuários, não atendimento ao prazo acordado, software de baixa qualidade e altos custos.

Uma das estratégias para se diminuir o impacto da evolução é a implantação de um processo de medições, baseado em métricas funcionais, como Pontos de Função (PF) [8], visando dimensionar o tamanho dos novos requisitos e estimar os impactos no esforço, prazo e custo de desenvolvimento. Estas estimativas apóiam as negociações com os clientes e as mudanças no plano do projeto. Vale notar que essa estratégia procura remediar o problema, mas não implementar um processo de construção centrado em evolução como proposto em [30].

As duas seções seguintes têm como propósito apresentar uma análise feita pela primeira autora das abordagens dos modelos: *Project Management Body Of Knowledge* (PMBOK) [17] e *Capability Maturity Model Integration* (CMMI) [20] para a gerência de requisitos.

2.1. PMBOK

O PMBOK é um guia que representa o conhecimento da profissão de gerência de projetos, que é definida como a aplicação de conhecimentos, habilidades, e técnicas para projetar atividades, visando atender as expectativas dos *stakeholders*, com relação ao projeto [17].

O processo de **Controle de Mudança de Escopo**, relacionado à gerência de requisitos, contido na área de conhecimento **Gerência do escopo do Projeto** do PMBOK, consiste em: tratar os fatores que criam mudanças no escopo para garantir que estas sejam benéficas; determinar que uma mudança ocorreu, e gerenciar efetivamente as mudanças [17].

O Controle de Mudança de Escopo deve ser completamente integrado com os outros processos de controle (controle de prazo, controle de custo, controle de qualidade, e outros). As mudanças de requisitos influenciam diretamente a mudança de escopo. Os principais insumos para o processo de Controle de Mudança de Escopo são os seguintes:

- **Plano de gerência do escopo:** descreve como o escopo do projeto será gerenciado e como as mudanças no escopo serão integradas ao projeto. O plano deve conter: uma avaliação da estabilidade esperada do escopo do projeto (quanto este muda, com que frequência e custo); e uma descrição sobre como as mudanças no escopo serão identificadas e classificadas.

- **Requisições de mudança:** As mudanças podem expandir ou reduzir o escopo do projeto. A maioria das demandas de mudança são resultados de: Um evento externo (por exemplo, mudanças em uma regulamentação governamental); Um erro ou omissão no detalhamento do escopo do projeto ou do produto; Uma mudança no valor agregado.

Uma ferramenta importante a ser utilizada é o **Sistema de Controle de Mudanças do Escopo**. Este sistema define os procedimentos pelos quais o escopo do projeto pode ser modificado. Inclui manuais, sistemas de monitoração e níveis de aprovação necessários para autorização das mudanças. A análise de indicadores de desempenho ajuda ao gerente a determinar a magnitude e as causas das mudanças, apoiando as decisões.

2.2. O Modelo CMMI

O modelo *Capability Maturity Model Integration* (CMMI) tem como objetivo fornecer um guia para melhorar os processos e a habilidade de gerenciar o desenvolvimento, aquisição e manutenção de produtos e serviços. O modelo coloca as melhores práticas dentro de uma estrutura que ajuda na avaliação da maturidade organizacional ou da capacidade da área do processo, estabelecendo prioridades para a melhoria, e implementando estas melhorias [20].

O modelo CMMI identifica cinco níveis de maturidade através dos quais uma organização deve evoluir para estabelecer uma cultura de excelência. Cada nível de maturidade contém **Áreas de Processo**, as quais são definidas como um conjunto de práticas que, quando executadas coletivamente, satisfazem um conjunto de objetivos para realizar uma melhoria.

A **Gerência de Requisitos de Software** é uma área de processo do nível 2 – Gerenciado do CMMI, tendo como propósito gerenciar os requisitos dos produtos, do projeto e dos componentes do produto e identificar as inconsistências entre os requisitos e os planos do projeto e produtos de trabalho.

Segundo o modelo, as principais atividades da gerência de requisitos são documentar as mudanças de requisitos e manter a rastreabilidade bidirecional entre requisitos fonte e todos os requisitos do produto e dos componentes do produto. [20]

É importante destacar que a área de processo **Medição e Análise** do nível 2 do CMMI, a qual tem como propósito desenvolver e sustentar uma capacidade de medição usada para suportar gerencialmente as necessidades de informação, relaciona-se diretamente com a área de processo Gerência de Requisitos, visando atender as necessidades de informação relativas à rastreabilidade e estabilidade dos requisitos.

Vale ressaltar o CMMI [20] faz uma distinção não presente no modelo SW-CMM [32]. O CMMI separa a Gerência de Requisitos (*Requirements Management*) da Elaboração de Requisitos (*Requirements Development*). Essa divisão pode ser considerada artificial, considerando a visão da constante evolução dos requisitos, mas ajuda a separar preocupações distintas, isto é a preocupação da gerência com a do entendimento e definição dos requisitos.

3. Indicadores

A área de processo Medição e Análise do CMMI envolve o seguinte [20]:

- Especificação dos objetivos de medição e análise de forma que estes sejam alinhados com as necessidades de informação identificadas e objetivos;
- Especificação das medidas, mecanismos de coleta de dados e de armazenamento, técnicas de análise, e mecanismos de comunicação e de *feedback*;
- Implementação da coleta, armazenamento, análise, e comunicação dos dados;
- Fornecimento de resultados objetivos que podem ser usados na tomada de decisão e implementação de ações corretivas apropriadas.

Assim, visando implementar a área de processo Medição e Análise, sugere-se o estabelecimento de **Indicadores**, os quais são definidos como dados ou representações numéricas de características de produtos e processos utilizados para acompanhar e melhorar os resultados ao longo do tempo.

Um indicador é um dado numérico, expresso em uma unidade de medida, ao qual se atribui uma meta e que é trazido periodicamente à atenção dos gestores dos processos, com a finalidade de apoiá-los na avaliação do desempenho [22] [27].

As decisões devem ser baseadas no resultado dos indicadores, levando em consideração as **tendências** e os **referenciais de comparação**. Uma análise de tendência leva em consideração o comportamento de um conjunto de resultados de um Indicador específico ao longo do tempo. Os Critérios de Excelência [26] consideram uma tendência favorável a variação consecutiva (com melhoria de resultados) de forma sustentada de no mínimo três períodos de tempo. Os resultados dos indicadores de referenciais comparativos podem ser internos ou externos à organização.

Os indicadores gerenciais definidos neste trabalho consideram apenas os **Requisitos Funcionais**, os quais referem-se aos aspectos da funcionalidade do *software* sendo desenvolvido, ou seja são as funções que deverão ser incorporadas pelo *software*, quando inserido em seu contexto de funcionamento. Os requisitos funcionais podem ser representados em forma de **sentenças**. Leite [14] apresenta a seguinte estrutura para a definição das sentenças de requisitos:

O sistema deve + [verbo + objeto| frase verbal] + [complemento de agente| null] + [condição| null]

Onde:

- **Verbo** é um verbo que expressa a funcionalidade do requisito.
- **Objeto** pode ser direto ou indireto, dependendo do tipo do verbo.
- **Frase verbal** é uma frase que expressa a funcionalidade do requisito.
- **Complemento de agente** é a identificação de um agente relacionado com o requisito. Um agente pode ser uma pessoa, uma instituição, um grupo ou um dispositivo físico externo ao *software*.
- Uma **condição** é uma sub-sentença que reflete uma situação específica.

Para este trabalho, as mudanças de requisitos estão associadas apenas às alterações nas funcionalidades da aplicação, ou seja inclusão, alteração ou exclusão de sentenças de requisitos. Neste contexto, a alteração de uma sentença é considerada uma mudança de requisito, apenas se houver mudança na funcionalidade do requisito. Suponha o seguinte requisito de uma aplicação hipotética “O sistema deve gerar um gráfico de barras, totalizando os projetos concluídos **por mês**”. Um exemplo de requisito alterado pode ser “O sistema deve gerar um gráfico de barras, totalizando projetos concluídos **por trimestre**”.

Deve-se selecionar um conjunto de métricas pequeno e equilibrado, que irá ajudar a organização a acompanhar o progresso na direção de seus objetivos [7]. O método **GQM (Goal Question Metric)** [1] [2] é utilizado para definir as métricas apropriadas aos seus objetivos. A aplicação do método GQM, inicia-se selecionando alguns objetivos de medição do processo ou da organização. As fontes para os objetivos podem ser necessidades gerenciais, técnicas, de projeto, de produto, ou de implementação do processo. Declaram-se os objetivos de modo que sejam quantificáveis e mensuráveis. Posteriormente, para cada objetivo, identificam-se as perguntas que precisam ser respondidas para determinar se o objetivo está sendo alcançado. Finalmente, identificam-se métricas que ajudam a responder cada pergunta.

Os indicadores apresentados nas seções seguintes foram definidos aplicando o método **GQM** (*Goal Question Metric*) [1] [2] em projetos do **Serviço Federal de Processamento de Dados (SERPRO)** e classificados em: Indicadores de Estabilidade e Indicadores de Rastreabilidade.

3.1. Indicadores de Estabilidade

A indústria tem mostrado que a instabilidade dos requisitos contribui fortemente para os riscos de pressão excessiva do cronograma e de não aceitação do produto final definidos em [10]. Mudanças em requisitos ocorrem enquanto os requisitos estão sendo elicitados, analisados e após o sistema ter entrado em produção, sendo resultantes da combinação de fatores descritos na **tabela II**.

Uma boa prática é a antecipação das mudanças de requisitos, que envolve classificar os requisitos para identificar os mais voláteis e prever possíveis mudanças. Estas informações são úteis para projetar o sistema de forma que os requisitos sejam implementados com independência de componentes, para minimizar a influência das mudanças no sistema.

A métrica Pontos de Função (PF) [8], a qual tem como objetivo medir a funcionalidade requisitada e recebida pelo usuário independentemente da tecnologia utilizada, ajuda a determinar o tamanho das mudanças de requisitos e a evolução do tamanho do sistema. Além do tamanho deve-se considerar a criticidade dos requisitos modificados.

TABELA II: Fatores de Mudanças de Requisitos [11]

Fator de mudança	Descrição
Erros em requisitos, conflitos e inconsistências	Conforme os requisitos são analisados e implementados, erros e inconsistências surgem e devem ser corrigidas.
Evolução do conhecimento do cliente	Conforme os requisitos são desenvolvidos, clientes e usuários finais desenvolvem uma melhor compreensão do que desejam.
Problemas técnicos, de custo ou cronograma	Problemas podem ser encontrados na implementação dos requisitos. Pode ser muito custoso implementar certos requisitos.
Mudanças nas prioridades do cliente	As prioridades do cliente podem mudar durante o desenvolvimento do sistema como resultado de mudanças no ambiente de negócios.
Mudanças de ambiente	O ambiente no qual o sistema será instalado pode mudar, assim os requisitos devem ser modificados para manter compatibilidade.
Mudanças organizacionais	A Organização que pretende usar o sistema pode mudar sua estrutura e processos, resultando em novos requisitos de sistema.

Os principais objetivos, perguntas e métricas (*Goal/Question/Metric*) relativos à estabilidade de requisitos funcionais incluem o seguinte:

GOAL: Controlar as Mudanças nos Requisitos (<i>Baseline dos Requisitos Alocados</i>).	
Questions	Metrics
Qual o percentual de novos requisitos no período?	<ul style="list-style-type: none"> <i>nº de requisitos novos/nº requisitos alocados</i> <i>requisitos novos (PFs)/ requisitos alocados (PFs)</i>
Qual o percentual de requisitos alterados no período?	<ul style="list-style-type: none"> <i>nº de requisitos alterados/nº requisitos alocados</i> <i>requisitos alterados (PFs)/ requisitos alocados(PFs)</i>
Qual o percentual de requisitos excluídos no período?	<ul style="list-style-type: none"> <i>nº de requisitos excluídos/nº requisitos alocados</i> <i>requisitos excluídos (PFs)/ requisitos alocados(PFs)</i>

O próximo passo é o refinamento das métricas criadas por meio do método GQM em um indicador. Os indicadores devem ser quantitativamente confiáveis e interpretáveis [15]. A seguir, utiliza-se o formulário de especificação de indicadores definido em [7] para apresentar o **Indicador de Estabilidade de Requisitos**.

ESPECIFICAÇÃO DO INDICADOR DE ESTABILIDADE DE REQUISITOS

Título	Sigla	Revisão
INDICADOR DE MUDANÇAS DE REQUISITOS	IMR	08/07/2002

Tipo –chave	Abrangência	Unidade de Medida	Periodicidade
Processos Relativos ao Produto	Unidades de Desenvolvimento	Percentual (%)	Revisões de Acompanhamento dos Projetos

1 – DEFINIÇÃO	O indicador de estabilidade de requisitos afere o grau de mudanças para a <i>Baseline</i> dos requisitos de software.
2 – OBJETIVO	O indicador fornece o impacto da mudança no tamanho da <i>Baseline</i> , sob o ponto de vista funcional.
3 – FÓRMULA DE CÁLCULO	$IMR = (RI + RA + RE) / RB$
4 – DEFINIÇÃO DOS PARÂMETROS	RI = Tamanho dos requisitos incluídos (novos) em PFs RA = Tamanho dos requisitos alterados em PFs RE = Tamanho dos requisitos excluídos em PFs RB = Tamanho dos requisitos alocados na <i>baseline</i> de requisitos do projeto em PFs
5 – SUGESTÃO DE FONTE DE DADOS	Documento de Requisitos do Projeto de Software
6 – MÉTODO DE MEDIÇÃO	Dimensionar o tamanho dos requisitos funcionais novos, alterados e excluídos, utilizando a técnica Análise de Pontos por Função. Caso os requisitos alterados e excluídos ainda não tenham sido implantados, considerar o percentual de PFs relativos às macro atividades concluídas.
7 – EXEMPLO	Um projeto de software implantado possui 300 PFs. Para facilitar os cálculos, supor que o fator de ajuste seja 1. Em um período determinado, 5 gráficos simples foram alterados (tamanho = 20PFs) e foram requisitados 2 relatórios estatísticos de complexidade média (tamanho = 10 PFs). Assim, RI =10 PFs, RA = 20 PFs e RE =0 PFs. Portanto, $IMR = (10 + 20 + 0) / 300 = 10 \%$
8 – MÉTODO DE ANÁLISE	Os resultados do indicador devem ser analisados nos marcos definidos plano do projeto. Caso o percentual esteja acima dos limites estabelecidos, deve-se analisar a possibilidade de refazer o produto.
9 – MÉTODO DE MELHORIA OU USO	O indicador deve ser analisado semestralmente ou anualmente, visando a verificação de sua aderência com os objetivos relativos à estabilidade de requisitos.
10 – REFERENCIAIS DE COMPARAÇÃO	Unidades de Desenvolvimento da organização
11 – OBSERVAÇÕES	A coleta de dados para o indicador IMR inicia-se no começo do projeto e continua em marcos pré definidos. A partir do tamanho (PF) pode-se estimar esforço e custo. E assim, obter-se o impacto mudança no esforço, custo e prazo do projeto, antes que esta seja implementada.
12 – RESPONSÁVEL PELA MEDIÇÃO E ANÁLISE	Gerente de Requisitos e Grupo de Garantia da Qualidade de Software
13 – RESPONSÁVEL PELA MELHORIA DO USO	Grupo de Garantia da Qualidade de Software
14- ARQUIVO	Intranet – seção de Indicadores da Gerência de Requisitos

De posse da especificação, inicia-se a coleta dos dados que irão alimentar o indicador. Explicitar os métodos de coleta ajuda a garantir que os dados corretos são coletados adequadamente. Mecanismos para coleta de dados devem ser bem integrados com os outros processos de trabalho, podendo incluir formas manuais ou automáticas e *templates*. Guias claros e concisos em procedimentos devem estar disponíveis pelos responsáveis pela realização do trabalho. Os responsáveis pela coleta de dados para o Indicador de Estabilidade especificado devem estar capacitados em Gerência de Requisitos e na Técnica Análise de Pontos de Função para garantir a consistência dos dados coletados.

Outro ponto a ser considerado é o gerenciamento e armazenamento dos dados de medições, especificações de medições e análise de resultados. Armazenar as informações relativas ao indicador habilita o uso futuro dos dados históricos e dos resultados. As informações a serem armazenadas incluem o seguinte: planos de medição, especificações de medidas, conjuntos de dados que tenham sido coletados e relatórios e gráficos de apresentações da análise.

Outra atividade fundamental do processo de medição é a análise dos resultados do indicador. Durante a análise devem ser preparados relatórios e gráficos para consolidação dos resultados da análise. Estes relatórios e gráficos devem ser previamente especificados e disseminados para as pessoas envolvidas. A apresentação dos resultados deve ser claramente entendida pelos envolvidos, que receberão os resultados. Além disso, é fundamental especificar os procedimentos administrativos para comunicação dos resultados da análise. As questões a serem consideradas incluem o seguinte: Identificar as pessoas e grupos responsáveis pela análise dos dados e apresentação dos resultados; Determinar os marcos para analisar os dados e apresentar os resultados; Determinar as formas para comunicar os resultados (exemplo: relatórios escritos, transmissão por correio eletrônico, ou reuniões com as equipes).

Sugere-se a utilização de gráficos para análise das medidas de mudanças nos requisitos. Os seguintes tipos de gráficos podem ser utilizados: Quantitativo de mudanças aprovadas, estratificado por tipo de mudança e pelo responsável pela solicitação da mudança (**figura 1**); Número de mudanças aprovadas e acumuladas por mês (**figura 2**). As mudanças abertas relacionam-se as mudanças que estão sendo desenvolvidas e as mudanças fechadas são aquelas que foram concluídas e entregues ao usuário no mês em questão. A identificação e a solução das mudanças cedo minimiza o impacto no cronograma do projeto.

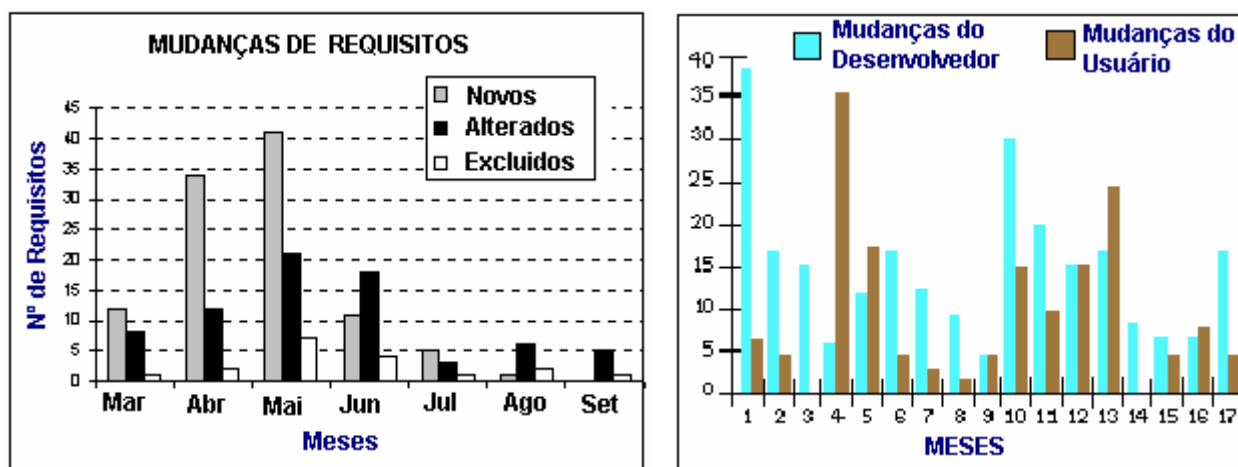


Figura 1: Gráfico Quantitativo de mudanças aprovadas estratificado por tipo de mudança e por responsável pela solicitação da mudança [19]

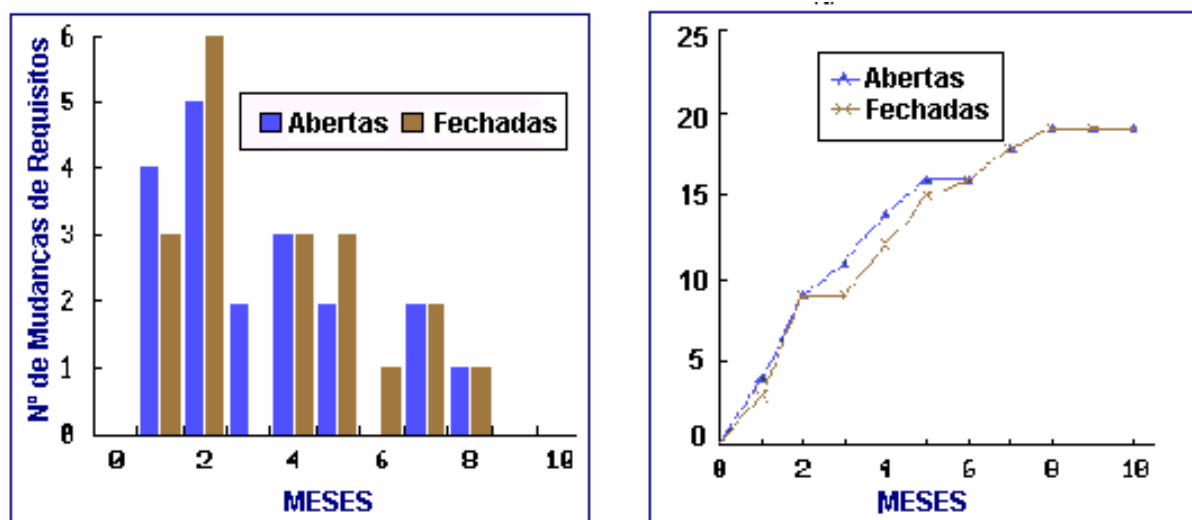


Figura 2: Número de mudanças aprovadas e acumuladas por mês [19]

Em um Sistema de Indicadores é essencial que os indicadores sejam correlacionados para o fornecimento de subsídios para a tomada de decisões baseadas em dados integrados. O Indicador de Estabilidade deve ser correlacionado com os indicadores de tamanho (Pontos de Função - PF), de esforço (PF/Pessoa_mês), de custo (R\$/PF) e de cronograma (Nº de dias úteis). Estes indicadores apóiam o Planejamento e o Acompanhamento do Projeto, visto que a implementação dos requisitos modificados influencia o tamanho do projeto, demanda esforço, e conseqüentemente possui tempo de cronograma e custo associados (**figura 3**).

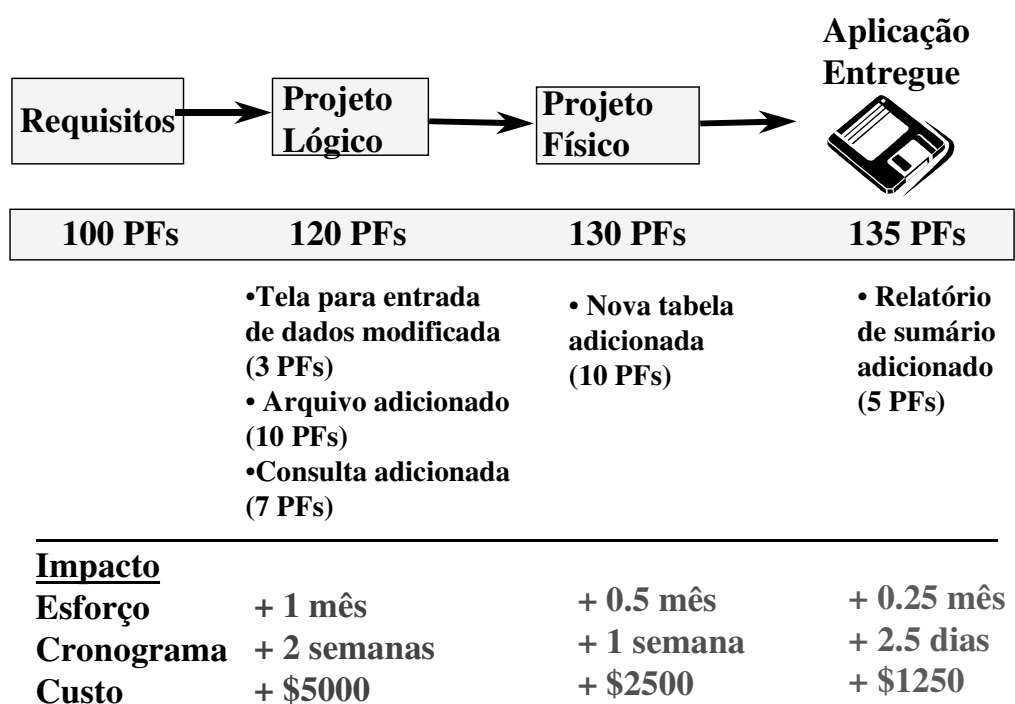


Figura 3: Medições para mudança de escopo gerencial

Também é importante destacar que as medidas de estabilidade de requisitos relacionam-se com a rastreabilidade, uma mudança de requisitos pode impactar nos rastros estabelecidos. E ainda, os rastros devem ser considerados em uma análise de impacto das mudanças.

3.2. Indicadores de Rastreabilidade

A Rastreabilidade fornece uma assistência fundamental ao entendimento dos relacionamentos que existem entre requisitos e outros artefatos do processo de software, sendo uma forma de garantir como e porque os artefatos satisfazem os requisitos dos clientes internos e externos, especialmente em sistemas complexos. Ou seja, a rastreabilidade auxilia na verificação e validação dos requisitos dos clientes [3] [16].

O processo de desenvolvimento deve produzir requisitos rastreáveis, isto é capazes de serem rastreados para a sua origem. Assim, a rastreabilidade de requisitos pode ser vista como a habilidade de acompanhar e descrever a vida de um requisito, em ambas as direções; pré-rastreabilidade documenta a movimentação e o contexto a partir do qual emergem os requisitos (origem dos requisitos); pós-rastreabilidade está relacionada ao refinamento, desdobramento e uso do requisito, vinculando os requisitos ao desenho do sistema e a sua implementação [3] [19]. A existência da pré-rastreabilidade é essencial para gerenciar a evolução do *software*, tornando possível a identificação de impacto de mudanças quando essas são solicitadas por clientes e ainda não sistematizadas como requisitos. A **figura 4** fornece uma idéia de onde se situa a pré (mundo dos clientes e usuários) e a pos rastreabilidade (mundo dos desenvolvedores).

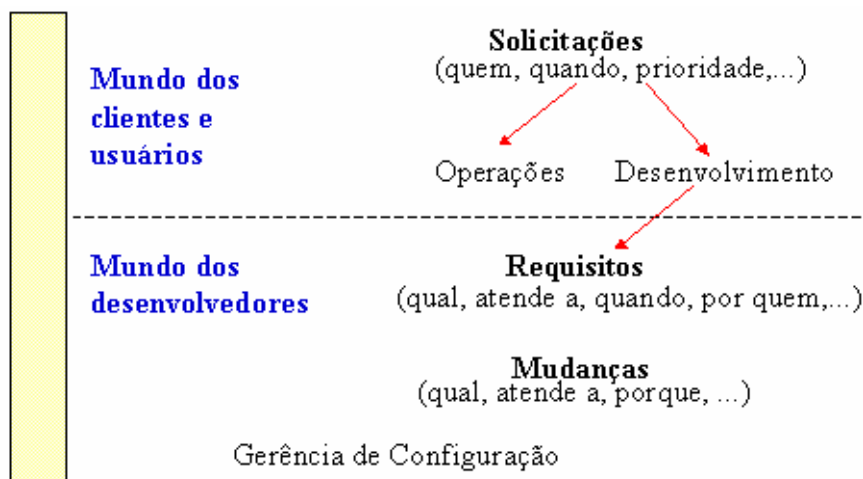


Figura 4: Pré e Pos Rastreabilidade¹

Os rastros dos requisitos devem continuar sendo atualizados após a implantação do *software*. A rastreabilidade é especialmente crítica para a fase de operação e manutenção para garantir que o sistema entregue continue satisfazendo as necessidades dos envolvidos (clientes, usuários, desenvolvedores). Quando mudanças nos requisitos emergirem, uma análise de impactos das mudanças deve ser executada, visando verificar a viabilidade de implementação imediata ou em uma próxima versão do produto, bem como o esforço, custo e cronograma associados a tal implementação.

Na evolução do sistema, a rastreabilidade apoia a referência cruzada entre as especificações de requisitos e as de projeto. Os rastros também ajudam na identificação do tamanho (PF) da mudança solicitada. Além disso, os procedimentos de teste são facilmente modificados quando um erro é encontrado, caso exista rastreabilidade aos requisitos e ao desenho [18].

¹ Esta figura é adaptada de uma figura usada no projeto PER (www.er.les.inf.puc-rio.br/projPER.htm) e originalmente proposta por Francisco Pinheiro.

Os principais objetivos, perguntas e métricas (*Goal/Question/Metric*) relativos à rastreabilidade de requisitos funcionais incluem o seguinte:

GOAL: Controlar a aderência dos produtos de software (especificações, desenho e código) com os requisitos deles nos vários níveis de especificação do produto	
Questions	Metrics
Qual o percentual de requisitos rastreáveis até sua origem? (pré-rastreabilidade)	<ul style="list-style-type: none"> • <i>nº de requisitos rastreáveis para a origem</i> • <i>nº requisitos rastreáveis para a sua origem/ nº total de requisitos alocados</i>
Qual o percentual de requisitos rastreáveis para o próximo nível mais baixo? (pós-rastreabilidade)	<ul style="list-style-type: none"> • <i>nº de requisitos rastreáveis para a próxima atividade</i> • <i>nº requisitos rastreáveis a próxima atividade / nº total de requisitos alocados</i>
Qual o impacto operacional dos requisitos modificados? (efeitos sobre componentes do software)	<ul style="list-style-type: none"> • <i>nº de requisitos impactados/ nº requisitos alocados</i> • <i>requisitos impactados (PFs)/ requisitos alocados(PFs)</i>

A seguir, considera-se as métricas estabelecidas por meio do método GQM na especificação do **Indicador de Requisitos Rastreáveis**, utilizando o formulário de especificação de Indicadores definido em [7].

ESPECIFICAÇÃO DO INDICADOR DE RASTREABILIDADE DE REQUISITOS

Título	Sigla	Revisão
INDICADOR DE REQUISITOS RASTREÁVEIS	IRR	08/07/2002

Tipo –chave	Abrangência	Unidade de Medida	Periodicidade
Processos Relativos ao Produto	Unidades de Desenvolvimento	Percentual (%)	Revisões de Acompanhamento dos Projetos

1 – DEFINIÇÃO	O indicador fornece um indicativo dos requisitos rastreáveis contidos na <i>baseline</i> de requisitos do software.
2 – OBJETIVO	O indicador de rastreabilidade de requisitos mede o percentual de requisitos que podem ser rastreados entre dois níveis adjacentes de especificação.
3 – FÓRMULA DE CÁLCULO	$IRR = RR/RA$
4 – DEFINIÇÃO DOS PARÂMETROS	RR = Requisitos rastreáveis RA = Requisitos alocados ao software na macro atividade (desenho, teste, etc.) considerada para cálculo do indicador
5 – SUGESTÃO DE FONTE DE DADOS	Matriz de Rastreabilidade
6 - MÉTODO DE MEDIÇÃO	O indicador é calculado como uma série de porcentagens de requisitos, os quais podem ser rastreados entre dois níveis adjacentes de especificações. Os dados para o indicador são coletados em dois passos: Primeiro são identificados os dados que constituem as especificações de requisitos. Os dados são usados para popular a matriz de rastreabilidade. O segundo conjunto de dados é o percentual de requisitos que podem ser rastreados para a origem.
7 – EXEMPLO	Um projeto de software possui 10 requisitos de negócio. Observando-se a matriz de rastreabilidade, nota-se que apenas 8 estão relacionados aos requisitos do software. Assim, o indicador $IRR = 8/10$ ou 80%.
8 - MÉTODO DE ANÁLISE	O indicador deve ser usado para determinar se o <i>software</i> está pronto para proceder para a próxima macro atividade. Por exemplo, o sistema SRH deve ter 90 % de rastreabilidade dos requisitos para implementação antes de iniciar-se os testes.

9 - MÉTODO DE MELHORIA OU USO	O indicador deve ser analisado periodicamente, visando a verificação de sua aderência com os objetivos relativos à rastreabilidade de requisitos.
10 – REFERENCIAIS DE COMPARAÇÃO	Unidades de Desenvolvimento da empresa
11 – OBSERVAÇÕES	A coleta de dados para o indicador inicia-se no começo do projeto e continua em marcos pré definidos.
12 – RESPONSÁVEL PELA MEDIÇÃO E ANÁLISE	Gerente de Requisitos do Projeto e Grupo de Garantia da Qualidade de Software
13 – RESPONSÁVEL PELA MELHORIA DO USO	Grupo de Garantia da Qualidade de Software
14- ARQUIVO	Intranet – seção de Indicadores da Gerência de Requisitos

A Matriz de Rastreabilidade (**figura 5**) é uma ferramenta que proporciona visibilidade ao rastreamento e ao relacionamento dentro e entre os requisitos, desenho, código e casos de teste. Note que os pontos de interrogação (?) indicam a rastreabilidade que ainda não ocorreu. A rastreabilidade facilita a comunicação entre as pessoas envolvidas no projeto, reduzindo os problemas decorrentes de falhas na comunicação. Os módulos que aparecem com mais frequência na matriz de rastreabilidade podem ser alvos para serem desenvolvidos inicialmente e realizar-se um exame cuidadoso nos testes.

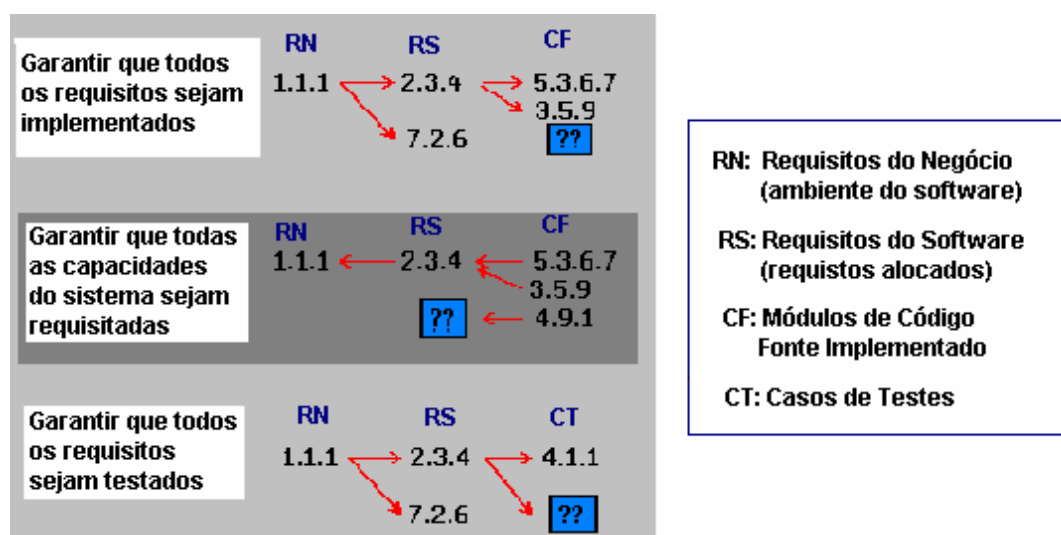


Figura 5: Matriz Rastreabilidade de Requisitos

Pode-se observar na **figura 6** que o **Indicador de Requisitos Rastreáveis** representa uma outra visão do processo de desenvolvimento de *software*. Esta nova visão apresenta o progresso do desenvolvimento de *software* sob o ponto de vista dos rastros identificados na matriz de rastreabilidade e consolidados no indicador.

O Indicador de Rastreabilidade de Requisitos exerce um impacto direto sob outros indicadores gerenciais, como, por exemplo, o indicador de progresso - **Percentual de Abrangência de Requisitos**, ilustrado na **figura 7**, que tem como propósito apresentar o percentual de requisitos estabelecidos para o projeto e implementados em cada iteração, levando em consideração as principais categorias de produtos de trabalho (ou artefatos) do processo desenvolvimento de *software*. Note que é importante que o percentual de requisitos rastreáveis seja alto para assegurar a correteza da interpretação dos resultados do indicador de Abrangência de Requisitos.

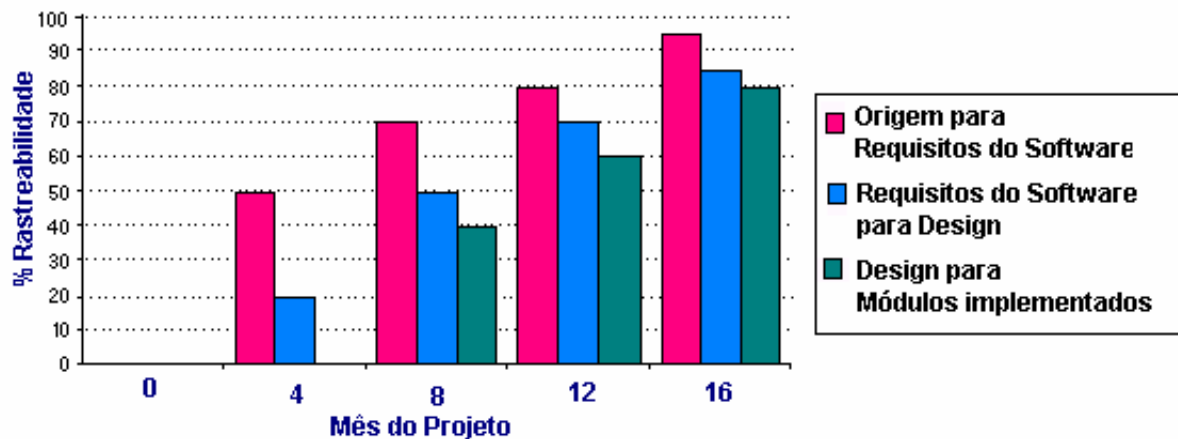


Figura 6: Análise do Indicador de Rastreabilidade de Requisitos

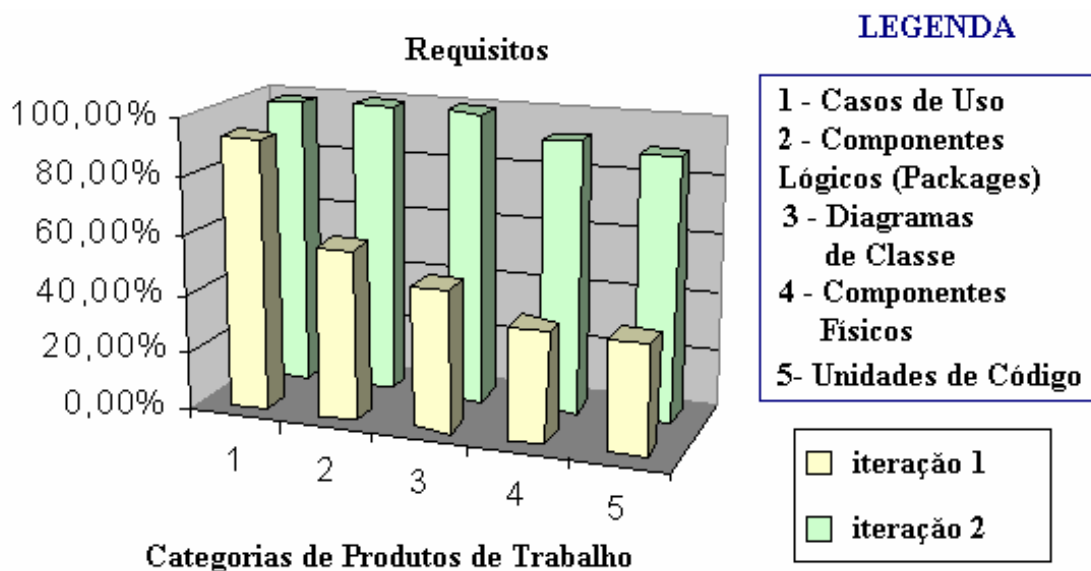


Figura 7: Percentual de Abrangência de Requisitos

4. Conclusão

Uma situação freqüente na indústria de software é que os projetos geralmente atrasam e na maioria das vezes isso é devido a um processo de requisitos inadequado. Vários autores apontam para esse fato, como por exemplo Kotonya [11].

A rastreabilidade fornece a habilidade de tratar a história de toda característica do sistema, apoiando na identificação dos impactos de mudanças nos requisitos [6]. Um requisito é rastreável se for possível identificar quem o solicitou, porque ele existe, quais os requisitos relacionados e como o requisito se relaciona a outras informações provenientes do *desenho*, implementações e documentos do usuário. Os artefatos rastreáveis são essenciais para lidar com mudanças, levam a menos erros durante o desenvolvimento, desempenham um papel importante em situações contratuais e melhoram a aceitação do *software* [3] [9].

Além disso, deve-se controlar a estabilidade de requisitos, inclusive após a implantação do produto de *software*. Embora a evolução esteja sempre presente, é comum que alguns requisitos sejam mais estáveis que outros. É fundamental que requisitos voláteis sejam acompanhados cuidadosamente [12]. Portanto, os indicadores de rastreabilidade e estabilidade definidos neste trabalho são peças fundamentais em uma gerência de requisitos efetiva.

Novos indicadores gerenciais para requisitos podem ser estabelecidos para atender as necessidades de informações relativas à estabilidade e à rastreabilidade de requisitos, utilizando os atributos descritos no formulário [7], apresentado neste trabalho. No entanto, é importante destacar que “O indicador é a expressão didática da realidade” [24]. Ou seja, o indicador deve ser referenciado à realidade, pois tem que exprimi-la. Nossos resultados são iniciais, mas baseados em sólida experiência com medição, em especial em Pontos de Função. Esperamos no futuro que essa proposta possa ser utilizada para que ocorra uma evolução natural de sua definição e adequabilidade. Como expressão da realidade, os indicadores devem evoluir e para isso é fundamental o seu uso. Um programa de acompanhamento do uso desses indicadores certamente será fundamental para que possamos aperfeiçoá-los.

Referências

- [1] BASILI. Software Modelling and Measurement: The Goal/Question/Metric Paradigm. Technical Report CS-TR-2956, University of Maryland, September 1992.
- [2] BASILI, CALDIERA, ROMBACH. Goal Question Metric Paradigm. Encyclopedia of Software Engineering, volume 1, John Wiley & Sons, 1994.
- [3] DAVIS. *Software Requirements Engineering*. Second Edition, IEEE Computer Society Press, 1997.
- [4] DEMARCO. Controle de Projetos de Software. Editora Campus, 1991.
- [5] EL EMAM. Causal Analyses of the Requirements Change Process for a Large System. IESE-Report No. 054.97/E, 1997.
- [6] HAMILTON & BEEBY. Issues of Traceability in Integrating Tools. IEEE Colloquium Tools and Techniques for Maintaining Traceability during Design, dezembro 1991.
- [7] HAZAN. Implantação de um Processo de Medições de Software. CITS:QS, Junho 2001.
- [8] IFPUG. Counting Practices Manual. Versão 4.1.1, Abril, 2000.
- [9] JARKE et all. Nature Requirements Engineering. Sahaker Verlag, 1999.
- [10] JONES. Assessment and Control of Software Risks. Prentice Hall, 1994.
- [11] KOTONYA & SOMMERVILLE. Requirements Engineering: Processes and Techniques. John Willey & Sons Ltd, 1998.
- [12] LAWRENCE. The Top Risks os Requirements Engineering. IEEE Software. November/December 2001.
- [13] LAM. Managing Requirements Change Using Metrics and Action Planning. Third European Conference on Software Maintenance and Reengineering, 1999.
- [14] LEITE. Engenharia de Requisitos: Notas de Aula – Parte IV, PUC-Rio, 1994.

- [15] MCGARRY Et All. Practical Software Measurement: Objective Information for Decision Makers. Addison Wesley, 2001
- [16] PALMER. Traceability. Software Requirements Engineering, R.H. Thayer and M. Dorfman, eds., pp. 364-374, 1997.
- [17] PMI - Project Management Institute. A guide to the project management body of knowledge. Syba: PMI Publishing Division, www.pmi.org, 2000.
- [18] RAMESH & JARKE. Towards reference Models for Requirements Traceability. IEEE Transactions Software Engineering, vol. 27, no. 1, pp. 58-93, Janeiro de 2001.
- [19] ROSENBERG. Testing Metrics for Requirement Quality. GSFC NASA, 2002.
- [20] SEI. CMMI: Capability Maturity Model Integration (CMMISM). Version 1.1 - CMU/SEI-2002-TR-012; Março 2002. www.sei.cmu.edu
- [21] SERPRO. Processo SERPRO de Desenvolvimento de Soluções (PSDS). Janeiro 2002.
- [22] TAKASHINA & FLORES. Indicadores da qualidade e do desempenho – como estabelecer metas e medir resultados, Qualitymark Editora, 1996.
- [23] THAYER & DORFMAN. Software Requirements Engineering. Second Edition, IEEE Computer Society, 1997.
- [24] KARDEC Et All. Gestão Estratégica e Avaliação do Desempenho. Qualitymark, 2002.
- [25] HAZAN. Introdução da Gerência pela Qualidade Total em Organizações de Desenvolvimento de Software. WQS, 1999.
- [26] FPNQ. Critérios de Excelência: O estado da arte da gestão para excelência do desempenho. Fundação para o Prêmio Nacional da Qualidade, www.fpnq.org.br, 2003.
- [27] FPNQ. Critérios de Excelência: Planejamento do Sistema de Medição do Desempenho Global. Fundação para o Prêmio Nacional da Qualidade, 2001.
- [28] LEITE, J.C.S.P Qualidade de Software: Teoria e Prática, Rocha, A. R., Maldonado, J.C. e Weber, K. C., Prentice Hall, 2001, pp. 238-246, 2001.
- [29] LEITE, J.C.S.P. Software Evolution, The Requirements Engineering View, keynote address 26 Jaio, Proceedings SoST'97, JAIQ, SADIO, Buenos Aires, 1997, pp. 21-23.
- [30] LEITE, J.C.S.P.; Rossi, G.; Maiorana, V.; Balaguer, F.; Kaplan, G.; Hadad, G.; Oliveros, A. Enhancing a Requirements *Baseline* with Scenarios. Requirements Engineering Journal, Vol. 2, N. 4, 1997, pp.184 -- 198.
- [31] LEITE, J.C.S.P., Oliveira, A.P.A. A Client Oriented Requirements *Baseline*, in Proceedings of the Second IEEE International Symposium on Requirements Engineering, York, UK, IEEE Computer Society Press, 1995 pp. 108-115.
- [32] PAULK, M. C.; et al. The Capability Maturity Model For *Software*. Version 1.1 (CMU/SEI-93-TR-24), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (USA), 1993.
- [33] LEHMAN, M.M. Evolution in the Context of Software Technology, Encyclopedia of Software Engineering, Wiley and Co., 1994, vol. 2, pp. 1202 - 1208