

Darcy Borges dos Reis Nogueira

**Proposta de indicadores para a avaliação do progresso
de desenvolvimento e qualidade de software**

Dissertação apresentada ao Instituto de Pesquisas Tecnológicas
do Estado de São Paulo - IPT, para obtenção do título de
Mestre em Engenharia de Computação,
área de concentração: Engenharia de Software.

Orientador: Prof. Dr. Ivanir Costa

São Paulo

Agosto de 2007

DEDICATÓRIA

*Ao meu pai, Raymundo, que me inculcou o hábito da leitura;
à minha mãe, Lucília, que me ensinou a calma para cultivá-lo.*

AGRADECIMENTOS

Ao Professor Doutor Ivanir Costa, meu orientador, que, desde o começo, ajudou a transformar um conjunto de boas idéias num trabalho coeso, no aspecto metodológico, e comigo discutiu o conteúdo de incontáveis versões;

Aos membros adicionais da Banca, Professor Doutor Mauro Mesquita Spinola e Professor Doutor Marcelo Novaes de Rezende, que com suas pormenorizadas observações e críticas, ajudaram a firmar o rumo da dissertação;

À minha esposa, Maria Amélia, pela paciência com que encarou os usuais períodos de ausência demandados por um mestrado, e pelo esforço de desbravar a prosa difícil de uma dissertação à cata de vírgulas mal colocadas e outras ofensas à Língua;

Ao meu filho, Tomás dos Reis Nogueira, formando da Escola Politécnica, que ajudou a espanar um pouco da poeira sobre algumas áreas do Cálculo requeridas pelo modelo de Putnam;

Ao amigo Loami Barros, pelas conversas esclarecedoras sobre análise de pontos de função;

Aos professores do IPT, pelo ganho de conhecimentos que pude obter durante o curso;

Ao Professor Doutor Mario Miyake, pelo apoio recebido ao longo do curso e preparação da dissertação;

A todos os meus mais sinceros agradecimentos.

RESUMO

Proposta de indicadores para a avaliação do progresso de desenvolvimento e qualidade de software

A literatura de Engenharia de Software descreve a dificuldade que os projetos de desenvolvimento de software enfrentam em concluir no tempo e orçamento previstos. Frequentemente, só muito próximo do final, os principais interessados no projeto – seus gerentes, usuários e corpo diretivo das organizações promotoras – se dão conta de que o produto não será entregue na data aprazada, apresenta problemas de qualidade ou são inviáveis.

O presente trabalho cria, a partir da lista dos riscos mais frequentes dos projetos, um conjunto de indicadores e investiga a possibilidade de utilizá-los para evidenciar a manifestação desses riscos. Esses indicadores, objetivos e quantificáveis, podem ser usados para medir o progresso do desenvolvimento e a qualidade do produto, ao longo do projeto, de forma a se obter uma avaliação realista de sua situação. O cálculo, análise e apresentação desses indicadores, de forma periódica, permitirão que situações de atraso, excesso de utilização de mão-de-obra e falta de qualidade sejam detectadas com antecipação, próximo à ocorrência do problema. Dessa forma, medidas corretivas podem ser tomadas para eliminar ou reduzir o impacto negativo nos projetos, com a definição de planos baseados em informações objetivas.

Os indicadores a serem usados para medir o progresso do desenvolvimento do software são escolhidos e especificados segundo os princípios e práticas da abordagem Practical Software Measurement (PSM). A técnica de contagem de pontos de função é utilizada para expressar tamanho do produto, e o modelo paramétrico desenvolvido por Putnam usado para estimativa com base no histórico de projetos da organização.

Partindo da análise das variáveis de risco de projeto, o trabalho conclui que é viável especificar um conjunto de indicadores que revelem a manifestação dos riscos. Para tanto, utilizou-se processo sistemático de construção, em que os riscos foram usados para definir as necessidades de informação do projeto, a partir das quais os indicadores foram gerados.

Palavras-Chave: Engenharia de software; controle de projetos; estimativa de tamanho de software; estimativa de esforço, tempo e qualidade; pontos de função; riscos dos projetos de desenvolvimento; modelo paramétrico de Putnam; Practical Software Measurement.

ABSTRACT

Proposal of indicators to assess software development progress and quality

The Software Engineering literature describes the difficulty that application development projects face to conclude on time and within budget. Very often, only close to the end, the main project stakeholders – its managers, users and the management of sponsor organizations – become aware that the product will not be delivered on the scheduled date, presents quality problems or is unfeasible.

Starting with a list of the more frequent project risks, this dissertation creates a set of indicators and investigates the feasibility of using them to reveal the manifestation of such risks. These indicators, objective and quantifiable, can be used to gauge the progress of the development and product quality as the project evolves, so that a realistic appraisal of the situation is obtained. The periodical calculation, analysis and communication of these indicators will permit that delays, excessive utilization of man-time and lack of quality be detected early, close to the inception of the problem. Prompt corrective measures can then be taken to eliminate or reduce the negative impact on the projects, including the definition of plans based on objective information.

The indicators to be used to measure software development progress are determined and specified according to the principles and practices of the Practical Software Measurement (PSM) approach. The function point counting technique is employed to express product size and the parametric model developed by Putnam is used for estimates based on the history of the projects of the organization.

By analyzing project risk variables, this dissertation concludes that it is feasible to specify a set of indicators that indicate the manifestation of the risks. A systematic construction process was used, in which the risks were employed to define project information needs from which the indicators were generated.

Keywords: Software Engineering; project control; software size estimation; effort, duration and quality estimation; function points; Putnam parametric model; software development risks; Practical Software Measurement.

Lista de ilustrações

Figura 1-1 – Metodologia de pesquisa adotada.....	17
Figura 2-1 – A abordagem dos processos cascata.....	22
Figura 2-2 – Interação entre fases e disciplinas.....	26
Figura 2-3 – Estrutura de um nível de maturidade na representação estagiada.....	29
Figura 2-4 – Exemplo de células com múltiplos itens.....	33
Figura 3-1 – Variação do parâmetro de produtividade.....	43
Figura 3-2 – Curva de Mão-de-obra de Rayleigh.....	46
Figura 3-3 – Variação de esforço em função do tempo de desenvolvimento.....	51
Figura 4-1 – Variáveis com correlação significativa com componentes de risco.....	57
Figura 4-2 – Fluxo do processo de criação de indicadores.....	58
Figura 4-3 – Matriz de variáveis de risco a avaliar.....	59
Figura 4-4 – O cone da incerteza.....	62
Figura 4-5 – Aplicabilidade dos indicadores no Processo Unificado.....	92
Figura 4-6 – Relevância dos conceitos para o modelo CMMI.....	92
Figura A-1 – Determinação dos parâmetros organizacionais.....	105
Figura A-2 Formulário de entrada de dados.....	106
Figura A-3 – Indicador de variação de funcionalidade (I-1).....	108
Figura A-4 – Indicador de variação de tamanho de produto (I-2).....	109
Figura A-5 – Indicador de atraso de projeto (I-3).....	110
Figura A-6 – Indicador de evolução de re-estimativas (I-4).....	111
Figura A-7 – Indicador de comparação entre esforço estimado e despendido (I-5).....	112
Figura A-8 – Indicador de acompanhamento de produtividade (I-6).....	113
Figura A-9 – Indicador de alocação de pessoal (I-7).....	114
Figura A-10 – Indicador de produção de linhas de código (I-8).....	115
Figura A-11 – Indicador de complexidade do projeto (I-9).....	116

Lista de tabelas

Tabela 2-1 – Áreas de processo do modelo CMMI.....	27
Tabela 2-2 – Subdivisão das áreas de processo.....	30
Tabela 2-3 – Estrutura de medição.....	32
Tabela 3-1 – Complexidade das funções de dados.....	38
Tabela 3-2 – Complexidade das funções de transação.....	39
Tabela 3-3 – Pesos para funções.....	39
Tabela 3-4 – Fatores de conversão entre pontos de função e linhas de código.....	41
Tabela 3-5 – Ajuste de Tamanho.....	42
Tabela 3-6 – Parâmetros de Produtividade de Processo.....	44
Tabela 3-7 – <i>Manpower Buildup Index</i>	48
Tabela 3-8 – Percentagens de tempo e esforço Modelo de Putnam.....	53
Tabela 3-9 – Percentagens de tempo e esforço COCOMO II.....	53
Tabela 4-1 – Risco de mudança constante de requisitos.....	61
Tabela 4-2 – Indicador de variação de funcionalidade.....	64
Tabela 4-3 – Risco de estimativa incorreta de tamanho.....	66
Tabela 4-4 – Indicador de variação de tamanho de produto.....	68
Tabela 4-5 – Risco de problemas de cronograma.....	69
Tabela 4-6 – Indicador de atraso de projeto.....	71
Tabela 4-7 – Risco de contínuas alterações de cronograma.....	73
Tabela 4-8 – Indicador de evolução de re-estimativas.....	74
Tabela 4-9 – Risco de custos estimados e reais em descompasso.....	76
Tabela 4-10 – Comparação entre esforço estimado e despendido.....	78
Tabela 4-11 – Risco de custos estimados e reais em descompasso.....	79
Tabela 4-12 – Indicador de acompanhamento de produtividade.....	81
Tabela 4-13 – Risco de estimativa de pessoal.....	83
Tabela 4-14 – Indicador de alocação de pessoal.....	85
Tabela 4-15 – Risco de uso uniforme do tempo do projeto.....	86
Tabela 4-16 – Indicador de produção de linhas de código.....	88
Tabela 4-17 – Risco de complexidade do projeto.....	89
Tabela 4-18 – Indicador de complexidade do projeto.....	91
Tabela 4-19 – Matriz de risco com indicadores.....	93

Lista de abreviaturas e siglas

AIE	Arquivo de Interface Externa
ALI	Arquivo Lógico Interno
ALR	Arquivo Lógico Referenciado
CAR	Causal Analysis and Resolution
CE	Consulta Externa
CM	Configuration Management
CMMI	Capability Maturity Model Integration
CMMI-DEV	CMMI for Development
COCOMO	Constructive Cost Model
DAR	Decision Analysis and Resolution
DER	Dado Elementar Referenciado
EE	Entrada Externa.
IFPUG	International Function Point Users Group
IPM	Integrated Project Management
ISBSG	International Software Benchmarking Standards Group
LOC	Lines of Code
MA	Measurement and Analysis
MBP	Manpower Buildup Parameter
MBI	Manpower Buildup Index
OID	Organizational Innovation and Deployment
OPD	Organizational Process Definition
OPF	Organizational Process Focus
OPP	Organizational Process Performance
OT	Organizational Training
PI	Product Integration
PMC	Project Monitoring and Control
PP	Parâmetro de Produtividade do Processo
PP	Project Planning
PPQA	Process and Product Quality Assurance
PSM	Practical Software Measurement
QPM	Quantitative Project Management
QSM	Quantitative Software Management, Inc.
RD	Requirements Development
REQM	Requirements Management
RISKM	Risk Management
RLR	Registro Lógico Referenciado
SAM	Supplier Agreement Management
SEI	Software Engineering Institute
SE	Saída Externa
TS	Technical Solution
UP	Processo Unificado
VAL	Validation
VER	Verification

Lista de símbolos

β	Fator de ajuste de tamanho de linhas de código
σ_F	Desvio padrão da variável F
E_{DEV}	Esforço de desenvolvimento (em homens-ano)
t_d	Tempo de desenvolvimento (em anos)

Sumário

1.	INTRODUÇÃO	12
1.1	Problema da pesquisa	13
1.2	Proposta do trabalho	14
1.3	Contribuição da pesquisa.....	15
1.4	Metodologia de trabalho.....	16
1.5	Organização do trabalho.....	18
1.6	Nota sobre terminologia	19
2.	PROCESSOS	21
2.1	Introdução.....	21
2.2	Processos de desenvolvimento de software.....	21
2.2.1	Processos seqüenciais (cascata).....	21
2.2.2	Processos iterativos.....	23
2.3	Processo Unificado.....	23
2.3.1	O ciclo de vida do Processo Unificado.....	24
2.4	CMMI.....	26
2.5	Practical Software Measurement (PSM)	30
3.	ESTIMATIVAS DE MEDIDAS DE DESENVOLVIMENTO.....	34
3.1	Introdução.....	34
3.2	Medidas de tamanho.....	35
3.2.1	Processo de contagem de pontos de função.....	36
3.2.2	As regras de contagem de pontos de função.....	36
3.2.3	Características gerais do sistema	39
3.2.4	Conversão de tamanho funcional para linhas de código	40
3.3	O modelo de Putnam	41
3.3.1	A equação de software de Putnam.....	41
3.3.1.1	Determinação da produtividade de processo	42
3.3.1.2	Dependência entre esforço e tempo de desenvolvimento	45
3.3.1.3	Interpretação da equação de software de Putnam.....	46
3.3.2	O modelo do Ciclo de Vida	46
3.3.2.1	Parâmetro de alocação de mão-de-obra	48
3.3.3	A solução paramétrica do modelo de Putnam	48
3.3.4	Comportamento do modelo de Putnam	50
3.3.5	Implementações do modelo de Putnam	51
3.4	Fases do ciclo de vida cobertas pelo modelo de Putnam	52
4.	INDICADORES DA FASE DE CONSTRUÇÃO.....	55
4.1	Introdução.....	55
4.2	Determinação das fontes de riscos dos projetos	55
4.3	Processo de criação de indicadores	57
4.4	Aplicação das limitações do trabalho.....	58
4.5	Definição dos Indicadores.....	60
4.5.1	Mudança constante de requisitos.....	61
4.5.2	Estimativa incorreta de tamanho	65
4.5.3	Problemas de cronograma	69
4.5.4	Alterações constantes de cronograma.....	72
4.5.5	Custos estimados e reais em descompasso	75

4.5.6	Utilização de recursos incompatível com data de conclusão do projeto	79
4.5.7	Estimativa de necessidade de pessoal.....	82
4.5.8	Uso uniforme do tempo do projeto.....	86
4.5.9	Dificuldade em gerenciar a complexidade do projeto.....	89
4.6	Enquadramento dos indicadores no ciclo de desenvolvimento de software e modelo CMMI.....	91
4.7	Apresentação do estado final da matriz de risco	92
5.	CONCLUSÃO	95
5.1	Retrospectiva do trabalho.....	95
5.2	Contribuição do trabalho.....	95
5.3	Limitações do trabalho	96
5.4	Possíveis extensões para o trabalho.....	97
	REFERÊNCIAS BIBLIOGRÁFICAS	99
	APÊNDICE A SIMULAÇÃO DE ACOMPANHAMENTO DE UM PROJETO	102
A.1.	Introdução.....	103
A.2.	Descrição do projeto hipotético.....	103
A.3.	Determinação dos parâmetros organizacionais	103
A.4.	Formulário de entrada de dados.....	105
A.5.	Indicador de variação de funcionalidade (I-1).....	107
A.6.	Indicador de variação de tamanho de produto (I-2)	108
A.7.	Indicador de atraso de projeto (I-3)	109
A.8.	Indicador de evolução de re-estimativas (I-4)	110
A.9.	Comparação entre esforço estimado e despendido (I-5)	111
A.10.	Indicador de acompanhamento de produtividade (I-6).....	112
A.11.	Indicador de alocação de pessoal (I-7)	113
A.12.	Indicador de produção de linhas de código (I-8).....	114
A.13.	Indicador de complexidade do projeto (I-9).....	115

1. INTRODUÇÃO

Relatórios periódicos do The Standish Group citados em (Royce, 1998), (Putnam, 2003) e (McConnell, 2006), dão conta de que, no período de 1994 a 2004, apenas cerca de 25% dos projetos de desenvolvimento de software foram concluídos no tempo e orçamento previstos. Aproximadamente 25% dos projetos foram cancelados no período, sem que os usuários tenham recebido qualquer produto, apesar dos investimentos efetuados. Metade de todos os projetos desenvolvidos foram concluídos com atraso ou acima do orçamento previsto.

Capers Jones (apud McConnell, 2006) reporta que, à medida que aumenta o tamanho do sistema em desenvolvimento, a porcentagem de projetos cancelados também aumenta, atingindo 65% para projetos acima de 100.000 pontos de função ou 10.000.000 de linhas de código.

A previsibilidade dos prazos, custos e resultados do desenvolvimento de software é, para muitos usuários, a característica mais valorizada em um projeto (Putnam, 2003). Entretanto, os usuários continuam a encontrar situações em que recebem continuamente relatórios de normalidade do andamento dos projetos, e, próximo do teste final de aceitação, recebem a informação de que o produto está atrasado, apresenta problemas de qualidade ou é inviável.

(McConnell, 2006) relata o otimismo e subjetividade com que os profissionais de desenvolvimento de sistemas elaboram estimativas e avaliações de *status* do projeto. Na mesma linha, (DeMarco, 1982) e, antes dele, (Brooks, 1975) ilustram os vários mecanismos usados por gerentes de software, para lidarem com o atraso dos projetos de forma a não revisar a data de entrega final do produto, seus custos ou funcionalidade.

Dessa forma, faz-se necessária a utilização de indicadores objetivos que meçam o progresso e a qualidade do produto periodicamente para determinar como os projetos estão se comportando em relação aos problemas e riscos que os afetam mais frequentemente. Essas medições devem ser utilizadas em avaliações de projeto a serem incluídas em seu plano. (McGarry, 2002) expõe como as medições podem ajudar a definir planos mais realistas, alocar recursos, acompanhar o progresso e desempenho do projeto em relação aos planos e tomar decisões baseadas em informações objetivas.

Tendo como motivação o cenário acima resumido, este capítulo introdutório formula o problema de pesquisa e a proposta de trabalho, para, em seguida, abordar a metodologia empregada e a organização do trabalho.

1.1 Problema da pesquisa

(Severino, 2002) declara que não se pode tratar de um tema sem que ele seja apresentado como um problema que descreva uma dificuldade a ser resolvida.

Segundo (Gil, 2002), o problema de pesquisa deve obedecer a uma série de requisitos em sua formulação:

- Ser expresso em forma de pergunta;
- Estar enunciado de forma clara e precisa;
- Ser de natureza empírica, evitando julgamentos de valor subjetivo;
- Ser passível de solução;
- Ser delimitado de forma a adequar-se aos meios disponíveis.

O problema de pesquisa estudado nessa dissertação enuncia-se da seguinte forma:

Pode um conjunto de indicadores objetivos e quantificáveis usados em avaliações periódicas de status de um projeto evidenciar a manifestação dos riscos mais frequentes com que os projetos se deparam em sua fase de construção?

Para que essa questão possa se formular de maneira empírica, faz-se necessário, inicialmente, identificar os riscos mais frequentes que comprometem os projetos. Em seguida, identificar medidas e indicadores que permitam saber como o projeto está conseguindo evitar ou controlar esses riscos.

A capacidade de definir, coletar ou calcular medidas e indicadores é compatível com a maturidade atingida por organizações avaliadas como CMMI nível 2, através da área de processo de Medição e Análise (Measurement and Analysis-MA). Essas organizações podem, a partir de suas necessidades de informação, no caso de gerenciamento de risco, identificar os seus objetivos de medição. Podem especificar medidas e indicadores e estabelecer procedimento de coleta, cálculo e análise dos dados, como parte da execução das práticas requeridas pelos dois objetivos específicos da área de processo MA (Ahern, 2003). Com isso, se depreende que o problema de pesquisa deste trabalho é passível de solução para as organizações que adotam processos de medição e análise em seus projetos, a exemplo das práticas de nível 2 do CMMI ou do Practical Software Measurement(PSM).

O presente trabalho limita os riscos para os quais são definidos indicadores àqueles que se relacionam com as atividades da fase de construção conforme definida pelo Processo

Unificado e com as principais disciplinas executadas nessa fase: implementação, teste, configuração e gerenciamento de mudanças e gerenciamento de projeto.

A limitação adicional imposta é que os riscos possam ser tratados pelo gerente do projeto ou por sua equipe, não se considerando riscos cujo tratamento requeira principalmente o envolvimento de pessoas ou entidades externas à equipe de projeto.

O trabalho foi desenvolvido a partir da perspectiva de sua adequação a projetos de médio porte, ou seja, projetos com duração de 3 a 12 meses e com equipe de desenvolvedores de 5 até 25 profissionais, segundo definição de (McConnell, 2006). Gerentes de projetos de pequeno porte podem dispensar a adoção dos indicadores definidos no trabalho por resultarem em esforço de controle elevado em relação ao esforço total do projeto. Já os gerentes de projeto de grande porte podem decidir-se por mecanismos mais elaborados dos que aqueles aqui apresentados.

Por outro lado, nenhum dos procedimentos adotados para a definição dos indicadores está restrito a um tipo específico de empresa, seja por sua atividade econômica, tamanho, ou por ter desenvolvimento de sistemas como sua atividade fim. Neste quesito, a aplicabilidade dos indicadores definidos não sofre limitação.

1.2 Proposta do trabalho

A presente dissertação de mestrado descreve a pesquisa de indicadores objetivos e quantificáveis para medir o progresso do desenvolvimento e a qualidade do produto, ao longo da evolução do projeto, de forma a se obter uma avaliação realista de sua situação, a partir dos riscos e problemas comumente enfrentados pelos projetos. Com isso, se busca responder ao problema de pesquisa enunciado em 1.1.

A pesquisa utiliza, como modelo de ciclo de vida de projeto, o Processo Unificado de Desenvolvimento de Software (Jacobson, 1999; Krutchen, 2003; Larman, 2001), que tem as seguintes principais características que facilitam a definição de indicadores objetivos e sua utilização no acompanhamento do projeto:

- Especificar requisitos na forma de casos de uso, cuja implementação e teste se podem acompanhar;
- Preconizar, desde suas fases e iterações iniciais, a criação de artefatos, inclusive código, que demonstrem a viabilidade da arquitetura, em torno da qual os casos de uso e o sistema evoluem;

- Defender que o andamento do projeto se mede pela quantidade de artefatos concluídos colocados sob controle de configuração;
- Subdividir-se em iterações que, desde o início, implementam casos de uso verificáveis pelo usuário.

Os indicadores para medir o progresso do desenvolvimento do software foram escolhidos e especificados segundo os princípios e práticas da abordagem Practical Software Measurement (PSM), descrita por (McGarry, 2002). Isso foi feito tomando como base as necessidades de informação dos gerentes do projeto e de outras pessoas interessadas em avaliar os projetos de forma factual e o menos subjetiva possível, tais como clientes e gerentes organizacionais.

A descrição dos indicadores foi feita através de estruturas de medição na forma preconizada pelo PSM. Essa especificação requer a definição de estimativas para várias medidas do produto ou do projeto, tais como estimativas de tamanho, esforço, duração. Essas estimativas são definidas a partir da técnica de contagem de pontos de função, para expressar tamanho do produto, e o modelo paramétrico desenvolvido por (Putnam, 1992). Esse modelo paramétrico se baseia no desempenho anterior dos projetos de uma organização e no tamanho do produto que se quer desenvolver para a criação das estimativas. Como organização, se denomina o departamento ou empresa em que os projetos são conduzidos, assumindo-se que os vários projetos do departamento ou empresa sejam conduzidos usando os mesmos processos, com equipes detentoras do mesmo grau de experiência.

As estruturas de medição dos indicadores especificam que as estimativas das medidas do produto e do projeto devem ser comparadas periodicamente com os valores medidos.

O presente trabalho inclui, em Apêndice, simulação do uso desses indicadores por um projeto hipotético. Isso se faz através de planilhas, que fazem uma implementação simplificada dos indicadores, representando-os inclusive em forma gráfica.

1.3 Contribuição da pesquisa

O PSM advoga que as necessidades de informação dos projetos – ou das organizações que os promovem – devem determinar a seleção das medidas de software que o projeto coletará e analisará (McGarry, 2002).

O presente trabalho utiliza os principais problemas e riscos do processo de desenvolvimento de software, de forma sistemática, como fonte de identificação dessas necessidades de informação e para a construção dos indicadores na forma preconizada pelo PSM.

Pretende-se que os indicadores definidos no presente trabalho – ou indicadores semelhantes criados com base nas práticas descritas – possam ser utilizados para efeito de acompanhamento dos projetos de desenvolvimento.

A utilização desses indicadores objetivos, em revisões freqüentes e periódicas de *status*, com a participação dos principais interessados no projeto, poderá contribuir para a detecção antecipada de problemas, permitindo a adoção de medidas para resolvê-los ou minorar suas conseqüências, com possíveis ajustes e melhorias de processos, prazo e funcionalidade. Disso poderão resultar projetos com maior produtividade e maior qualidade dos produtos gerados.

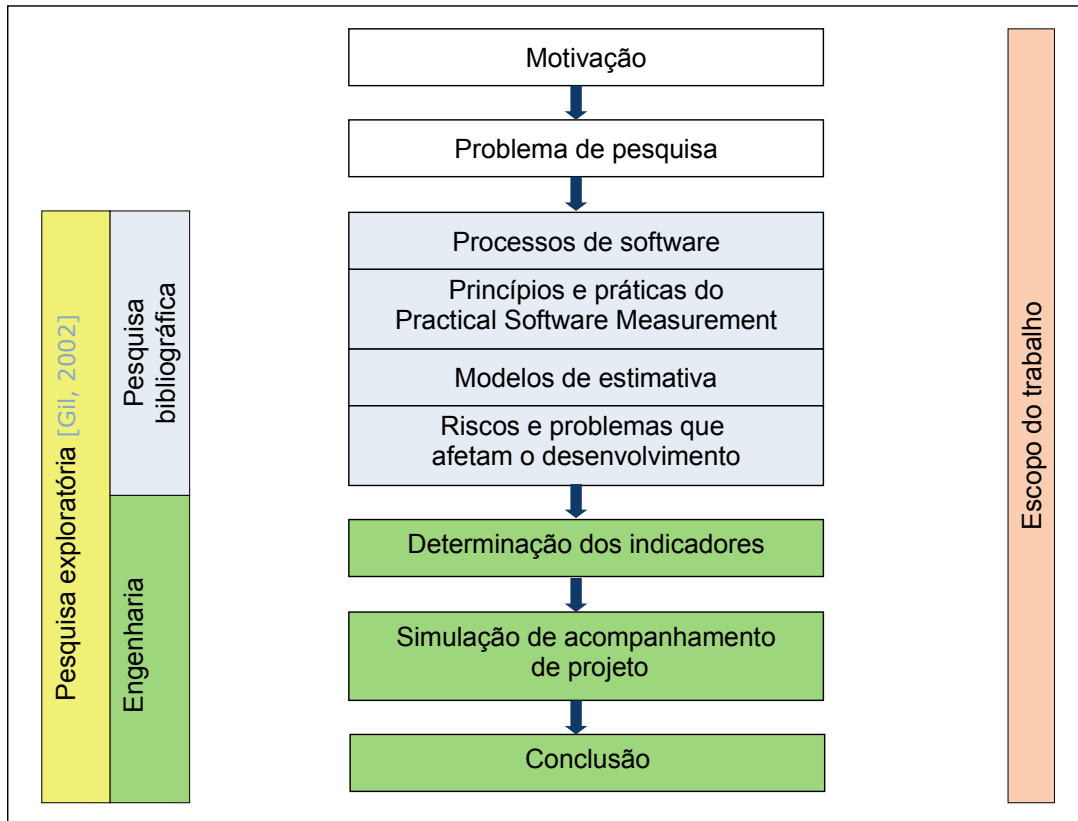
1.4 Metodologia de trabalho

Segundo (Gil, 2002, p.24), a determinação de resposta a questões de “como fazer algo de maneira eficiente” ou “como fazer as coisas”, à similaridade do problema levantado por este trabalho, constitui um problema de engenharia. Ainda, segundo o autor citado, a ciência pode fornecer inferência acerca do acerto das respostas, mas não responder diretamente às questões formuladas, ou seja, determinar qual o mecanismo que deve ser criado ou modificado para se conseguir o objetivo desejado.

(Gil, 2002, p41) também menciona que pesquisas exploratórias “têm como objetivo proporcionar maior familiaridade com o problema com vistas a torná-lo mais explícito ou a constituir hipóteses”.

A solução pretendida – determinação de indicadores a serem utilizados na revisão periódica de projetos de desenvolvimento para evidenciar a manifestação de riscos de desenvolvimento – foi, portanto, conseguida através de procedimentos característicos da pesquisa exploratória, com a adoção dos passos ilustrados na Figura 1-1 e descritos a seguir:

Figura 1-1 – Metodologia de pesquisa adotada



- Pesquisa bibliográfica para apreensão dos principais conceitos relacionados aos processos de software e modelos utilizados no decorrer do trabalho a fim de contextualizá-lo. Dentre os processos e modelos descritos estão: processo de desenvolvimento de software em cascata, iterativo, o Processo Unificado e o modelo de maturidade CMMI;
- Pesquisa bibliográfica para descrição dos princípios e práticas da abordagem Practical Software Measurement (PSM), utilizados na definição dos indicadores requeridos por este trabalho;
- Pesquisa bibliográfica para a determinação de medidas para estimativa de tamanho de software, bem como de modelo paramétricos para estimativas de esforço e duração de projeto para utilização no presente trabalho;
- Pesquisa bibliográfica para determinação dos principais riscos e problemas que afligem os projetos de desenvolvimento. A pesquisa a artigos científicos foi feita com o objetivo de identificar levantamentos de informações prévios feitos por outros pesquisadores, de forma que o trabalho se aproveite, de forma indireta, dos resultados

obtidos por essas pesquisas. (Gil, 2002) relaciona, entre as vantagens da pesquisa bibliográfica, a de permitir a cobertura de fenômenos de forma mais ampla do que poderia ser feita diretamente;

- Determinação dos indicadores a partir do exame sistemático dos riscos de projeto identificados na pesquisa bibliográfica, com a utilização dos modelos de estimativa pesquisados e da abordagem do PSM. Os indicadores são descritos através de estruturas de medição, buscando tornar sua definição precisa. O objetivo desta etapa é demonstrar que esses indicadores constituem um conjunto de medidas cujo uso permita detectar os riscos de projeto que serviram de base para sua definição.
- Simulação das atividades de acompanhamento de um projeto de desenvolvimento com a utilização dos indicadores determinados no passo anterior. A simulação é apresentada em apêndice e permite testar a viabilidade de determinação dos indicadores propostos e ilustrar sua utilização.

1.5 Organização do trabalho

Os capítulos que se seguem tratam dos seguintes assuntos, já abordados na metodologia do trabalho:

- Capítulo 2 – “Processos” descreve, em forma sucinta, os principais processos de engenharia de software utilizados ou referenciados no decorrer do trabalho a fim de contextualizá-lo;
- Capítulo 3 – “Estimativas de medidas de desenvolvimento” trata dos modelos que servem de base para criar estimativas para os indicadores que são usados no trabalho;
- Capítulo 4 – “Indicadores da fase de construção” toma como ponto de partida os principais riscos que afetam os projetos de desenvolvimento, para desenvolver os indicadores que serão usados para acompanhamento de projeto na fase de construção;
- Capítulo 5 – “Conclusão” faz resumo dos resultados obtidos com o trabalho e indica possíveis desdobramentos.

O Apêndice contém a simulação da utilização dos indicadores no acompanhamento de um projeto de desenvolvimento de médio porte.

1.6 Nota sobre terminologia

Adotam-se, neste trabalho, os termos “medida”, “medição”, “medir” e “indicadores”, com as seguintes definições extraídas de (McGarry, 2002):

- “**Medida**: uma variável à qual se assinala um valor para representar um ou mais atributos”, em inglês *measure*. Por extensão, também se denomina “medida” o resultado numérico de uma medição;
- “**Medição**: conjunto de operações como o objetivo de determinar o valor de uma medida”, em inglês *measurement*;
- “**Medir**: efetuar uma medição”, em inglês *measure*;
- “**Indicador**: uma medida que provê uma estimativa ou avaliação de atributos especificados com respeito a necessidades de informação definidas e derivados de um modelo de análise”. Por extensão, um gráfico, que contém várias medidas obtidas em geral ao longo do tempo, pode ser considerado um indicador.

As definições desses termos são consistentes com o padrão ISO/IEC Standard 15939-Software Measurement Process(2001) (apud McGarry, 2002). São igualmente as definições empregadas pelo modelo CMMI. A seguir, apresentam-se exemplos de uso desses termos consistentes com as definições apresentadas:

- (Florac, 1999): “... uma medida, a exemplo do número de defeitos reportados, que podemos usar para caracterizar um processo... (*... a measure such as the number of reported defects that we can use to characterize a process...*);”
- (Florac, 1999): “As atividades operacionais de medição iniciam-se com a coleta de dados. (*The operational activities of measurement start with collecting data*)”;
- (Chrissis, 2006): “Ao estabelecer objetivos de medição, os especialistas frequentemente pensam nos critérios necessários para se especificarem medidas. (*When establishing measurement objectives, experts often think ahead about necessary criteria for specifying measures...*);”
- (IFPUG, 2005): “O uso de pontos de função como uma medida do tamanho funcional de software.”

Entretanto, vários outros autores, alguns dos quais citados nesta dissertação, utilizam-se do termo “métrica” com o significado definido para o termo “medida” acima, ou seja, variável

que representa quanto um objeto ou processo possui de um determinado atributo. (IFPUG, 2005) utiliza o termo “métrica” como “uma combinação de duas ou mais medidas ou atributos.” São exemplos desses dois usos:

- (Boehm, 1980): (“o uso do conceito de instruções fontes escritas por homem-mês como métrica de produtividade”);
- (Putnam, 2003), no próprio título: (“Cinco Métricas Essenciais – *Five core metrics*”);
- (Humphrey, 1995): (“a definição de métricas toma tempo – *metrics definition takes time*”).

Neste trabalho, o termo “métricas” está restrito às situações em que seja necessário manter a fidelidade ao pensamento dos autores que o utilizam.

2. PROCESSOS

2.1 Introdução

O propósito deste capítulo é descrever, em forma sucinta, os principais conceitos relacionados aos processos de software utilizados no decorrer do trabalho a fim de contextualizá-lo. (Severino, 2002) alerta que não se pode falar de tudo em uma monografia, sendo importante ater-se ao substancial da pesquisa, evitando contextualizações muito amplas.

Este capítulo, portanto, é curto, mas possibilita que o trabalho possa ser lido e compreendido por pessoas interessadas que não tenham conhecimento prévio dos processos neles mencionados. Estão descritos, de forma resumida, conceitos referenciados no trabalho, relacionados a:

- Processos de desenvolvimento de software (seqüenciais e iterativos);
- O Processo Unificado;
- As áreas de processo do CMMI;
- Practical Software Measurement.

2.2 Processos de desenvolvimento de software

(Jacobson, 1999, p.4) define processo de desenvolvimento de software como “um conjunto de atividades necessárias para transformar requisitos de usuário em um sistema de software”.

Processos de desenvolvimento de software surgiram como resposta à forma *ad hoc* como se criava software nas décadas de 1950 e 1960, sem maiores preocupações com a adoção de atividades ou formas de trabalho que se repetissem entre as equipes de uma mesma organização (Larman, 2004).

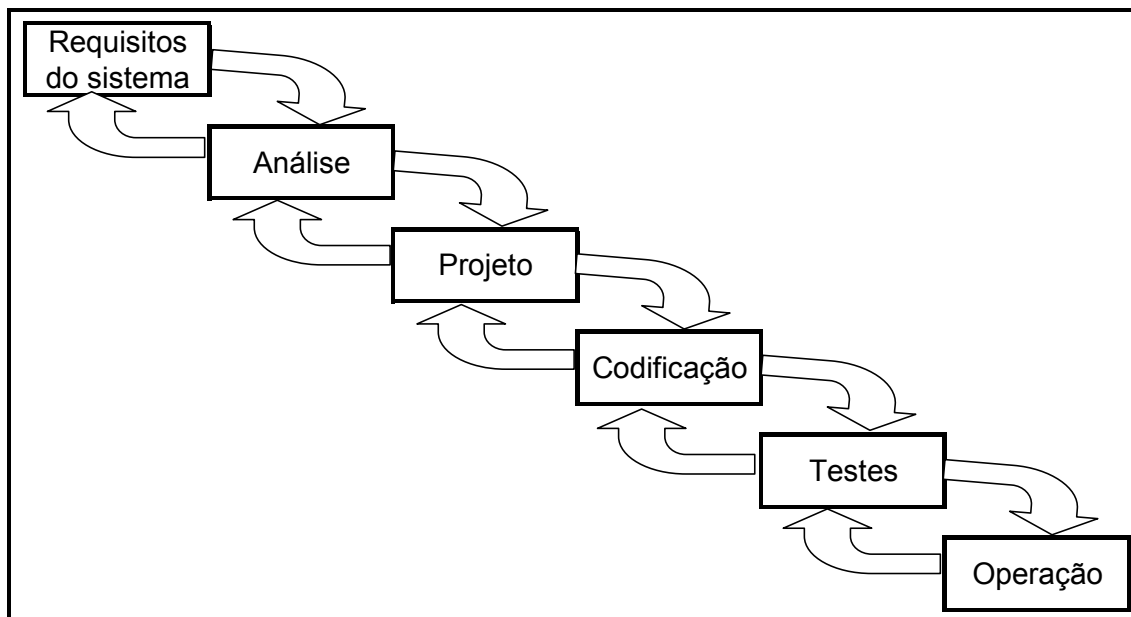
A seguir, se resumem os processos seqüenciais – a primeira modalidade de processos largamente adotada – e as inovações que ganharam divulgação na metade da década de 1980: os processos iterativos.

2.2.1 Processos seqüenciais (cascata)

Os processos caracterizados como seqüenciais, também conhecidos como cascata (*waterfall*), adotavam a abordagem de tratar o desenvolvimento de sistemas como uma série de atividades

e fases empreendidas uma seguida da outra. A Figura 2-1 mostra as fases usuais encontradas num projeto e ilustra a razão do nome cascata.

Figura 2-1 – A abordagem dos processos cascata



Adaptado de (Royce, 1998)

Embora o artigo mais citado com a proposta de desenvolvimento em cascata – Managing the Development of Large Scale Software Systems, de Winston Royce – preconizasse realimentação entre as diversas fases, a prática tradicional recomendava que as fases se sucedessem. Uma fase só se iniciaria com a conclusão da anterior. Em particular, os processos em cascata enfatizavam o detalhamento de todos os requisitos, com extensa documentação formal, antes que as fases seguintes iniciassem.

Os padrões para desenvolvimento e contratação, em especial os oriundos do Departamento de Defesa dos Estados Unidos (DoD-STD-2167), requeriam a utilização de práticas dos processos cascata (Larman, 2004), que são ainda largamente utilizadas no momento. (Pressman, 2001) admite seu uso em situações em que se tem garantia de que os requisitos são estáveis e não sofrerão modificações.

Em geral, as seguintes principais objeções são levantadas contra as práticas dos processos cascata (Larman, 2004) (Royce, 1998):

- Inadequado para projetos inovadores, em que os usuários não têm claros os requisitos do sistema ao seu início;

- Muitos dos riscos do projeto só se resolvem ao seu final, durante o teste de aceitação do usuário, quando é demorado e dispendioso promover alterações;
- Inadequação para lidar com mudanças em requisitos de forma ágil.

O uso de processos seqüenciais é favorecido pelos modelos de contratação de software a preço fixo, que dependem da utilização de disciplina de gerenciamento de projetos voltada ao cumprimento de um plano baseado em requisitos fixos, na forma de especificações assinadas pelos usuários (Poppendieck, 2003).

2.2.2 Processos iterativos

(Larman, 2004, p.9) define processo iterativo como a “abordagem de construção de software cujo ciclo de vida é composto de várias iterações em seqüência, sendo cada iteração um miniprojeto, composto de atividades de determinação de requisitos, projeto, programação e teste”. Dessa forma, em cada iteração, podem-se executar todas as atividades que nos processos cascata só se executam no projeto como um todo.

O objetivo de cada iteração, ou de um conjunto delas, é criar código que efetivamente funcione, possa ser testado pelo usuário, ou até mesmo ser colocado em produção. Dessa forma, o sistema vai crescendo, ou evoluindo, de forma incremental. A duração de cada iteração varia de cerca de duas semanas até três meses, dependendo do processo de software específico usado pela organização, do tamanho total do produto a desenvolver e da fase em que se encontra o projeto.

Dentre as principais práticas dos processos iterativos, podem-se citar (Larman, 2004):

- Medir o progresso do projeto através da quantidade de software produzido e testado, inclusive pelo usuário;
- Buscar resolver riscos do projeto mais cedo, através da antecipação da definição da arquitetura e codificação para os casos de uso mais controversos do projeto;
- Utilizar produto parcial produzido mais cedo para confirmar a adequação dos requisitos já definidos e facilitar a definição dos restantes.

2.3 Processo Unificado

(Jacobson, 1999) define o Processo Unificado (UP-Unified Process) como um processo de desenvolvimento de software, dotado das seguintes características:

Baseado em casos de uso. Para o UP, o caso de uso representa a funcionalidade do sistema que provê valor para o usuário e, portanto, deve direcionar todo o processo de desenvolvimento, desde especificação dos requisitos até a entrega do produto.

Centrado na arquitetura. O UP define a arquitetura como as diferentes visões do sistema que está sendo construído. A arquitetura evolui a partir dos casos de uso, na medida em que eles refletem as necessidades dos usuários, mas é também influenciada por outros fatores como a plataforma tecnológica do sistema e requisitos não funcionais, como o desempenho.

Iterativo e incremental. O UP preconiza a divisão do projeto em miniprojetos ou iterações. Cada iteração desenvolve um incremento ao produto, que, idealmente, pode ser testado pelo usuário e ser utilizado a partir daí. A ordem de desenvolvimento dos incrementos é influenciada pelo usuário e é feita de forma que os casos de uso que apresentam mais risco sejam desenvolvidos primeiramente. O desenvolvimento controlado através das iterações reduz o risco do projeto, uma vez que a identificação antecipada do risco permite a sua resolução a tempo e a um custo menor.

Através dessas características, o Processo Unificado facilita especificar os requisitos do sistema à medida que progride o desenvolvimento. Isso é importante, principalmente, naqueles casos em que os requisitos não estão claros desde o início.

2.3.1 O ciclo de vida do Processo Unificado

O ciclo de vida do Processo Unificado consiste de quatro fases, que se descrevem com seus principais focos de interesse (Krutchen, 2003):

Iniciação: entendimento dos requisitos gerais do sistema, a determinação dos objetivos do projeto e a sua primeira avaliação de esforço, duração e custo. Poderá incluir o detalhamento de casos de uso mais importantes e mesmo sua implementação, se necessária para validar a viabilidade do projeto, a escolha de uma arquitetura provável ou ajudar a definir requisitos importantes dos usuários. À conclusão da fase, diversos artefatos deverão estar criados ou iniciados: o documento de visão, o modelo de casos de uso, a demonstração de viabilidade econômica do projeto, avaliação de risco e o plano de projeto inicial, documentos esses que evoluirão ao longo das outras fases.

Elaboração: detalhamento dos requisitos, concomitante com a criação de uma sólida fundação de arquitetura e codificação de casos de uso que contribuam para a comprovação da arquitetura e eliminação dos riscos técnicos da solução. A fase conclui com o modelo de casos de uso próximo de estar completo, a descrição da arquitetura, um modelo executável que comprove a viabilidade da arquitetura e plano de desenvolvimento para o projeto.

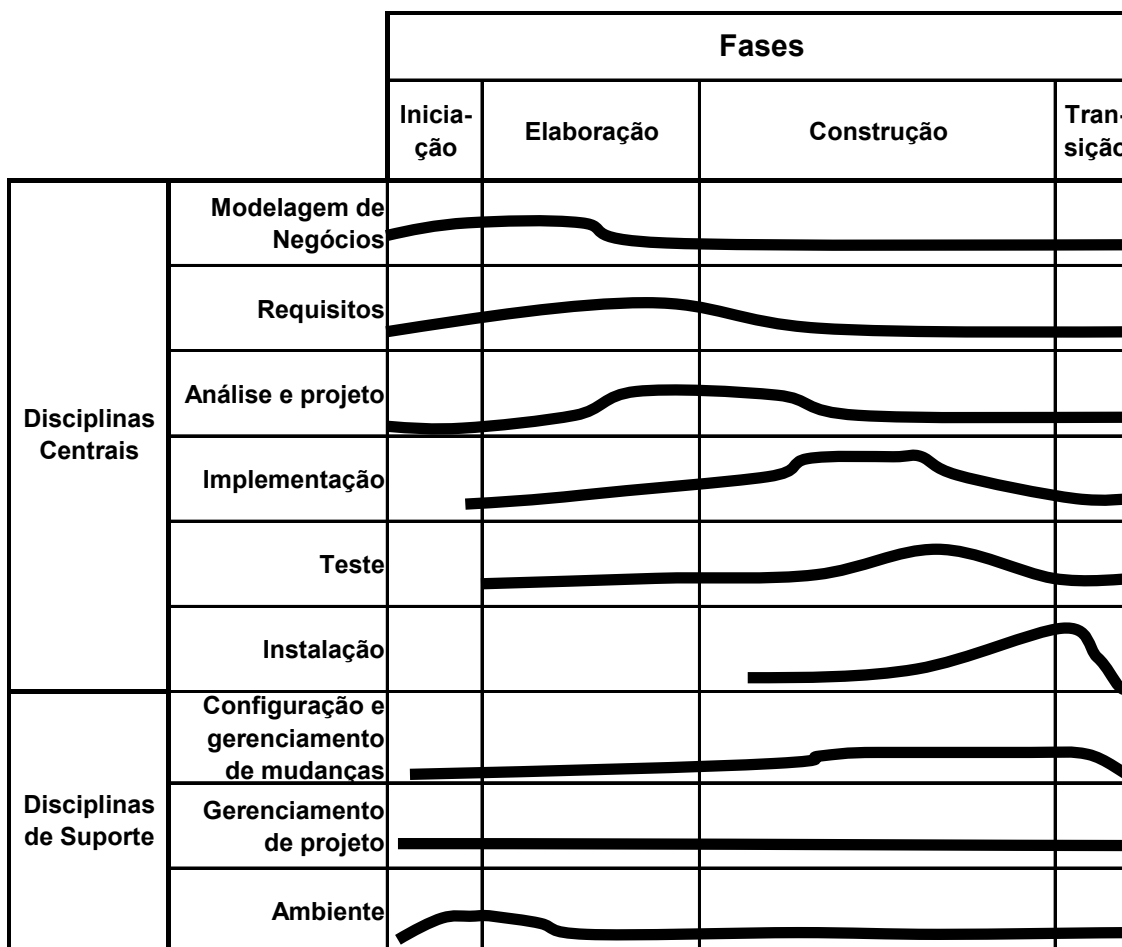
Construção: projeto e implementação da maior parte dos casos de uso, além da conclusão da definição de casos de uso restantes e dos detalhes da arquitetura. Deve concluir com um produto operacional, integrado e testado, complementado com documentação que descreva o *release* e documentação de usuário.

Transição: verificação da qualidade do produto, treinamento de usuários e instalação do produto para uso.

Cada fase termina em um *milestone*, que reflete a conclusão de determinados artefatos – planos, casos de uso, modelos, descrições de arquitetura, código fonte, documentação – bem como a tomada de decisão pelos principais integrantes do projeto sobre o desenvolvimento das futuras fases. Por ocasião do término de cada fase, também se controla o progresso do trabalho e se procede à re-estimativa de tempo e esforço para as fases seguintes. Cada uma das fases pode ser composta de várias iterações.

Em cada fase e iteração, se podem executar serviços de cada uma das disciplinas em que se agrupam as atividades técnicas e de suporte requeridas nas iterações. A Figura 2-2 indica a interação entre as fases e iterações, que no Processo Unificado representam a dimensão tempo, com as várias disciplinas, que representam as atividades que se executam em cada uma das fases e iterações.

Figura 2-2 – Interação entre fases e disciplinas



Adaptado de (Krutchen, 2003)

A diferente largura das colunas que representam as fases indica aproximadamente a duração destas em um projeto típico.

2.4 CMMI

(Fernandes, 2006, p.188) qualifica o CMMI (Capability Maturity Model Integration) como um modelo, cujo principal objetivo é “fornecer diretrizes para a melhoria de processos e habilidades organizacionais, com o foco no gerenciamento do desenvolvimento, aquisição e manutenção de produtos e serviços”. O modelo é publicado pelo Software Engineering Institute (SEI) da Carnegie Mellon University. A criação do modelo foi comissionada à SEI pelo Departamento de Defesa dos Estados Unidos, que buscava uma forma de qualificar seus fornecedores.

Em sua versão mais recente – V 1.2 – o modelo é subdividido em *constellations*, que (SEI, 2006) define como uma coleção de componentes que inclui um modelo, materiais de treinamento e documentação para avaliação. A *constellation* de interesse para este trabalho é CMMI for Development (CMMI-DEV), que corresponde, na versão 1.1 anterior, à publicação CMMI-SE/SW, para as disciplinas de engenharia de sistemas e engenharia de software. O CMMI-DEV contém práticas que cobrem gerência de projetos, gerência de processos, engenharia de sistemas, engenharia de hardware, engenharia de software e outros processos de suporte usados em desenvolvimento e manutenção (Chrissis, 2007).

O modelo CMMI, como seu predecessor, o modelo CMM-SW, também da SEI, diz o que as organizações devem fazer para a melhoria de seus processos de software em termos gerais, mas não como fazê-lo.

O modelo é subdividido hierarquicamente em (Chrissis, 2007):

Áreas de processo: um conjunto de práticas que coletivamente satisfazem um conjunto de objetivos necessários para melhorias na área. Há 22 áreas de processo no CMMI-DEV. Cada área é descrita por um propósito. A Tabela 2-1 descreve as áreas de processo do modelo através de suas declarações de propósito. Os nomes e abreviaturas das áreas de processo foram mantidos em inglês segundo a descrição original do modelo.

Tabela 2-1 – Áreas de processo do modelo CMMI

Abre- viatura	Área de Processo	Propósito
CAR	<i>Causal Analysis and Resolution</i>	Identificar causas dos defeitos e outros problemas e tomar ação corretiva para prevenir sua ocorrência no futuro.
CM	<i>Configuration Management</i>	Estabelecer e manter a integridade dos produtos de trabalho usando identificação, controle, <i>status</i> e auditoria de configuração.
DAR	<i>Decision Analysis and Resolution</i>	Analisar possíveis decisões usando processo formal de avaliação que considere alternativas identificadas em função de critérios estabelecidos.
IPM	<i>Integrated Project Management</i>	Estabelecer e gerenciar o projeto e o envolvimento dos interessados de acordo com processo integrado e definido customizado a partir do conjunto de processos padrão da organização.
MA	<i>Measurement and Analysis</i>	Desenvolver e sustentar uma capacitação de medição usada para atender às necessidade de informação da gerência.
OID	<i>Organizational Innovation and Deployment</i>	Selecionar e implementar melhorias incrementais e contínuas que quantitativamente melhorem processos e tecnologias da organização, em apoio aos objetivos de qualidade e de desempenho de processo derivados dos objetivos de negócios.
OPD	<i>Organizational Process Definition</i>	Estabelecer e manter um conjunto praticável de processos organizacionais e padrões ambientais de trabalho.

Abre- viatura	Área de Processo	Propósito
OPF	<i>Organizational Process Focus</i>	Planejar, implementar e disseminar melhorias de processos organizacionais com base em entendimento dos pontos fortes e fracos dos processos.
OPP	<i>Organizational Process Performance</i>	Estabelecer e manter entendimento quantitativo do desempenho do conjunto padrão de processos da organização e prover dados de desempenho de processo, <i>baselines</i> e modelos para gerenciar os projetos da organização.
OT	<i>Organizational Training</i>	Desenvolver as habilidades e conhecimento do pessoal para que possam desempenhar seu papel de forma eficaz e eficiente.
PI	<i>Product Integration</i>	Integrar o produto a partir de seus componentes, garantir que o produto funcione adequadamente conforme integrado e entregar o produto.
PMC	<i>Project Monitoring and Control</i>	Prover entendimento do progresso do projeto, de tal forma que ação corretiva apropriada seja tomada quando de desvios significativos do plano.
PP	<i>Project Planning</i>	Estabelecer e manter planos que definam as atividades do projeto.
PPQA	<i>Process and Product Quality Assurance</i>	Prover às equipes e gerência visibilidade dos processos e respectivos produtos de trabalho.
QPM	<i>Quantitative Project Management</i>	Gerenciar os processos definidos para o projeto de forma quantitativa para alcançar os objetivos de qualidade e desempenho de processos estabelecidos.
RD	<i>Requirements Development</i>	Produzir e analisar requisitos do cliente, produto e componentes do produto.
REQM	<i>Requirements Management</i>	Gerenciar os requisitos dos produtos do projeto e de componente dos produtos e identificar as inconsistências entre esses requisitos e os planos e produtos de trabalho do projeto.
RISKM	<i>Risk Management</i>	Identificar problemas antes que ocorram para que as atividades de gerenciamento de risco sejam planejadas e invocadas quando necessário no ciclo de vida do produto ou projeto para mitigar impactos no atingimento dos objetivos.
SAM	<i>Supplier Agreement Management</i>	Gerenciar a aquisição de produtos de fornecedores.
TS	<i>Technical Solution</i>	Projetar, desenvolver e implementar soluções para requisitos, incluindo produtos, componentes de produto e processos relacionados ao ciclo de vida do produto.
VAL	<i>Validation</i>	Demonstrar que os produtos ou componentes de produto cumpram o uso pretendido quando colocado no ambiente planejado.
VER	<i>Verification</i>	Garantir que os produtos do trabalho cumpram com os requisitos especificados.

Fonte: (SEI, 2006)

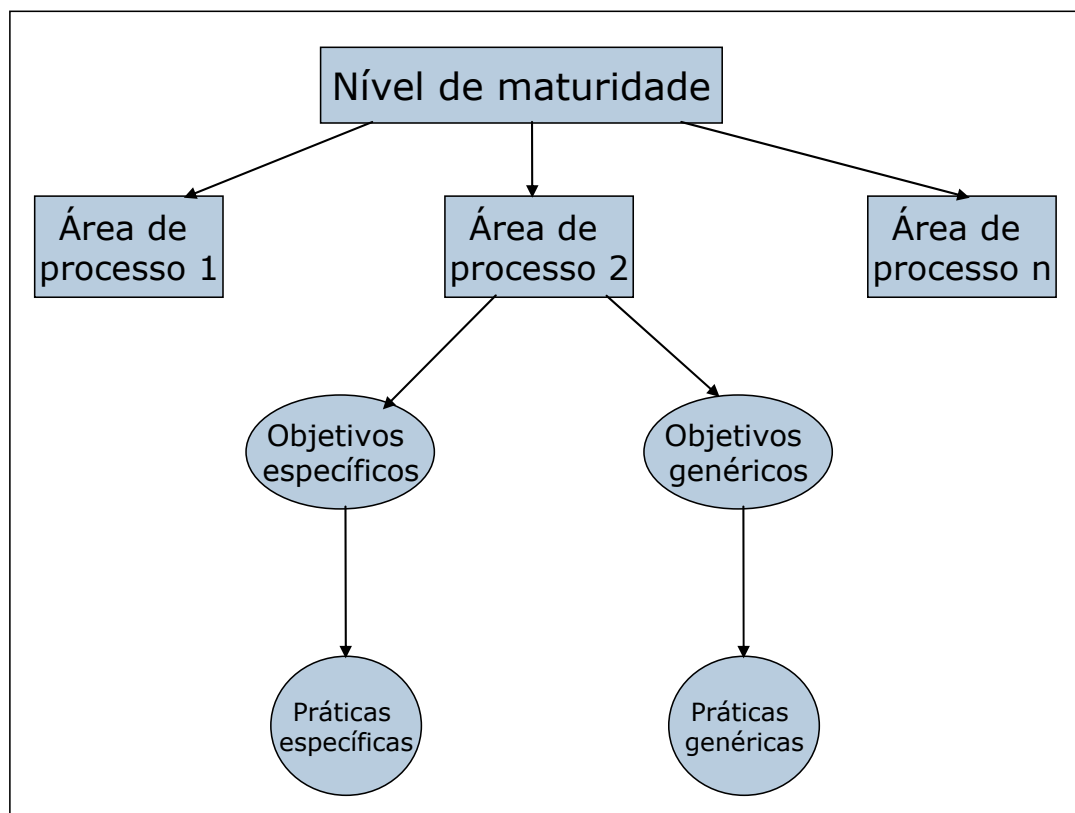
Objetivos (goals): podem ser específicos e genéricos. Os objetivos específicos descrevem as características específicas que devem ser obedecidas para satisfazer uma área de processo. Os objetivos genéricos, por sua vez, se aplicam a várias áreas de processo e, em geral, descrevem características que devem estar presentes para institucionalizar os processos de uma área de processo.

Práticas: que também podem ser específicas e genéricas, dependendo na natureza do objetivo no qual estão definidas. As práticas específicas são atividades que são necessárias para que se considere atingido um objetivo específico. Já as práticas genéricas são atividades necessárias para que se considere atingindo um objetivo genérico.

As organizações que escolhem o CMMI como modelo para melhoria de processos têm dois caminhos a seguir: a) a abordagem ou representação contínua, em que escolhem uma área de processo ou um subconjunto de áreas de processo no qual desejam progredir; b) a abordagem estagiada, que especifica a distribuição de conjuntos de áreas de processos em níveis de maturidade que devem ser alcançados em ordem seqüencial.

A Figura 2-3 ilustra como se compõe um nível de maturidade, com suas áreas de processo, objetivos genéricos e específicos e práticas.

Figura 2-3 – Estrutura de um nível de maturidade na representação estagiada



Adaptado de (SEI, 2006)

No caso da representação estagiada, uma organização diz-se ter sido avaliada em um determinado nível se atende a todas as práticas requeridas para o atingimento dos objetivos específicos e genéricos de todas as áreas de processo daquele nível.

As áreas de processo se dividem em 4 categorias por afinidade e na representação estagiada se enquadram em 4 níveis. A Tabela 2-2 indica como se subdividem as áreas de processo tanto por afinidade quanto por nível.

Tabela 2-2 – Subdivisão das áreas de processo

	Nível 2	Nível 3	Nível 4	Nível 5
Gerenciamento de Projetos	PP PMC SAM		QPM	
Engenharia	REQM	RD TS PI VER VAL		
Gerenciamento de Processos		OPF OPD OT IPM RISKM	OPP	OID
Suporte	MA CM PPAQ	DAR		CAR

Fonte: (SEI, 2006)

Empresas buscam obter avaliação no modelo CMMI, não só em função de estarem em um determinado nível requerido em processos de contratação – razão inicial da criação do modelo – mas, também, pelo ganho de produtividade e qualidade que a adoção das práticas do modelo traz. (Galín, 2006) elaborou levantamento bibliográfico de relatos comparando o desempenho de várias organizações antes e depois de mudanças de nível no modelo CMMI, e chegou à conclusão, com nível de significância de 1%, de que, efetivamente, o aumento de nível de maturidade no modelo traz ganhos de qualidade e produtividade.

2.5 Practical Software Measurement (PSM)

O presente trabalho utiliza os princípios e práticas do PSM, que definem as medidas a coletar, a calcular e a analisar, inclusive indicadores, com base nas necessidades de informação do projeto e dos indivíduos e entidades que com ele interagem. Segundo o PSM, adotam-se os seguintes passos para a identificação das medidas que são utilizadas como indicadores de projeto (McGarry, 2002):

- Identificação e priorização das necessidades de informação, a partir dos objetivos do projeto e das várias fontes de riscos e problemas que podem afetá-lo: cronograma, recursos, custos, tamanho do produto e sua estabilidade, qualidade, desempenho de processos, satisfação do cliente, entre outros;
- Mapeamento das necessidades de informação do projeto a conceitos mensuráveis, que atendam à necessidade de informação e que podem ser os sugeridos pelo PSM ou criados pelo projeto;
- Criação de estruturas de medição (*measurement constructs*) que implementem os conceitos mensuráveis de forma a atender à necessidade de informação;
- Identificação de indicadores, medidas primitivas ou medidas derivadas que se combinem nas estruturas para definirem o conceito mensurável.

Segundo (McGarry, 2002), indicadores são medidas que provêm a avaliação de determinados atributos em resposta a necessidades de informação. Medida primitiva é a forma mais elementar em que uma medida pode ser capturada. Medidas derivadas são calculadas a partir de medidas primitivas.

As necessidades de informação de um projeto podem estar ligadas a diferentes conceitos mensuráveis em cada fase do ciclo de vida de um projeto. (McGarry, 2002) exemplifica com o fato de que, durante a fase de construção, o acompanhamento de cronograma pode estar ligado aos artefatos concluídos, enquanto na fase de testes, o cronograma se ocupa de casos de testes que foram executados com sucesso.

A especificação de estruturas de medição provê a definição operacional das medidas. Neste trabalho, os indicadores são definidos utilizando o formato de estrutura de medição definido por (McGarry, 2002), descrito na Tabela 2-3, na qual se anota a descrição geral de cada um dos diferentes elementos.

Na descrição dos indicadores, de forma consistente com os *guidelines* do PSM, toma-se como ponto de partida a necessidade de informação da gerência do projeto, da gerência da organização e possivelmente dos usuários.

Tabela 2-3 – Estrutura de medição

Estrutura de medição	
Informação necessária	O que o usuário da medição precisa saber a fim de tomar decisões
Conceito mensurável	Conceito que atende à necessidade de informação
Entidades relevantes	Objetos que são medidos
Atributos	Propriedade de um objeto que é medida
Medidas primitivas	Medida relacionada a atributo definida por um método de medição.
Método de medição	Operações que definem o cálculo de medida primitiva
Tipo de método	O tipo poderá ser subjetivo ou objetivo
Escala	Conjunto de categorias ou valores usados na medida primitiva.
Tipo de escala	Tipo de relação na escala: nominal, ordinal, de intervalo ou razão
Unidade de medida	Unidade de medida da medida primitiva
Medidas derivadas	Medida derivada como função de uma ou mais medidas primitivas
Função de medição	Fórmula usada para calcular uma medida derivada
Indicador	Medidas ou gráficos que provêm a informação necessária
Modelo de análise	Função de tomada de decisão em resposta a valores dos indicadores
Critérios de decisão	Limites usados para deflagrar ações ou investigações
Aplicabilidade	Fase do processo unificado em que se aplica o indicador

Adaptado de (McGarry, 2002)

Nas estruturas de medição, quando as células de medidas primitivas e medidas derivadas contêm mais de um item, esses itens são numerados sequencialmente. Com isso, consegue-se relacionar o conteúdo das células seguintes ao item correto. Como exemplo, no caso da Figura 2-4, a célula de medidas derivadas contém quatro itens, que foram numerados para associá-los corretamente às respectivas funções de medição da célula seguinte.

Figura 2-4 – Exemplo de células com múltiplos itens

Medidas derivadas	<ol style="list-style-type: none"> 1. Tamanho 2. Esforço estimado (E_{DEV}) 3. Duração estimada (t_d) 4. Estimativa acumulada do esforço ($E(t)$)
Função de medição	<ol style="list-style-type: none"> 1. tamanho = funcionalidade modificada * <i>gearing factor</i> 2. $E_{DEV} = (0,39 \text{ MBP})^{4/7} PP^{-9/7} (\text{tamanho } \beta^{1/3})^{9/7}$ 3. $t_d = (0,39 \text{ MBP})^{-1/7} PP^{-3/7} (\text{tamanho} * \beta^{1/3})^{3/7}$ 4. $E(t) = (E_{DEV} / 0,39) [1 - \exp(-t^2 / (2 t_d^2))]]$

3. ESTIMATIVAS DE MEDIDAS DE DESENVOLVIMENTO

3.1 Introdução

Este capítulo dedica-se a expor, com algum nível de detalhe, conhecimentos requeridos para estimativa de projetos de desenvolvimento de aplicativos. Em particular, se descreve a técnica de Análise de Pontos de Função, que permite atribuir tamanho ao aplicativo e o modelo paramétrico de Putnam, que permite estimar esforço, duração e tamanho da equipe. Esses tópicos foram extensamente empregados no desenvolvimento da questão de pesquisa, o que motiva o nível de detalhe apresentado.

Em geral, os modelos paramétricos citados na literatura utilizam linhas de código lógicas (excluindo comentários) ou pontos de função como principal parâmetro de tamanho (Stutzke, 2005). Seria igualmente viável utilizar outro modelo paramétrico, a exemplo do COCOMO II (Constructive Cost Model), descrito em (Boehm, 2000), ou mesmo outros processos de estimativa. Uma vez que o particular modelo paramétrico utilizado não é determinante dos resultados que se pretende obter, a escolha foi influenciada pela familiaridade do Autor com o modelo escolhido.

Não sendo o principal objetivo deste trabalho tratar de modelos de estimativa, limita-se a descrever apenas uma técnica de medição de tamanho e um modelo de estimativa específico. Cabe aqui registrar que essas técnicas e modelos devem ser usados com cuidado, uma vez que nenhum modelo reflete completamente as características de um produto, nem dispensam profissionais experientes para utilizá-los. Para maior garantia da acurácia dos resultados dos modelos, deve-se calibrá-los com dados da própria organização (Humphrey, 1989).

Adicionalmente, pode-se utilizar mais de um modelo de estimativa, comparando seus resultados, e promovendo investigação das eventuais discrepâncias. (McGarry, 2002) descreve modelo de estimativa baseado nas atividades do projeto, ou *bottom-up estimation*. Nesse modelo, o projeto é subdividido em suas atividades e produtos, para os quais são desenvolvidas estimativas de esforço e duração, com base em dados históricos, analogia, ou julgamento de especialistas. As estimativas das partes são então adicionadas para se chegar ao resultado do projeto como um todo, que pode então ser comparado com a estimativa dos outros modelos paramétricos adotados.

3.2 Medidas de tamanho

Há muitas diferentes medidas de tamanho de software que podem ser usadas para caracterizar o tamanho do produto a ser desenvolvido. (Stutzke, 2005) descreve pontos de caso de uso, pontos de aplicação, pontos de função, linhas de código, entre outras.

Neste trabalho, utilizar-se-á a medida de pontos de função, por ser facilmente aplicável aos modelos paramétricos que são descritos a seguir e pelas seguintes outras razões (Garmus, 2001):

- É uma medida definida por norma (ISO/IEC 20926:2003 Software Engineering - IFPUG 4.1) e por organização sem fins lucrativos, o International Function Points Users Group (IFPUG);
- Mede o tamanho do software em termos compreensíveis ao usuário;
- Pode ser utilizada desde o início do projeto, antes até de a codificação ser iniciada.

A medida de pontos de função, por ser expressa em termos do domínio de aplicação compreensíveis para o usuário, é independente da implementação escolhida para a aplicação. Pode ser usada desde as primeiras fases do ciclo de vida do projeto, atendendo, deste ponto de vista, os requisitos desejáveis de um modelo de especificação de métricas para controle de projetos (DeMarco, 1982).

Da mesma forma que outras medidas que se poderiam utilizar, a medição de tamanho em pontos de função aumenta de precisão à medida que se reduz o grau de incerteza dos requisitos dos sistemas, ao longo do ciclo de vida do projeto (Boehm, 2000).

O presente trabalho utiliza essa característica de aumento de precisão com o tempo, ao preconizar que os projetos refinem as estimativas de seus principais parâmetros (esforço, tempo de desenvolvimento e distribuição de mão-de-obra) à medida que aumenta a precisão da estimativa de tamanho. (Garmus, 2001) estima que a técnica de contagem pode ter precisão de cerca de 50% ao se especificarem os requisitos e de 10% ao se concluir o projeto detalhado, o que reforça a necessidade de se refinarem as estimativas ao longo do projeto. Dessa forma, o presente trabalho recomenda que o tamanho do software em desenvolvimento seja novamente medido quando de mudanças de escopo e em *milestones* específicos do projeto, inclusive quando da definição da arquitetura e ao final do projeto.

Adicionalmente, a medida de pontos de função tem um grau de precisão caracterizado de forma empírica: dois contadores experimentados produzem contagens, a partir dos mesmos

requisitos, que diferem em cerca de 10% entre si (Stutzke, 2005), dado este que também é levado em conta neste trabalho.

3.2.1 Processo de contagem de pontos de função.

O ponto de partida para uma determinação de tamanho de software através de pontos de função é a existência de requisitos bem definidos, que podem tomar várias formas (Garmus, 2001), incluindo: casos de uso, diagramas de contexto, modelos entidade e relacionamento, diagramas de fluxo de dados, modelos de processos, diagramas de objetos, casos de teste, ou descrição textual dos requisitos. Entretanto, as técnicas de contagem foram estendidas para, no caso de haver indefinição de parte dos requisitos, ainda assim permitir contagem, embora com grau de imprecisão maior, útil principalmente nas fases iniciais do ciclo de vida.

A descrição abaixo do processo de contagem é feita do ponto de vista de um projeto de desenvolvimento de um novo sistema. Entretanto, gerentes de projeto necessitam estar cientes da possibilidade de utilizar o processo de contagem para melhorias (*enhancements*), inclusive quando de alteração significativa dos requisitos, ou eliminação de parcelas significativas do código já escrito.

3.2.2 As regras de contagem de pontos de função.

(Garmus, 2001) descreve os passos e regras de contagem com base na versão 4.1 do Function Point Counting Practices Manual publicado pelo IFPUG, que são:

1. Determinar o tipo de contagem;
2. Identificar as fronteiras/escopo;
3. Contar as funções de dados;
4. Contar as funções de transação;
5. Determinar os pontos de função não ajustados;
6. Determinar o fator de ajuste;
7. Calcular os pontos de função ajustados.

O primeiro passo é determinar se se trata de uma contagem inicial para um novo desenvolvimento, uma contagem para melhoria ou manutenção, ou contagem de um aplicativo existente, independentemente da história do seu desenvolvimento, para determinar seu tamanho.

O segundo passo do processo é a definição do escopo de contagem e fronteira da aplicação. O escopo poderá ser expresso em termos das funcionalidades que serão requeridas no produto. Já a fronteira é utilizada para delimitar a aplicação em consideração, separando-a de outras aplicações com que deva interagir e das funções executadas diretamente pelos usuários. O critério para determinar tanto escopo como fronteira deve ser o ponto de vista do usuário, e não considerações ligadas ao ambiente tecnológico, em linha com a definição dos requisitos desejáveis de um modelo de especificação de medidas mencionado acima.

O terceiro passo do processo é identificar e classificar, quanto à complexidade, as funções de dados existentes na aplicação. Há duas diferentes funções de dados sumariadas abaixo:

- Arquivo lógico interno (ALI): um grupo logicamente relacionado de dados ou informações de controle, reconhecido pelo usuário e mantido pelos processos da aplicação;
- Arquivo de interface externa (AIE): um grupo logicamente relacionado de dados ou informações de controle, existente fora da fronteira da aplicação, que é referenciado pelos processos da aplicação, mas mantido por outra aplicação externa.

Cada uma dessas diferentes funções é classificada em grau de complexidade – baixa, média ou alta – em função da:

- Quantidade de Tipos de Dados Elementares Referenciados (DERs) associados;
- Quantidade de Tipos de Registros Elementares (RLRs), considerando, como diferentes tipos de registros, cada subgrupo dos dados de um arquivo lógico, ou subclasse no caso de orientação a objeto.

Com essas regras, busca-se estimar o tamanho de forma objetiva contando-se elementos, ao invés de assinalar complexidade com base em julgamento subjetivo.

A Tabela 3-1, de dupla entrada, é utilizada para a determinação de complexidade dos dois tipos de função de dados:

Tabela 3-1 – Complexidade das funções de dados

Tipos de Registros Elementares (RLRs)	ALI / AIE	Tipos de Dados Elementares (DERs)		
		< 20	20 -50	> 50
1		Baixa	Baixa	Média
2 - 5		Baixa	Média	Alta
> 5		Média	Alta	Alta

Adaptado de (IFPUG, 2005)

O quarto passo do processo é identificar e classificar, quanto à complexidade, as funções de transação existentes na aplicação. Há três diferentes funções de transação:

- Entrada externa (EE): Processo elementar da aplicação que, recebendo dados do exterior, mantém arquivos internos à aplicação ou modifica seu comportamento;
- Saída externa (SE): Processo elementar da aplicação que envia dados ou informações de controle ao exterior da aplicação, e ao fazê-lo executa ao menos um dos seguintes processos: ou mantém arquivos internos à aplicação, ou modifica seu comportamento, ou efetua cálculos, ou cria dados derivados;
- Consulta externa (CE): Processo elementar da aplicação que envia dados ou informações de controle ao exterior da aplicação através da simples recuperação de dados, sem manter arquivos internos à aplicação, modificar seu comportamento, efetuar cálculos nem criar dados derivados.

Cada uma das diferentes funções é classificada em grau de complexidade – baixa, média ou alta – em função da:

- Quantidade de Tipos de Dados Elementares Referenciados (DERs) associados;
- Quantidade de Tipos de Arquivos Lógicos Referenciados (ALRs), tanto arquivos lógicos internos quanto arquivos de interface externa.

A Tabela 3-2, de dupla entrada, é utilizada para a determinação de complexidade dos três tipos de função de transação:

Tabela 3-2 – Complexidade das funções de transação

Tipos de arquivos referenciados (ALRs)	EE	Tipos de Dados Elementares (DERs)		
		< 5	5 - 15	> 15
< 2		Baixa	Baixa	Média
2		Baixa	Média	Alta
> 2		Média	Alta	Alta

tipos de arquivos referenciados (ALRs)	SE / CE	Tipos de Dados Elementares (DERs)		
		< 6	6- 19	> 19
< 2		Baixa	Baixa	Média
2 - 3		Baixa	Média	Alta
> 3		Média	Alta	Alta

Adaptado de (IFPUG, 2005)

O quinto passo é, com base na complexidade de cada função, atribuir-lhe pontuação e, em seguida, totalizar o número de pontos de função da aplicação. Para a atribuição de pontuação para os diferentes tipos de função, utiliza-se a Tabela 3-3:

Tabela 3-3 – Pesos para funções

Tipo da função		Complexidade		
		Baixa	Média	Alta
Tipo da função	Arquivo lógico interno (ALI)	7	10	15
	Arquivo de interface externa (AIE)	5	7	10
	Entrada externa (EE)	3	4	6
	Saída externa (SE)	4	5	7
	Consulta externa (CE)	3	4	6

Adaptado de (IFPUG, 2005)

O resultado obtido da soma dos pesos ou pontuações de cada função é denominado pontos de função não-ajustados na documentação do IFPUG.

3.2.3 Características gerais do sistema

As regras de contagem do IFPUG contêm fatores de ajustes para levar em consideração características não funcionais do sistema (tais como comunicações de dados, desempenho

requerido), o tipo de processamento utilizado (como, por exemplo, processamento distribuído, batch, on-line), características da plataforma de processamento (como taxa de transações), requisitos de usabilidade e reusabilidade, e facilidade de se promoverem mudanças no sistema. Há no total 14 fatores que determinam um índice, entre 0,65 e 1,35, que, multiplicado pelos pontos de função não-ajustados, gera o valor denominado pontos de função ajustados.

Esse novo valor ajustado pode ser então utilizado para cálculo dos parâmetros de esforço e duração do projeto, através da utilização de relações simplificadas de estimativa que assumem proporcionalidade entre os parâmetros do projeto e seu tamanho (McGarry, 2002). Para isso, se utiliza, como coeficiente de proporcionalidade, um fator de produtividade para a plataforma de desenvolvimento, obtido no mercado ou do histórico da organização que conduz o projeto.

Entretanto, neste trabalho, optou-se por não promover o ajuste, deixando de executar o sexto e o sétimo passos das regras de contagem do IFPUG, uma vez que os valores de esforço, duração e tamanho de equipe do projeto são gerados através da utilização de modelo parametrizado de Putnam, que se descreve abaixo. O mesmo se dá no modelo COCOMO II, que utiliza multiplicadores de esforço e fatores de escala para calcular as estimativas dos parâmetros do projeto.

3.2.4 Conversão de tamanho funcional para linhas de código

O modelo paramétrico de Putnam, que é utilizado neste trabalho, e o COCOMO II formulam suas equações tendo o tamanho do sistema como uma das variáveis independentes. Ambos exprimem tamanho como quantidade de linhas de código lógicas, excluindo comentários (Putnam, 1992) (Boehm, 2000). Para isso, os valores de pontos de função não-ajustados necessitam ser convertidos em linhas de código com a utilização de fatores de conversão determinados empiricamente. Esses fatores, chamados de fatores de *backfiring* ou *gearing factors*, podem ser obtidos de tabelas, ou, ainda melhor, calculados para cada organização, como a média da relação entre linhas de código e pontos de função dos projetos completados, segmentados por plataforma.

A Tabela 3-4, adaptada de (QSM, 2006), indica médias, medianas, valores máximos e mínimos de fatores de conversão para algumas linguagens de programação e tecnologias de uso mais frequente.

Tabela 3-4 – Fatores de conversão entre pontos de função e linhas de código

Linguagem/Tecnologia	Linhas de código/PF			
	Média	Mediana	Baixo	Alto
ASP	69	62	32	127
Assembler	172	157	86	320
C	148	104	9	704
C++	60	53	29	178
C#	59	59	51	66
COBOL	73	77	8	400
Java	60	59	14	97
JavaScript	56	54	44	65
JSP	59	-	-	-
Natural	60	52	22	141
Oracle	38	29	4	122
Oracle Dev 2K/FORMS	41/42	30	21/23	100
PeopleSoft	33	32	30	40
Perl	60	-	-	-
PL/1	59	58	22	92
PL/SQL	46	31	14	110
Visual Basic	50	42	14	276

Adaptado de (QSM, 2006)

3.3 O modelo de Putnam

Lawrence Putnam desenvolveu seu modelo já no final da década de 70 (Putnam, 1978) com base em evidências empíricas extraídas de projetos cujas informações coletou. Utilizou a observação de (Brooks, 1975) de que, em desenvolvimento de software, mão-de-obra e tempo não são intercambiáveis, não se podendo tomar a multiplicação desses dois fatores como uma constante na determinação do esforço de um projeto. Também se baseou na observação de pesquisadora que o precedeu de que o modelo de estimativa de recursos é complexo, não se prestando a uma simples relação linear de causa e efeito entre variáveis (Morin, L. apud Putnam, 1978). Seu modelo especifica duas equações principais: a primeira, relacionando esforço com tempo de desenvolvimento e tamanho do produto e a segunda, seu modelo de Ciclo de Vida, especificando a velocidade de adição de pessoal ao projeto através da função de densidade de probabilidade de Rayleigh.

3.3.1 A equação de software de Putnam

Análise das tendências observadas nos dados de projeto que coletou levou Putnam a propor a seguinte equação (Putnam, 1992):

$$\text{tamanho} = (E_{\text{DEV}} / \beta)^{1/3} t_d^{4/3} \text{ PP} \quad [3-1]$$

onde:

tamanho é representado pelo número de linhas de código do produto (LOC);

E_{DEV} é o número de homens-ano utilizados nas fases de construção principal do produto, incluindo projeto, codificação, integração, testes, documentação e as atividades de gerenciamento do projeto e configuração.

β é fator de ajuste de tamanho para levar em conta que as necessidades de integração, teste e gerenciamento aumentam com a complexidade introduzida pelo tamanho do sistema. (Putnam, 1992). A Tabela 3-5 contém os valores do fator a usar em função do tamanho do produto.

Tabela 3-5 – Ajuste de Tamanho

Tamanho (LOC)	Fator β
5-10K	0,16
20K	0,18
30K	0,28
40K	0,34
50K	0,37
>70k	0,39

Adaptado de (Putnam, 1992)

t_d é a duração, em anos, dos projetos de desenvolvimento medida desde o início da fase de projeto até a conclusão dos testes integrados.

PP é o parâmetro definido pelo modelo para caracterizar a produtividade e obtido pela calibração de projetos anteriores. Esse parâmetro difere do que Putnam chama de produtividade convencional definida como a divisão do tamanho pelo esforço.

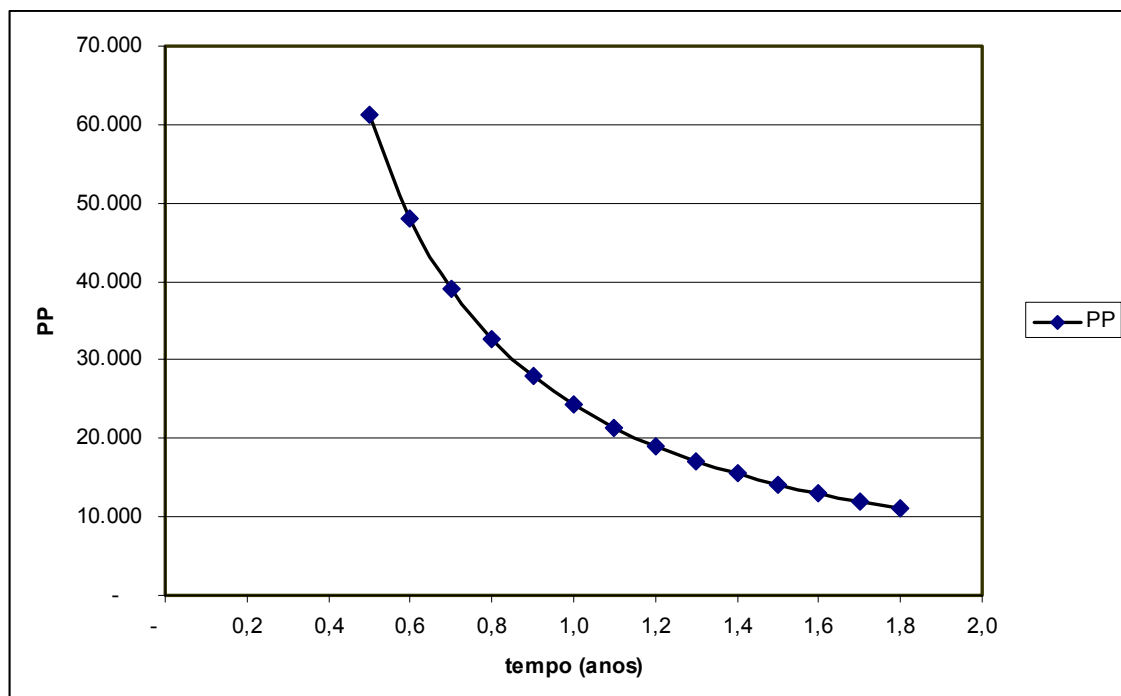
3.3.1.1 Determinação da produtividade de processo

A própria equação de software é utilizada para definir o significado do parâmetro de produtividade de processo. Colocando esse parâmetro em evidência, obtém-se:

$$PP = \text{tamanho} / [(E_{DEV} / \beta)^{1/3} t_d^{4/3}] \quad [3-2]$$

Com essa definição, o tempo de desenvolvimento alocado ao projeto influencia o parâmetro de produtividade do processo. Em outras palavras, dois projetos de mesmo tamanho desenvolvidos com o mesmo esforço têm o mesmo valor de coeficiente de produtividade convencional, geralmente expresso pelo número de horas por ponto de função. Entretanto, no modelo de Putnam, o projeto feito mais rapidamente exibe parâmetro de produtividade de processo maior. A Figura 3-1 exemplifica esse resultado, tomando como base um projeto de 1000 pontos de função em Java, com *gearing factor* de 60 linhas de código por ponto de função e esforço total de 9.000 horas. O gráfico indica o parâmetro de produtividade de processo atingido pelo projeto em função da duração do desenvolvimento.

Figura 3-1 – Variação do parâmetro de produtividade



Pode-se compreender esse resultado, inclusive de forma intuitiva, considerando-se que dentre os dois projetos que tiveram o mesmo coeficiente de produtividade convencional (e portanto o mesmo esforço), o que foi mais rápido, exibe maior domínio sobre o processo de produção de software.

Cada organização pode, com base nos projetos semelhantes (em termos de linha de produto, ambiente, experiência de equipe de desenvolvimento e utilização de ferramentas) que executou no passado, determinar seu próprio parâmetro de produtividade de processo médio e

sua variância. Isso, claro, é possível se os projetos anteriores colheram medidas de projeto de forma adequada, especialmente tamanho, esforço e duração. (Putnam, 1992) propõe que esse parâmetro de produtividade de processo pode ser usado para acompanhar o resultado dos esforços de melhoria de processo das organizações, bem como para fazer *benchmark* entre organizações diferentes.

Uma vez que os parâmetros de produtividade crescem de forma significativa em função dos diferentes fatores que afetam a produtividade, (Putnam, 1992) definiu um índice de produtividade (PI – Productivity Index), unicamente para efeito de referência, através da linearização da escala do parâmetro de produtividade. Também indicou, para cada um dos principais tipos de aplicação, o índice e o parâmetro de produtividade usuais com base em seu Banco de Dados de projetos. A Tabela 3-6 reproduz essas definições.

Tabela 3-6 – Parâmetros de Produtividade de Processo

Índice de produtividade	Parâmetro de produtividade	Tipo de aplicação
PI	PP	
1	754	
2	987	Microcódigo
3	1,220	
4	1,597	Firmware
5	1,974	Aviação embarcada
6	2,584	
7	3,194	Radar
8	4,181	
9	5,186	Controle de processo
10	6,765	
11	8,362	Telecomunicações
12	10,946	
13	13,530	Software de sistemas e científico
14	17,711	
15	21,892	
16	28,657	Sistemas de negócio
17	35,422	
18	46,368	
19	57,314	
20	75,025	
21	92,736	
22	121,393	
23	150,050	
24	196,418	
25	242,786	
26	317,811	
27	392,836	
28	514,229	
29	635,622	
30	832,040	

Adaptado de (Putnam, 1992)

Um alto valor do parâmetro de produtividade do processo está em geral associado aos seguintes fatores: conhecimento do domínio de aplicação, ambiente de desenvolvimento avançado, ferramentas eficientes, equipe treinada e experiente, maturidade em processos, gerenciamento adequado de projeto e comunicação efetiva. Na medida em que esses fatores, em geral, demandam tempo para evoluir, o parâmetro de produtividade tem certa inércia, evoluindo lentamente com o tempo. (Armour, 2006) cita relatório da QSM, a companhia que produziu a primeira implementação do modelo de Putnam, em que os projetos recentes de seu Banco de Dados exibem essencialmente a mesma produtividade alcançada pelos projetos do final da década de 80.

Uma vez que o parâmetro de produtividade do modelo de Putnam já incorpora o efeito dos fatores ambientais e características de equipe, devem-se utilizar pontos de função não ajustados para a conversão para linhas de código, para a determinação de tamanho do software.

Antes que se possa utilizar a equação de software, é necessário determinar o parâmetro de produtividade de processo. Na inexistência de dados internos, pode-se recorrer a banco de dados de projetos de outras organizações. O International Software Benchmarking Standards Group (ISBSG), organização sem fins lucrativos, provê repositório de projetos dos mais diferentes tipos de organizações, tipos de produto e ambientes, que pode ser usado para estimativa desse parâmetro. Seleção cuidadosa dos projetos a incluir na análise permitirá determinar o parâmetro de produtividade a utilizar, sendo uma alternativa aceitável antes que a própria organização possa determinar seu próprio parâmetro interno.

3.3.1.2 Dependência entre esforço e tempo de desenvolvimento

Outra consequência da equação de software de Putnam pode ser observada expressando-se o esforço em função da duração:

$$E_{DEV} = [\text{tamanho } \beta^{1/3} / PP]^3 / t_d^4 \quad [3-3]$$

Uma vez que os termos entre colchetes são constantes para um dado projeto, o modelo de Putnam atribui uma forte dependência entre esforço e a duração determinada para o projeto, na forma de inverso da quarta potência. Putnam deduziu esse expoente através de análise de regressão com transformação logarítmica. Essa quarta potência se reflete, entre outras situações, na elevação substancial do esforço ao se comprimir a duração dos projetos, por um lado, e na redução também significativa ao se planejar alongar o projeto.

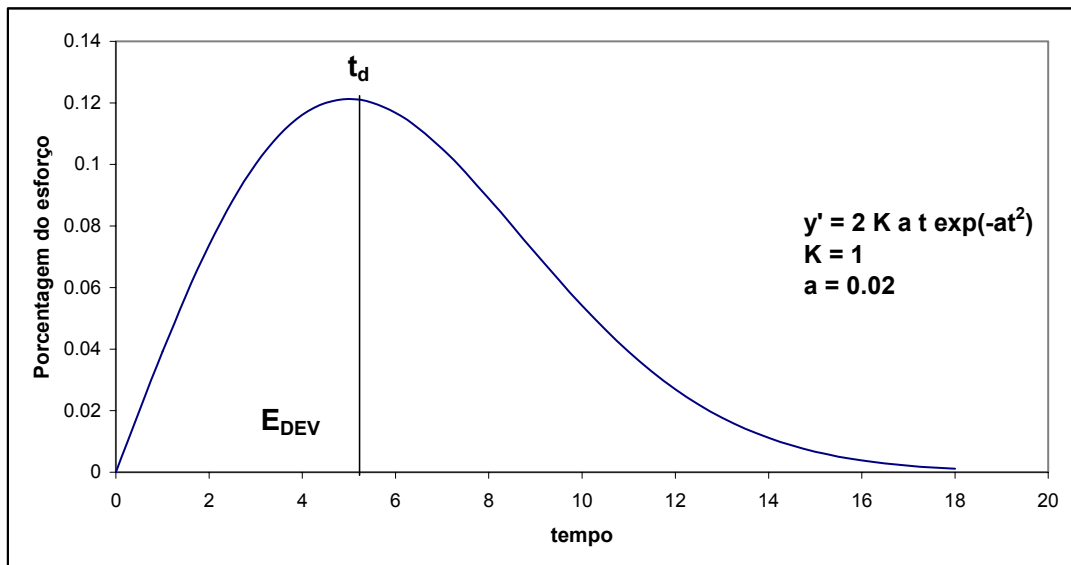
3.3.1.3 Interpretação da equação de software de Putnam

Uma vez determinados o parâmetro de produtividade de processo e o tamanho do produto em linhas de código, a equação de software [3-1] determina um conjunto infinito de pares de soluções para esforço e tempo de desenvolvimento. Há, portanto, necessidade de complementar o modelo para determinar as soluções que são utilizadas para o planejamento do projeto, o que Putnam faz com o uso seu modelo de Ciclo de Vida.

3.3.2 O modelo do Ciclo de Vida

(Putnam, 1978) observa que há uma taxa natural para utilização de pessoas em projetos, a taxa na qual as pessoas podem ser aplicadas de forma útil ao projeto. Para essa formulação, baseou-se no modelo teórico criado por Peter Norden, em 1963, que observou que o padrão de utilização de pessoal em projetos pode ser ajustado pela forma derivada da distribuição de Rayleigh, conforme ilustrado na Figura 3-2:

Figura 3-2 – Curva de Mão-de-obra de Rayleigh



Adaptado de (Putnam, 1992)

onde:

y' é o esforço por período, ou, em outros termos, a derivada do esforço acumulado. Sua dimensão é homens-ano/ano.

K é o esforço total gasto (incluindo a manutenção). Sua dimensão é homens-ano.

a é um parâmetro que indica a velocidade de adição de mão-de-obra do projeto.

t_d , o ponto máximo da curva e da utilização de mão-de-obra, na maioria dos projetos corresponde ao tempo em que o sistema adquire capacidade operacional, ou seja o tempo de desenvolvimento (excluindo-se a fase de determinação de requisitos). Sua dimensão é tempo.

(Putnam, 1992) traça, a partir da evidência empírica obtida dos projetos que pesquisou, associações entre os parâmetros da distribuição de Rayleigh e a aplicação de mão-de-obra no ciclo de vida de software, desde a fase de projeto (após a determinação de requisitos) até a fase de manutenção.

Da definição de t_d como ponto máximo, obtém-se a relação conectando o tempo de desenvolvimento e o parâmetro a :

$$t_d = (1/2a)^{1/2} \quad [3-4]$$

Integrando-se a forma derivada da equação de Rayleigh entre 0 e t_d , o tempo de desenvolvimento, obtém-se que:

$$K = E_{DEV} / 0,39 \quad [3-5],$$

onde

E_{DEV} representa o esforço desde o início da fase de projeto até o atingir-se capacidade operacional em t_d . Sua dimensão é homens-ano.

Essa relação indica que o esforço de desenvolvimento corresponde a 39% do esforço total dedicado ao sistema, ficando os 61% restantes alocados a manutenção, aí se incluindo a resolução dos defeitos encontrados após o sistema ser inicialmente colocado à disposição de seus usuários.

Substituindo-se [3-4] e [3-5] na equação derivada de Rayleigh chega-se à forma:

$$E'(t) = [E_{DEV} / (0,39 t_d^2)] t \exp(-t^2/2 t_d^2) \quad [3-6]$$

onde $E(t)$ representa o esforço acumulado no tempo t do projeto, e $E(t)'$ é a sua derivada.

Integrando, obtém-se a equação integral de Rayleigh, que dá no tempo t o esforço acumulado do projeto:

$$E(t) = (E_{DEV} / 0,39) [1 - \exp(-t^2 / (2 t_d^2))] \quad [3-7]$$

3.3.2.1 Parâmetro de alocação de mão-de-obra

(Putnam, 1978) observou empiricamente que os projetos de uma organização tendem a apresentar comportamento similar em termos de velocidade em que os sistemas se desenvolvem e no padrão de alocação de mão-de-obra. Caracterizou esse comportamento comum dos projetos de uma organização através de um parâmetro de alocação de mão-de-obra (*Manpower Buildup Parameter-MBP*), que definiu como:

$$MBP = E_{DEV} / (0,39 t_d^3) \quad [3-8]$$

Cada organização pode calcular seu próprio MBP, utilizando as medidas de esforço e tempo de desenvolvimento de seus projetos semelhantes do passado.

(Putnam, 1992) computou esse parâmetro para os projetos de seu Banco de Dados e, para efeitos de referência, criou índice paralelo (*Manpower Buildup Index – MBI*) para denotar valores selecionados do MBP e a caracterizar para velocidade de desenvolvimento associada. A Tabela 3-7 indica a correspondência entre MBI e MBP.

Tabela 3-7 – *Manpower Buildup Index*

MBI	MBP	Velocidade de alocação de mão-de-obra
1	7,3	Devagar
2	14,7	Moderadamente devagar
3	26,9	Moderada
4	55,0	Rápida
5	89,0	Muito rápida
6	233,0	Extremamente rápida

Adaptado de (Putnam, 1992)

Valores mais altos do parâmetro MBP levam à adição de mão-de-obra ao projeto, e reduzem a sua duração, o que pode ser visto através do exame da equação [3-8].

3.3.3 A solução paramétrica do modelo de Putnam

As equações [3-3] e [3-8], que são reproduzidas para referência,

$$E_{DEV} = [\text{tamanho } \beta^{1/3} / PP]^3 / t_d^4 \quad [3-3]$$

$$MBP = E_{DEV} / (0,39 t_d^3) \quad [3-8]$$

agora nos dão condições de resolver as medidas de duração e esforço do projeto, uma vez que as outras variáveis estão todas determinadas pelo sistema que se pretende desenvolver (no caso, o tamanho e o fator β relacionado), ou por características da organização e equipe de projeto determinadas pelas medições dos projetos anteriores (no caso, PP e MBP).

Obtêm-se as seguintes fórmulas para esforço (E_{DEV}) e tempo de desenvolvimento (t_d):

$$t_d = (0,39 MBP)^{-1/7} PP^{-3/7} (\text{tamanho } \beta^{1/3})^{3/7} \quad [3-9]$$

$$E_{DEV} = (0,39 MBP)^{4/7} PP^{-9/7} (\text{tamanho } \beta^{1/3})^{9/7} \quad [3-10]$$

lembrando que t_d está expresso em anos e E_{DEV} em homens-anos.

Os valores de tempo e esforço calculados acima são valores esperados. Como foram obtidos a partir de variáveis e parâmetros que carregam variação e imprecisão, é necessário calcular a variância dos resultados. A variância σ^2 de uma função de múltiplas variáveis $F(x_1, \dots, x_n)$ é dada por (Stutzke, 2005):

$$\sigma_F^2 = \sigma_{x_1}^2 \left(\frac{\partial F}{\partial x_1} \right)^2 + \dots + \sigma_{x_n}^2 \left(\frac{\partial F}{\partial x_n} \right)^2 \quad [3-11]$$

Por economia de espaço necessária pelo número de termos da fórmula acima, as equações [3-9] e [3-10] são transformadas com as seguintes mudanças no nome das variáveis: MBP \rightarrow M; PP \rightarrow P; tamanho \rightarrow T, resultando nas seguintes equações simplificadas:

$$t_d = 0,39^{-1/7} \beta^{1/7} M^{-1/7} P^{-3/7} T^{3/7}$$

$$E_{DEV} = 0,39^{4/7} \beta^{3/7} M^{4/7} P^{-9/7} T^{9/7}$$

Aplicando a relação [3-11] às funções de múltiplas variáveis t_d e E_{DEV} , obtêm-se suas variâncias:

$$\begin{aligned} \sigma_{t_d}^2 = & 0,39^{-2/7} \beta^{2/7} 1/49 [\sigma_M^2 (M^{-8/7} P^{-3/7} T^{3/7})^2 + \sigma_P^2 (3 M^{-1/7} P^{-10/7} T^{3/7})^2 + \\ & + \sigma_T^2 (3 M^{-1/7} P^{-3/7} T^{-4/7})^2] \end{aligned} \quad [3-12]$$

$$\sigma_{E_{DEV}}^2 = 0,39^{8/7} \beta^{6/7} 1/49 [\sigma_M^2 (4 M^{-3/7} P^{-9/7} T^{9/7})^2 + \sigma_P^2 (9 M^{4/7} P^{-16/7} T^{9/7})^2 +$$

$$+ \sigma_T^2 (9 M^{4/7} P^{-9/7} T^{2/7})^2] \quad [3-13]$$

Com os valores esperados e o desvio padrão obtido da variância, é possível determinar, utilizando-se a distribuição normal, a probabilidade de que uma dada estimativa de tempo de desenvolvimento e esforço não seja excedida. Também se pode utilizar o desvio padrão para expressar estimativas em forma de intervalo, em vez de valores pontuais, promovendo assim a noção de que se deve esperar variação nos resultados.

3.3.4 Comportamento do modelo de Putnam

Análise da equação [3-3] indica que o esforço tem dependência do inverso da quarta potência do tempo de desenvolvimento.

Essa dependência acarreta que, planejando-se uma redução do tempo de desenvolvimento, o esforço aumentará de forma significativa. Como exemplo: para que se possa planejar redução de tempo de projeto de 10%, deve-se prever alocação de mão-de-obra que eleve o esforço em 52%. Para tanto, deve ser possível ao projeto trabalhar com um MBP maior. (Putnam, 2003) estabelece limites para essa compressão, declarando haver um tempo de desenvolvimento mínimo para um projeto, uma vez estabelecidos o tamanho do produto, seus requisitos de qualidade e a produtividade de processo da equipe de desenvolvimento.

Por outro lado, aumentando o tempo de desenvolvimento, o esforço será consideravelmente reduzido. Como exemplo, aumentando-se o tempo em 10%, o esforço correspondente é reduzido em 32%. Nas equações [3-9] e [3-10] esse mesmo resultado seria obtido através da utilização de um MBP menor, correspondente à dilatação do tempo. (Putnam, 2003), da mesma forma, estabelece limites práticos para esse ganho de produtividade, uma vez que se delongando demais a conclusão do projeto, a equipe se reduz e se perdem habilidades necessárias à sua eficiente condução, além de se postergarem os benefícios econômicos com a implementação do sistema.

No modelo de Putnam, com sua forte dependência entre esforço e tempo, o gerente de projeto deve planejar a alocação de sua equipe de forma a equilibrar esses dois fatores levando em conta as necessidades de seus clientes.

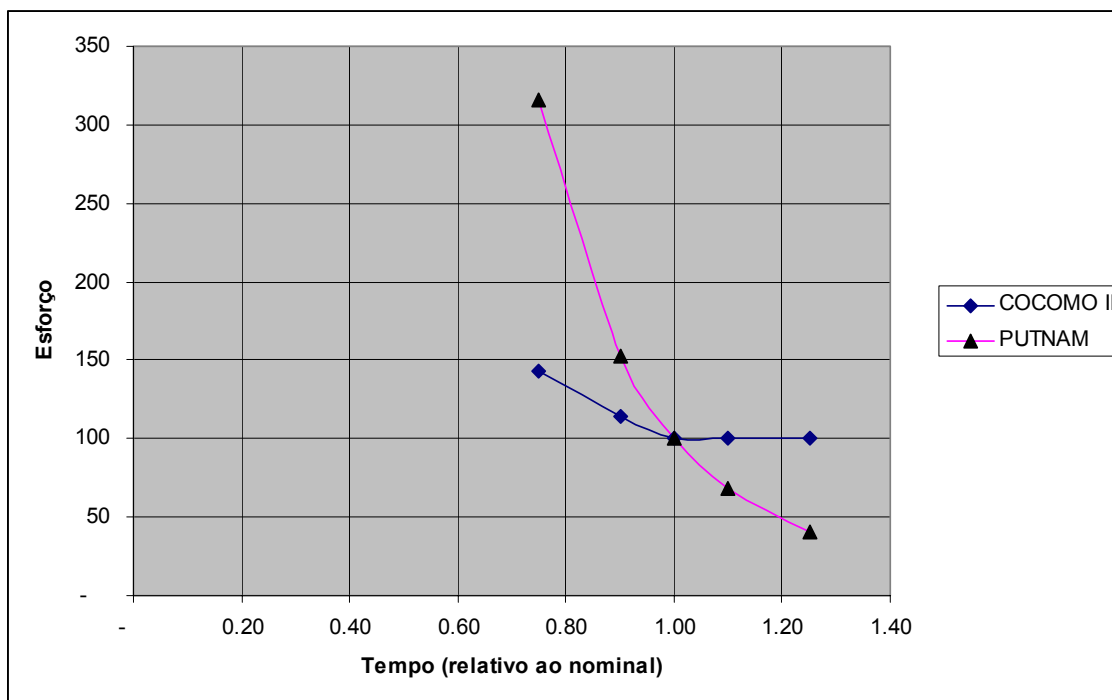
A dependência entre tempo e esforço também é reconhecida em outros modelos paramétricos de estimativa (McConnell, 2006). Como exemplo, o modelo COCOMO II (Boehm, 2000), também, reconhece haver um tempo mínimo de desenvolvimento, estabelecendo um limite prático de compressão de tempo de 25% em relação ao valor nominal calculado pelo modelo.

A dependência entre esforço e tempo é, contudo, menos acentuada nesse modelo: com esse valor máximo de compressão de tempo, 25%, o esforço do projeto aumentaria em 43% (enquanto no modelo de Putnam, para essa mesma compressão, o esforço aumentaria em 216%, possivelmente inviabilizando o projeto).

Por outro lado, o modelo COCOMO II não confere qualquer redução de esforço ao projeto, planejando-se extensão de sua duração.

A Figura 3-3 mostra o diferente comportamento dos dois modelos citados à compressão e extensão do tempo.

Figura 3-3 – Variação de esforço em função do tempo de desenvolvimento



Adaptado de (McConnell, 2006)

3.3.5 Implementações do modelo de Putnam

No desenvolvimento deste trabalho, as estimativas necessárias são feitas com as equações paramétricas acima. Cabe notar, entretanto, haver implementações, tanto comerciais como disponíveis livremente, do modelo de Putnam.

Essas implementações podem facilitar o trabalho do gerente de projeto ou analista de medição, permitindo-lhes obter os parâmetros de produtividade de processo (PP) e de alocação de mão-de-obra (MBP), a partir do repositório de projetos com o qual a ferramenta

foi calibrada. Também permitem ao analista de medição criar repositório de projetos da organização, que serão usados para calcular os parâmetros PP e MBP locais, para uso nos projetos futuros.

(McConnell, 2006) contém lista das ferramentas de estimativas mais empregadas no momento, destacando, dentre as que seguem o modelo de Putnam:

- SLIM, ferramenta comercial original da QSM criada por Putnam.
- Contrux Estimate, ferramenta disponível livremente, que combina o modelo de Putnam com o uso dos multiplicadores de esforço do modelo COCOMO II.

Essas duas ferramentas têm em comum o uso da técnica de simulação de Monte Carlo, para a construção de diferentes soluções a partir das distribuições de probabilidade ligadas à incerteza do tamanho do sistema e dos parâmetros PP e MBP. Os valores de esforço e tempo, bem como suas variações, são então obtidos da análise das várias soluções geradas através da simulação. Essas ferramentas podem escolher os parâmetros PP e MBP com base em informações providas pelos usuários, que dessa forma não precisam ter conhecimento profundo do modelo para utilizá-las.

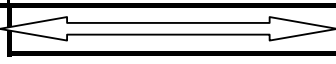
3.4 Fases do ciclo de vida cobertas pelo modelo de Putnam

Na formulação original do seu modelo (Putnam, 1978 e Putnam, 1992), a equação de software foi aplicada ao *main build*, ou seja, às fases de projeto, codificação, integração e testes, seguindo o ciclo de vida do modelo cascata do desenvolvimento do software. As fases de estudo de viabilidade e projeto funcional (requisitos) eram cobertas através de extensão dos valores calculados para o *main build*. Calculavam-se o esforço e duração para essas fases como uma porcentagem do esforço e duração respectivamente despendidos com o *main build*. Já (Putnam, 2003) trabalha tanto com projetos que seguem o modelo cascata quanto com os que se utilizam do Processo Unificado. Apesar das diferenças conceituais marcantes entre os dois modelos, há equivalência uma a uma das diferentes fases.

A Tabela 3-8, calculada com base em dados extraídos de (Putnam, 2003), indica duração e esforço das outras fases em relação ao *main build* ou construção, normalizados como 100, tanto para o modelo do ciclo de vida em cascata ou do Processo Unificado:

Tabela 3-8 – Percentagens de tempo e esforço
Modelo de Putnam

Cascata	Estudo de viabilidade	Projeto funcional (requisitos)	<i>Main build</i> (projeto, código, integração e testes)	Implantação
Processo Unificado	Iniciação	Elaboração	Construção	Transição
Tempo	25%	30-35%	100	25%
Esforço	5-10%	20%	100	



Captura pelo modelo de Putnam

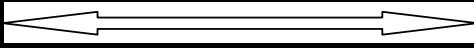
Adaptado de (Putnam, 2003)

A Tabela 3-9, criada a partir de dados de (Boehm, 2000), com normalização pela duração e esforço da fase de construção, permite comparar os modelos de Putnam e COCOMO II:

Tabela 3-9 – Percentagens de tempo e esforço
COCOMO II

Cascata		Planos e requisitos	Projeto do produto	Projeto detalhado, código, integração e testes	Implantação
	Tempo	27%	35%	100	18%
	Esforço	10%	20%	100	12%

Processo Unificado		Iniciação	Elaboração	Construção	Transição
	Tempo	20%	60%	100	20%
	Esforço	8%	31%	100	15%



Captura pelo COCOMO II

Adaptado de (Boehm, 2000)

Apesar das diferenças conceituais marcantes entre o modelo de Putnam e o COCOMO II, os números tabelados não diferem de forma significativa para as fases equivalentes, à exceção da duração da fase de elaboração do projeto no Processo Unificado.

Neste trabalho, os indicadores que são usados para o acompanhamento e controle dos projetos dão ênfase aos artefatos e processos que preponderam na fase de construção. No Processo Unificado já há atividades de implementação e teste nas fases anteriores à construção, mas é nessa fase em que se concentra a maioria do esforço ligado a essas disciplinas ou *workflows* de implementação e teste. A fase de construção abrange, segundo (Jacobson, 1999), tipicamente, 60 a 65% do esforço do desenvolvimento e 40 a 50% da duração do projeto, indicando que o modelo de Putnam já cobre a maioria do esforço despendido no desenvolvimento.

As fases anteriores, correspondentes às primeiras iterações do projeto, podem ser controladas e acompanhadas, conjuntamente por usuários e desenvolvedores, através da modalidade de *timeboxing* (Larman, 2004). Não havendo requisitos bem desenvolvidos que se possam utilizar no processo de estimativa, os participantes do projeto fixam tanto o tempo e quanto o esforço que utilizarão nas primeiras iterações das fases de iniciação ou elaboração. Essas iterações servirão, principalmente, para o detalhamento dos requisitos, e para o desenvolvimento de programação que sirva de prova de conceito para demonstração de viabilidade da solução.

De forma similar, tanto o modelo de Putnam, quanto o COCOMO II não cobrem diretamente a fase de transição ou implantação. Há grandes diferenças de esforço e tempo para essas atividades, em função de quantidades de usuários, locais de instalação do produto, atividades de treinamento, etc.

4. INDICADORES DA FASE DE CONSTRUÇÃO

4.1 Introdução

Neste capítulo, inicialmente, descreve-se o processo pelo qual se identificaram os principais componentes de riscos que afetam os projetos de desenvolvimento, com os quais os gerentes de projetos se devem ocupar. Esses componentes de riscos são, então, transformados em indagações que visam a descobrir se os problemas e riscos se estão manifestando nos projetos que se devem acompanhar. Essas indagações constituem, então, no contexto deste trabalho, as necessidades de informação, que segundo o PSM, permitem determinar as medidas que os projetos devem coletar e indicadores a determinar e analisar.

Através do cálculo, análise e apresentação desses indicadores, pode-se verificar a ocorrência da manifestação dos riscos nos projetos, consistindo, portanto, na resposta à questão de pesquisa, como se pretende demonstrar.

4.2 Determinação das fontes de riscos dos projetos

A disciplina de gerenciamento de riscos, uma das áreas de processo do CMMI, no nível 3, pode ser usada para identificar as principais causas de problemas que afligem os projetos. (DeMarco, 2003, p.16) descreve um risco como “um possível evento futuro que leve a um resultado indesejado”. (Ropponen, 2000, p.), por sua vez, descreve risco como “um aspecto particular ou propriedade de uma tarefa de desenvolvimento, processo ou ambiente que, se ignorado, aumentará a probabilidade de fracasso do projeto”.

(DeMarco, 2003) descreve um processo de identificação de riscos, que pode ser usado por qualquer organização, através da enumeração das falhas encontradas no histórico de projetos já realizados. As organizações que se empenham em melhorias de processos podem utilizar o seu processo de análise de conclusão de projetos, ou *postmortem*, para identificar onde os projetos encerrados se desviaram de seu planejamento de prazo, esforço, custos e qualidade de produto e identificar as causas raiz que levaram a esses desvios. Essas causas raiz constituem os riscos que devem ser analisados pelos gerentes de projeto da organização para possível inclusão na lista de riscos do projeto. O gerenciamento de riscos, além de ser prática essencial no Processo Unificado, que se define como orientado a risco (Jacobson, 1999), é também prática obrigatória para as organizações que pretendem atingir o nível de maturidade 3 no modelo CMMI (Chrissis, 2007).

Outra forma de identificar riscos, ou seja, os possíveis problemas que os projetos podem vir a enfrentar, é a referência à bibliografia existente, livros ou periódicos, que foi a abordagem utilizada por este trabalho.

(Ropponen, 2000), partindo de uma lista inicial dos 10 riscos mais freqüentes encontrados em projetos, criada empiricamente por Barry Boehm, desenvolveu trabalho de consolidação e ortogonalização para chegar a uma lista de 6 componentes independentes de risco:

- Cronograma e duração;
- Funcionalidades do sistema;
- Subcontratação;
- Gerenciamento de requisitos;
- Uso de recursos e desempenho;
- Gerenciamento de pessoal.

Em seguida, o estudo piloto descrito na obra citada identificou, através de questionários de pesquisas recebidos de 83 gerentes, responsáveis por cerca de 1.100 projetos, uma lista de 19 itens ou variáveis de risco, que foram categorizados em relação aos componentes acima citados. A matriz da Figura 4-1 indica essas 19 variáveis, e a que componentes de risco contribuem de forma significativa, através do preenchimento da célula da matriz, que se situa na intersecção da linha que representa a variável de risco e a coluna que representa o componente de risco.

Figura 4-1 – Variáveis com correlação significativa com componentes de risco

Variáveis de risco	Componentes de risco					
	Cronograma e duração	Funcionalidades do sistema	Subcontratação	Gerenciamento de requisitos	Uso de recursos e desempenho	Gerenciamento de pessoal
Deficiências de pessoal						■
Problemas de cronograma	■					
Utilização de recursos incompatível com data de conclusão					■	
Custos estimados e custos reais em descompasso	■					
Estimativa incorreta de tamanho	■					
Estimativa de necessidade de pessoal	■	■	■			
Uso uniforme do tempo de projeto				■		■
Expertise insuficiente em métodos, software e equipamentos						■
Dificuldade em gerenciar complexidade do projeto	■				■	
Funções do software desenvolvido em descompasso com necessidades dos usuários		■				
Software inclui funcionalidade apenas marginalmente necessária ("gold plating")				■		
Mudança constante de requisitos				■		
Cronograma mudando continuamente	■			■		
Usuários descontentes com interface do usuário implementada		■				
Componentes adquiridos externamente não atendem suas especificações			■			
Expectativas irrealistas das habilidades dos membros da equipe do projeto						■
Requisitos de desempenho (tempo de resposta, etc) estimados incorretamente					■	■
Tarefas subcontratadas realizadas de forma insatisfatória			■			
Estimativa inadequada da infraestrutura de hardware e software		■			■	

■ Variáveis que contribuem para o componente de risco

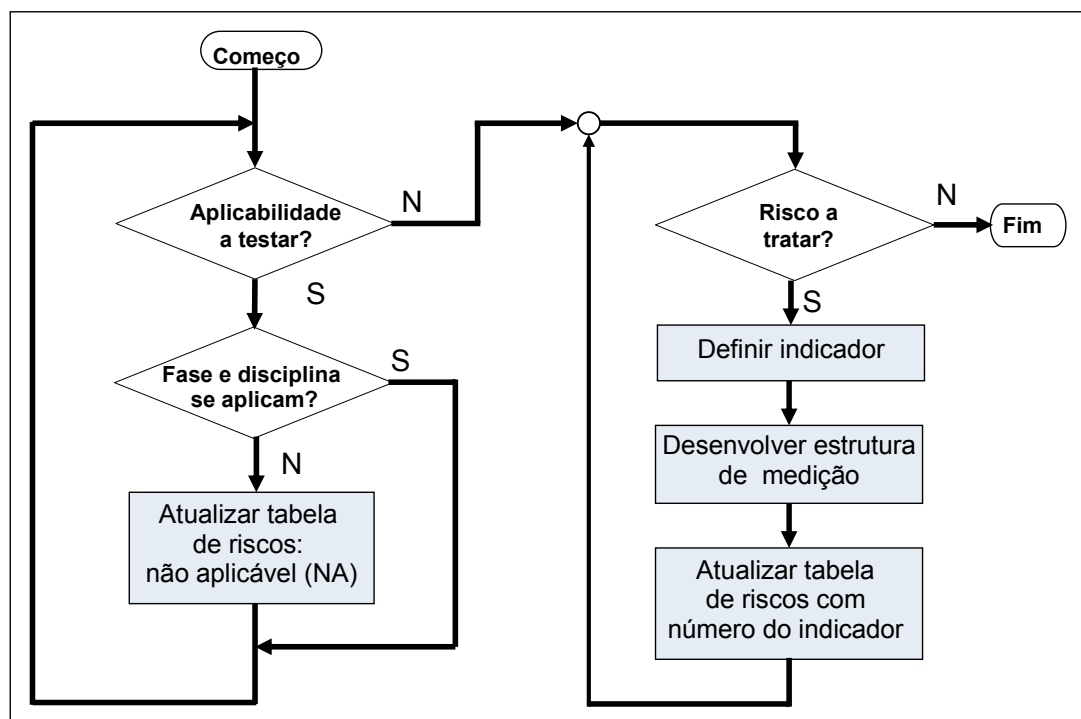
Adaptado de (Ropponen, 2000)

4.3 Processo de criação de indicadores

A análise das variáveis de risco para determinação dos indicadores é feita em duas etapas conforme ilustrado no fluxo de processo da Figura 4-2.

- Inicialmente, como demonstra o laço à esquerda da figura, cada uma das variáveis de risco da matriz da Figura 4-1 foi verificada quanto à sua aplicabilidade em relação às limitações estabelecidas para o presente trabalho. O processo executado é descrito em detalhe na seção 4.4.
- Numa segunda etapa, como ilustra o laço da direita da Figura 4-2, cada uma das variáveis de risco aplicáveis foi examinada para efeito de definição de indicadores. O processo executado é descrito em detalhe na subseção 4.5.

Figura 4-2 – Fluxo do processo de criação de indicadores



As seções a seguir desenvolvem esse processo.

4.4 Aplicação das limitações do trabalho

Conforme se estabelece na seção Problema da Pesquisa, o presente trabalho limita as variáveis de riscos para os quais são definidos indicadores. São consideradas as variáveis que se relacionam com as atividades da fase de construção do Processo Unificado e com as principais disciplinas dessa fase: implementação, teste, configuração, gerenciamento de mudanças e gerenciamento de projeto. Limitação adicional é que os riscos possam ser

tratados pelo gerente do projeto ou por sua equipe, não se considerando riscos cujo tratamento requeira principalmente o envolvimento de pessoas ou entidades externas à equipe de projeto.

A aplicação dessas limitações à combinação de variável de risco e componentes de risco da matriz da Figura 4-1 reduz os elementos da matriz que devem ser avaliados para a definição de indicadores conforme ilustra a Figura 4-3:

Figura 4-3 – Matriz de variáveis de risco a avaliar

Variáveis de risco	Componentes de risco					
	Cronograma e duração	Funcionalidades do sistema	Subcontratação	Gerenciamento de requisitos	Uso de recursos e desempenho	Gerenciamento de pessoal
Deficiências de pessoal						NA
Problemas de cronograma						
Utilização de recursos incompatível com data de conclusão						
Custos estimados e custos reais em descompasso						
Estimativa incorreta de tamanho						
Estimativa de necessidade de pessoal			NA			
Uso uniforme do tempo de projeto						NA
Expertise insuficiente em métodos, software e equipamentos						NA
Dificuldade em gerenciar complexidade do projeto						
Funções do software desenvolvido em descompasso com necessidades dos usuários		NA				
Software inclui funcionalidade apenas marginalmente necessária ("gold plating")				NA		
Mudança constante de requisitos						
Cronograma mudando continuamente						
Usuários descontentes com interface do usuário implementada		NA				
Componentes adquiridos externamente não atendem suas especificações			NA			
Expectativas irrealistas das habilidades dos membros da equipe do projeto						NA
Requisitos de desempenho (tempo de resposta, etc) estimados incorretamente					NA	NA
Tarefas subcontratadas realizadas de forma insatisfatória			NA			
Estimativa inadequada da infraestrutura de hardware e software		NA			NA	

	Variáveis que contribuem para o componente de risco
NA	Não aplicável

Adaptado de (Ropponen, 2000)

Os elementos indicados com NA (não aplicável) foram excluídos em função de não atenderem às limitações impostas quando da formulação do problema de pesquisa, reproduzidas acima nesta seção.

Desta forma, foram excluídos elementos da matriz, associados:

- Ao componente de risco de gerenciamento de pessoal, por envolverem em grande escala questões de recrutamento e formação de pessoal tratadas no âmbito da empresa ou organização à qual se vincula o projeto;
- Ao componente de risco de subcontratação, por se tratar de entidade externa ao projeto;
- À disciplina de requisitos com maior relevância nas fases de iniciação e elaboração; variáveis de risco relativas ao descompasso com as necessidades dos usuários; inclusão de funcionalidade apenas marginalmente necessária; descontentamento dos usuários com a interface implementada; e requisitos de desempenho estimados incorretamente;
- A riscos associados a entidade externa à equipe de projeto: estimativa inadequada de infra-estrutura de hardware e software.

4.5 Definição dos Indicadores

Nas subseções a seguir, segundo o processo descrito no lado direito Figura 4-2, determinam-se e descrevem-se indicadores para atender aos objetivos de identificar potenciais problemas para os projetos. A Figura 4-3 – Matriz de variáveis de risco a avaliar é examinada e cada variável de risco que não foi eliminada em função das limitações definidas para este trabalho é associada a um ou mais indicadores. A seguir, se desenvolve a estrutura de medição do indicador. A cada iteração, se atualiza a Matriz de variável de risco a avaliar, com o número do indicador.

Apresentam-se, inicialmente, os indicadores cuja definição ou resultados intermediários utilizados são requeridos na definição de outros indicadores, não seguindo portanto rigidamente a ordem da Figura 4-3.

Ao final do Capítulo, se reproduz a matriz de variáveis de risco, indicando quais as variáveis que foram excluídas, em função da limitação definida para o trabalho, e as que tiveram indicador associado efetivamente a elas.

4.5.1 Mudança constante de requisitos

Para tratamento do risco de mudança constante de requisitos, e a verificação de se o planejamento do projeto está sendo atualizado de forma consistente, propõe-se um indicador que examina a evolução do tamanho funcional do sistema ou produto. O tamanho funcional de um dado produto é obtido através da contagem de pontos de função na forma descrita no capítulo 3.

A evolução do tamanho funcional pode ocorrer por inclusões, exclusões e mudanças de funcionalidade. Cabe ao projeto acompanhar a evolução do tamanho funcional através de recontagem de pontos de função em determinados pontos do projeto, ou *milestones*. A recontagem se deve fazer mesmo quando o usuário não demanda alteração explícita de requisitos.

O indicador proposto responde à questão sobre se os requisitos se alteraram significativamente ao longo do projeto. O indicador consiste de gráfico que anota, a cada ponto de revisão do projeto, o valor da funcionalidade instalada e da funcionalidade modificada do sistema (definidas a seguir), comparando-as com a funcionalidade inicial.

A Tabela 4-1 representa esquematicamente a associação entre variável de risco e indicador que detecta a manifestação do risco.

Tabela 4-1 – Risco de mudança constante de requisitos

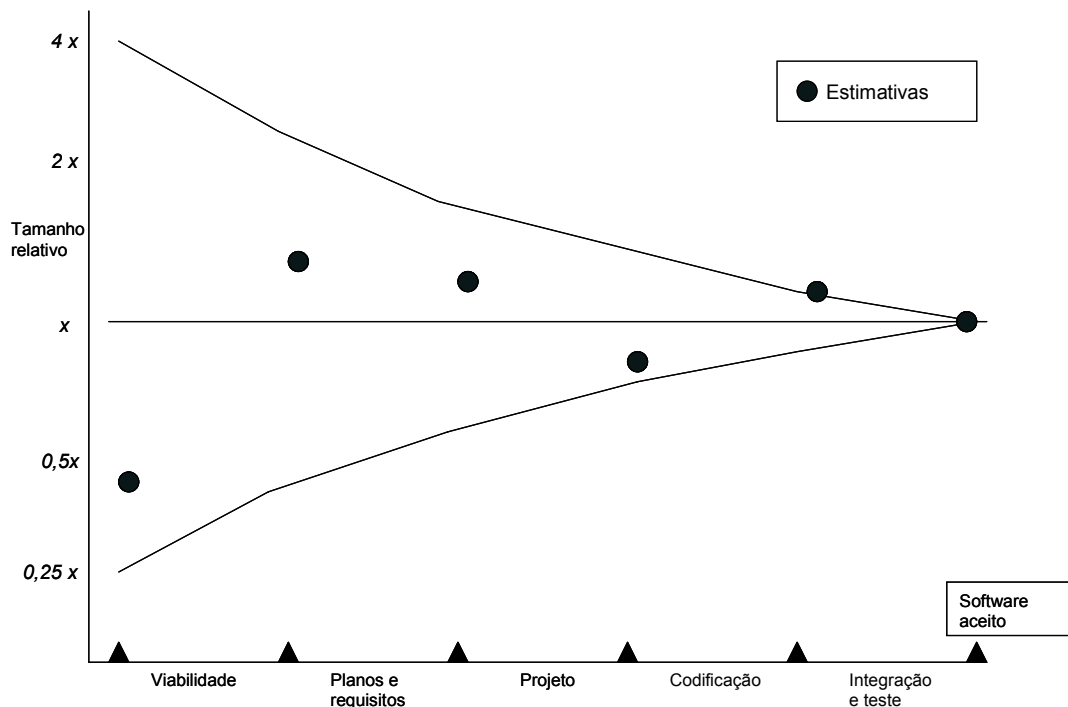
Variável de risco		Componente de risco afetado
Mudança constante de requisitos		Gerenciamento de requisitos
Questões	Os requisitos foram substancialmente alterados ao longo do projeto?	
Indicador	Gráfico comparando a funcionalidade instalada e a modificada do sistema no momento com a funcionalidade inicial, todas medidas em pontos de função.	I-1

Discussão

O tamanho do sistema se altera, ao longo do tempo, segundo padrão expresso pelo *funnel graph* descrito por (Boehm, 2000). Esse gráfico, ilustrado na Figura 4-4, mostra como a indefinição do tamanho do sistema se vai reduzindo ao longo do tempo. (McConnell, 2006), que denomina essa figura de “cone da incerteza”, descreve como um projeto de uma

organização com processos de medição maduros consegue fazer com que suas estimativas rapidamente aumentem de acurácia e converjam para o tamanho do produto entregue.

Figura 4-4 – O cone da incerteza



Adaptado de (Boehm, 2000 e McConnell, 2006)

Para as finalidades deste trabalho interessa identificar dois tipos de tamanho funcional:

- Tamanho funcional instalado: Valor da contagem, a cada momento, da funcionalidade ativa de um sistema. Por funcionalidade ativa entende-se a funcionalidade inicial ou a funcionalidade resultante de mudanças que se implementaram no projeto, ou ainda de recontagem regular feita em determinados *milestones*. Na conclusão do projeto, o tamanho do produto é o tamanho funcional instalado final;
- Tamanho funcional modificado: Valor funcional inicial do sistema em um dado momento, acrescido de adição, deleção ou mudança de requisitos. Essas mudanças, mesmo no caso de simples deleção, acarretam perdas por retrabalho, devendo, portanto, ser refletidas no tamanho funcional modificado.

O tamanho funcional instalado, ou abreviadamente funcionalidade instalada, é o que melhor reflete o tamanho do produto construído.

Por seu lado, o tamanho funcional modificado, ou funcionalidade modificada, serve para efeito de cálculo de esforço e de custo (o tratamento das questões de custo está fora do escopo deste trabalho, mas sem dúvida seu impacto em um projeto é relevante).

Na determinação do tamanho funcional instalado e do modificado são utilizados os critérios do IFPUG para melhorias, expressos nas fórmulas [4-1] e [4-2] (Garmus, 2001):

$$\text{Modificado} = \text{Base} + \text{Adição} + \text{Alteração} + \text{Eliminação} \quad [4-1]$$

$$\text{Instalado} = \text{Base} + \text{Adição} - \text{Alteração}_{\text{antes}} + \text{Alteração}_{\text{depois}} - \text{Eliminação} \quad [4-2]$$

Base aqui é o valor do tamanho funcional anterior às adições, alterações ou eliminação de funcionalidade.

A funcionalidade base ou inicial é a contagem original. A partir desse ponto, cada mudança dos requisitos determinará recontagem em que serão determinadas variações de ponto de função resultantes de adição, alteração e eliminação de funcionalidade. As três mudanças de funcionalidade – adição, alteração e eliminação – contribuem para o aumento da funcionalidade modificada, segundo as regras do IFPUG e fórmulas [4-1] e [4-2].

O IFPUG adota o critério de somar funcionalidade alterada ou eliminada ao tamanho base, para gerar tamanho modificado. Isso se deve a que, depois que um artefato é desenvolvido, sua modificação, ou mesmo eliminação, requer esforço para adaptação do restante do código, bem como para o teste do produto integrado. Essa orientação deve ser claramente exposta a todos os participantes, diretos ou indiretos do projeto, uma vez que, embora a funcionalidade eliminada não contribua para o tamanho final, afeta o esforço e prazo finais do projeto, bem como seu custo.

No caso de recontagem por ocasião do atingimento de *milestones*, apenas a funcionalidade resultante da maior clareza nos requisitos será adicionada (ou reduzida) ao tamanho funcional instalado e ao modificado.

Indicador

O indicador que se descreve na Tabela 4-2, através de estrutura de medição, utiliza a variação entre o tamanho instalado, o tamanho modificado e o tamanho inicial do sistema, em um dado momento, para aferir o grau de mudança de requisitos ao longo do projeto.

Tabela 4-2 – Indicador de variação de funcionalidade

I-1 - Variação de funcionalidade	
Necessidade de informação	Os requisitos foram substancialmente alterados ao longo do projeto?
Conceito mensurável	Tamanho funcional
Entidades relevantes	1.Requisitos base (inicial ou recontagem dos requisitos iniciais) 2.Requisitos adicionados 3.Requisitos alterados 4.Requisitos eliminados
Atributos	Funcionalidade associada aos itens 1 a 4 anteriores
Medidas primitivas	1.Tamanho funcional dos requisitos base 2.Tamanho funcional dos requisitos adicionados 3.Tamanho funcional dos requisitos alterados 4.Tamanho funcional dos requisitos eliminados
Método de medição	Regras de contagem do IFPUG 4.2
Tipo de método	Objetivo
Escala	Números inteiros
Tipo de escala	Razão
Unidade de medida	Pontos de função
Medidas derivadas	1.Funcionalidade instalada 2.Funcionalidade modificada
Função de medição	1.Instalada = base + adicionados - alterados _{antes} + alterados _{depois} - eliminados 2.Modificada = base + adicionados + alterados + eliminados
Indicador	Gráfico comparando a funcionalidade instalada e a modificada do sistema no momento da revisão com a funcionalidade inicial, todas medidas em pontos de função.
Modelo de análise	Variação superior à estabelecida nos critérios de decisão requer exame de impacto no esforço e prazo e renegociação de preço e prazo.
Crítérios de decisão	Diferença entre funcionalidade modificada e funcionalidade inicial superior a 3% da funcionalidade inicial.
Aplicabilidade	Fase de elaboração e construção

Usos do indicador no gerenciamento de riscos

O gerente do projeto atualizará o indicador a cada revisão do projeto. A frequência de revisão poderá variar, tipicamente, de uma vez por semana até a uma vez por mês, dependendo do padrão da organização no qual o projeto é desenvolvido.

O indicador permitirá ao gerente do projeto, quando houver a manifestação do risco, averiguar as questões abaixo:

- A duração e o esforço do projeto e a tamanho da equipe técnica foram revistos?
- O custo (e se aplicável o preço) e a duração dos projetos foram renegociados com clientes e usuários?
- Os artefatos a serem gerados foram revistos em função das modificações introduzidas?
- A variação entre a funcionalidade modificada e a funcionalidade inicial está tendendo a um valor constante, permitindo inferir que os requisitos tendem a se estabilizar?

4.5.2 Estimativa incorreta de tamanho

Para tratamento do risco de estimativa incorreta de tamanho do sistema e a verificação se o planejamento do projeto está sendo afetado por essa incorreção, propõe-se um indicador que reflita a estimativa de tamanho em linhas de código. Esse indicador evolui de forma sincronizada com a evolução do tamanho funcional instalado do sistema. O tamanho funcional instalado do sistema, termo esse definido na subseção anterior, indica a funcionalidade ativa de um sistema, sem levar em conta como se chegou a essa funcionalidade. Em outras palavras, pouco importa se a funcionalidade instalada resultou unicamente de um novo desenvolvimento ou se várias inclusões, alterações e exclusões aconteceram.

O tamanho estimado do produto é um indicador obtido como medida derivada através de contagem de pontos de função não ajustados e sua transformação em linhas de código lógicas desprezando-se comentários, na forma descrita no capítulo 3. Para efeito de conversão para linhas de código, utiliza-se a contagem de pontos de função instalada, uma vez que é a funcionalidade ativa que deve gerar código para inserção nas bibliotecas do projeto, descontadas as eliminações e alterações em função de mudanças de requisitos.

O indicador proposto responde à questão sobre se o tamanho do sistema, em linhas de código, reflete as mudanças, pelas quais passaram os requisitos desde a estimativa inicial do projeto.

O indicador consiste de gráfico que anota, a cada ponto de revisão do projeto, o tamanho do sistema em linhas de código e o compara com a estimativa base ou inicial, ou seja, a variação de tamanho do produto.

A Tabela 4-3 representa esquematicamente a associação entre variável de risco e indicador que detecta a manifestação do risco.

Tabela 4-3 – Risco de estimativa incorreta de tamanho

Variável de risco		Componente de risco afetado	
Estimativa incorreta de tamanho		Cronograma e duração	
Questões	As alterações de funcionalidade requeridas foram incorporadas à estimativa de tamanho do produto?		
Indicador	Gráfico mostrando a variação de tamanho do produto, através da diferença entre a estimativa inicial e a estimativa atual do tamanho do produto em linhas de código.		I-2

Discussão

Do ponto de vista de controle do projeto, o tamanho funcional instalado é usado em comparações com os artefatos produzidos para acompanhamento de progresso.

Na determinação do tamanho funcional instalado é utilizada a fórmula [4-2], reproduzida da seção anterior, segundo os critérios do IFPUG (Garmus, 2001):

$$\text{Instalado} = \text{Base} + \text{Adição} - \text{Alteração}_{\text{antes}} + \text{Alteração}_{\text{depois}} - \text{Eliminação} \quad [4-3]$$

Base aqui é o valor do tamanho funcional anterior às adições, alterações ou eliminações de funcionalidade.

O valor instalado inicial é a contagem original. A partir desse ponto, cada mudança dos requisitos determinará recontagem em que serão determinadas variações de ponto de função resultantes de adição, alteração e eliminação de funcionalidade. Apenas o valor da adição e deleção de funcionalidade influencia substancialmente o tamanho instalado do produto, segundo a fórmula [4-2], assumindo-se, por simplificação das operações de contagem, que os valores das funções alteradas, antes e depois da alteração, sejam aproximadamente iguais.

No caso de recontagem por ocasião do atingimento de *milestones*, apenas a funcionalidade resultante da maior clareza nos requisitos é adicionada ao valor funcional instalado ou dele deduzida.

O tamanho funcional instalado serve de base para determinar a estimativa do tamanho do sistema ou produto, na forma descrita na subseção 3.2.4, ou seja, através da multiplicação do tamanho instalado pelo *gearing factor* da linguagem de programação adotada. O tamanho estimado do produto é importante por fornecer ao gerente do projeto uma medida em relação à qual se pode acompanhar o progresso das atividades de construção.

Indicador

O indicador que se descreve através de estrutura de medição da Tabela 4-4, reflete a variação entre a estimativa inicial do tamanho do produto e a estimativa atual. Ambas as estimativas de tamanho são derivadas da funcionalidade instalada dos requisitos, multiplicada pelo *gearing factor*.

O tamanho estimado do produto em linhas de código deve ser recalculado a cada mudança significativa nos requisitos e também quando o projeto atinge seus *milestones* mais importantes e passa por recontagem de pontos de função.

Usos do indicador no gerenciamento de riscos

O indicador permitirá ao gerente do projeto averiguar a manifestação do risco, associado às questões abaixo, quando os critérios de decisão descritos na Tabela 4-4 se verificarem:

- O aumento do tamanho do produto a construir está considerado no planejamento do projeto?
- A equipe do projeto foi revista para permitir a construção do código adicional necessário?
- A duração do projeto foi revista em função do aumento do tamanho do produto a construir?

Tabela 4-4 – Indicador de variação de tamanho de produto

I-2 - Variação de tamanho do produto	
Necessidade de informação	As alterações de funcionalidade requeridas foram incorporadas à estimativa de tamanho do produto?
Conceito mensurável	Tamanho funcional do produto
Entidades relevantes	Requisitos base (inicial ou recontagem dos requisitos iniciais)
Atributos	Tamanho em linhas de código
Medidas primitivas	1.Funcionalidade inicial dos requisitos 2.Funcionalidade instalada dos requisitos 3. <i>Gearing Factor</i>
Método de medição	1.Regras de contagem do IFPUG 4.2 2.Regras de contagem do IFPUG 4.2 3.Média da indústria
Tipo de método	1.Objetivo 2.Objetivo 3.Objetivo
Escala	1.Números inteiros 2.Números inteiros 3.Números reais
Tipo de escala	1.Razão 2.Razão 3.Razão
Unidade de medida	1.Pontos de função 2.Pontos de função 3.Linhas de código por ponto de função
Medidas derivadas	1.Estimativa inicial de linhas de código 2.Estimativa atual de linhas de código
Função de medição	1.Estimativa inicial = Funcionalidade inicial * <i>gearing factor</i> 2.Estimativa atual = Funcionalidade instalada * <i>gearing factor</i>
Indicador	Gráfico mostrando a variação de tamanho do produto através da diferença entre a estimativa inicial e a estimativa atual do tamanho do produto em linhas de código.
Modelo de análise	Variações superiores às especificadas nos critérios de decisão requerem análise de causa e re-estimativa de esforço e prazo.
Crítérios de decisão	Diferença entre estimativa inicial e estimativa atual superior a 10% do valor inicial.
Aplicabilidade	Fase de Construção

4.5.3 Problemas de cronograma

Para tratamento do risco de problemas de cronograma que levem a atraso na entrega do produto, propõe-se indicador que compara a duração estimada inicial ou renegociada do projeto com a duração prevista no momento da revisão. Para a determinação da duração prevista do projeto se leva em conta, o tempo decorrido desde o início da fase de construção do projeto, somado ao tempo estimado para a conclusão da codificação restante.

O indicador proposto responde à questão sobre se há variação entre a duração estimada inicial do projeto e a duração estimada atual. O indicador consiste de gráfico que anota, a cada ponto de revisão do projeto, a nova estimativa de duração e a compara com a estimativa inicial ou renegociada.

A Tabela 4-5 representa esquematicamente a associação entre variável de risco e indicador que detecta a manifestação do risco.

Tabela 4-5 – Risco de problemas de cronograma.

Variável de risco		Componente de risco afetado
Problemas de cronograma		Cronograma e duração
Questões	Há variação entre a duração estimada inicial do projeto e a duração estimada atual?	
Indicador	Gráfico mostrando potencial atraso do projeto, através da comparação entre a duração estimada inicial ou renegociada e a duração estimada atual a cada revisão do projeto com base no progresso.	I-3

Discussão

A duração estimada dos projetos é obtida, neste trabalho, através da aplicação do modelo de Putnam, descrito no capítulo 3. Entretanto, é possível que essa duração estimada seja substituída, para efeito de planejamento do projeto, em favor de uma duração fixada em função de necessidades dos usuários através de decisão gerencial. A duração do projeto é importante, em primeira análise, por permitir determinar a data de entrega do produto para que seja implementado em produção para os usuários na fase de transição. Interessa ao gerente do projeto e à gerência da organização determinar se as mudanças de escopo e de

requisitos estão sendo consideradas para efeito de re-estimativa da duração e se seu impacto na duração do projeto está sendo calculado. Também é importante determinar se as atividades do projeto estão produzindo artefatos no ritmo necessário para que o projeto conclua na data prevista.

Na determinação da duração estimada dos projetos é importante tomar uma série de cuidados:

- entender claramente quais fases do ciclo de vida do projeto correspondem às estimativas elaboradas;
- usar a precisão adequada, empregando duas casas decimais, para durações expressas em anos, e uma casa decimal quando expressas em meses;
- quando se for comparar com a duração real descontar eventuais tempos de paralisação do projeto.

Esses pontos, frequentemente ignorados, dão origem à boa parte da imprecisão dos indicadores calculados (Putnam, 1992).

Ao projeto importa comparar sua duração estimada inicialmente, ou mais propriamente no caso específico deste trabalho, a duração inicial estimada da fase de construção, com a previsão atualizada de conclusão. A previsão atualizada de conclusão leva em conta o tempo decorrido desde o início da construção e a estimativa de tempo para a conclusão das atividades restantes do projeto.

Caso a previsão atualizada de conclusão da fase de construção seja superior à duração estimada inicialmente ou a duração renegociada, o projeto deve ser objeto de análise causal para a aplicação de ações corretivas.

Indicador

O indicador que se descreve através de estrutura de medição da Tabela 4-6 aponta possível atraso no projeto, expresso pela variação entre a estimativa inicial ou renegociada da duração do projeto e a estimativa de duração atual. A estimativa de duração inicial é obtida pelas equações descritas na subseção 3.3.2. A estimativa de duração atual é derivada do tempo decorrido do início da fase de construção, mais cálculo de tempo para desenvolvimento do restante do código, na forma descrita pela estrutura de medição da Tabela 4-6.

Tabela 4-6 – Indicador de atraso de projeto

I-3 - Indicador de atraso do projeto	
Necessidade de informação	Há variação entre a duração estimada inicial do projeto, ou duração renegociada, e a duração estimada atual?
Conceito mensurável	Cumprimento de prazo
Entidades relevantes	Plano do projeto
Atributos	Duração do projeto
Medidas primitivas	1.Funcionalidade instalada dos requisitos 2.Parâmetros PP e MBP (Modelo paramétrico de Putnam) 3.Tempo decorrido do início do projeto (decorrido) 4.Linhas de código dos artefatos desenvolvidos e testados (pronto)
Método de medição	1.Regras de contagem do IFPUG 4.2. 2.Estimativa dos parâmetros da instalação 3.Medição de tempo calendário 4.Contagem de linhas de código
Tipo de método	1.Objetivo 2.Objetivo 3.Objetivo 4.Objetivo
Escala	1.Números inteiros 2.Números reais 3.Numero reais 4.Número inteiros
Tipo de escala	1.Razão 2.Razão 3.Razão 4.Razão
Unidade de medida	1.Pontos de função 2.Dimensão de PP e MBP 3.Anos 4.Linhas de código
Medidas derivadas	1.Tamanho do projeto em linhas de código (tamanho) 2.Duração estimada (t_d) 3.Tempo restante até a conclusão 4.Código que falta desenvolver (faltante) 5.Duração estimada atual (atual)
Função de medição	1. tamanho = funcionalidade * <i>gearing factor</i> 2. $t_d = (0,39 \text{ MBP})^{-1/7} \text{ PP}^{-3/7} (\text{tamanho} * \beta^{1/3})^{3/7}$ 3. tempo restante = t_d - decorrido 4. cód. faltante = tamanho - pronto 5. atual = decorrido+tempo restante*cód. faltante/(tamanho - L(t)) [L(t) definido no I-8]
Indicador	Gráfico mostrando potencial atraso do projeto através da comparação entre a duração estimada inicial, ou renegociada, e a duração estimada atual a cada revisão do projeto.
Modelo de análise	Analisar causa e determinar ação corretiva, inclusive revendo a data de implantação e a funcionalidade a implantar
Crítérios de decisão	Diferença entre estimativa inicial de duração e a estimativa atual superior ao desvio padrão.
Aplicabilidade	Fase de construção

Usos do indicador no gerenciamento de riscos

O indicador permitirá ao gerente do projeto averiguar a manifestação do risco, associado às questões abaixo, quando os critérios de decisão descritos na Tabela 4-6 se verificarem:

- Houve aumento de funcionalidade que não ocasionou revisão de prazo de entrega?
- O progresso do desenvolvimento é inferior ao previsto e necessário para a conclusão do projeto no prazo previsto?
- A equipe de desenvolvimento foi alocada segundo o previsto no plano de projeto?
- A equipe de desenvolvimento é qualificada para a tarefa?
- Está havendo retrabalho significativo em artefatos que já haviam sido testados e haviam passado pelos critérios de aceitação para serem colocados sob controle de configuração?
- Está havendo redução da equipe de desenvolvimento alocada para o projeto?

4.5.4 Alterações constantes de cronograma

Para tratamento do risco de contínuas alterações de cronograma que indiquem instabilidade do projeto e ameaça de atraso na entrega do produto, propõe-se indicador que compare a data compromissada de conclusão do projeto com a data prevista no momento de cada revisão periódica, com base na alteração dos requisitos. A data compromissada é a data prevista inicialmente, a menos que tenha havido renegociação de outra data. A comparação das sucessivas estimativas de conclusão a cada revisão indica se o risco está se manifestando. As datas de conclusão do projeto serão determinadas a partir da funcionalidade inicial ou renegociação e da funcionalidade modificada.

O indicador proposto pode verificar se a data estimada de conclusão está sendo alterada com frequência, o que pode ser indicativo de problemas no projeto. O indicador consiste de gráfico que anota, a cada ponto de revisão do projeto, a nova estimativa de conclusão e a compara com a data compromissada.

A Tabela 4-7 representa esquematicamente a associação entre variável de risco e indicador que detecta a manifestação do risco.

Tabela 4-7 – Risco de contínuas alterações de cronograma

Variável de risco		Componente de risco afetado
Cronograma mudando continuamente		Cronograma e duração Gerenciamento de requisitos
Questões	Há repetidas alterações da data de conclusão do projeto?	
Indicador	Gráfico mostrando evolução das re-estimativas de conclusão do projeto, através da comparação entre a data estimada inicial ou renegociada e a data estimada atual a cada revisão do projeto com base na variação dos requisitos.	I-4

Discussão

Ao projeto importa comparar sua data compromissada para o fim da fase de construção, com a previsão atualizada de conclusão.

Caso as datas estimadas de conclusão se estejam alterando com frequência e a diferença entre a previsão inicial e as estimativas revisadas seja crescente ao longo das repetidas revisões periódicas, o projeto deve ser objeto de análise causal para a aplicação de ações corretivas.

Indicador

O indicador da Tabela 4-8, que se descreve através de estrutura de medição, detecta alterações repetidas da data de conclusão do projeto. O critério escolhido para isso é estarem as três últimas re-estimativas de conclusão, ao mesmo tempo, após a data compromissada e exibirem valores crescentes. Isso pode indicar instabilidade da data de conclusão do projeto. A data compromissada é a estimativa da data de conclusão inicial obtida através do cálculo de duração inicial pelas equações descritas na subseção 3.3.2, ou data renegociada. A estimativa das datas revisadas de conclusão é obtida de forma semelhante a partir dos novos requisitos do projeto, na forma descrita pela estrutura de medição da Tabela 4-8.

Tabela 4-8 – Indicador de evolução de re-estimativas

I-4 - Evolução das re-estimativas	
Necessidade de informação	Há repetidas alterações da data de conclusão do projeto?
Conceito mensurável	Tempo de desenvolvimento
Entidades relevantes	A fase de construção
Atributos	Data de conclusão
Medidas primitivas	1.Funcionalidade inicial dos requisitos 2.Parâmetros PP e MBP (Modelo paramétrico de Putnam) 3.Funcionalidade modificada 4. <i>Gearing Factor</i>
Método de medição	1.Regras de contagem do IFPUG 4.2. 2.Estimativa dos parâmetros da instalação 3.Regras de contagem do IFPUG 4.2. 4.Média da indústria
Tipo de método	1.Objetivo 2.Objetivo 3.Objetivo 4.Objetivo
Escala	1.Números inteiros 2.Números reais 3.Número inteiros 4.Números reais
Tipo de escala	1.Razão 2.Razão 3.Razão 4.Razão
Unidade de medida	1.Pontos de função 2.Dimensão de PP e MBP 3.Pontos de função 4.Linhas de código por ponto de função
Medidas derivadas	1.Tamanho inicial (tamanho) 2.Prazo estimado inicial (inicial) 3.Tamanho modificado 3.Prazo estimado atual (atual)
Função de medição	1. tamanho = funcionalidade inicial * <i>Gearing Factor</i> 2. inicial = início + (0,39 MBP) ^{-1/7} PP ^{-3/7} (tamanho * $\beta^{1/3}$) ^{3/7} ou negociada. 3. modificado = funcionalidade modificada * <i>Gearing Factor</i> 4. atual = início + (0,39 MBP) ^{-1/7} PP ^{-3/7} (modificado * $\beta^{1/3}$) ^{3/7}
Indicador	Gráfico mostrando evolução das re-estimativas de conclusão do projeto, através da comparação entre a data estimada inicial ou renegociada e a data estimada atual a cada revisão do projeto com base na variação dos requisitos.
Modelo de análise	Analisar causa e determinar ação corretiva inclusive revendo a data de implantação e a funcionalidade a implantar
CrITÉrios de decisão	As três últimas re-estimativas de conclusão estarem, ao mesmo tempo, após a estimativa inicial ou renegociada e exibirem valores crescentes
Aplicabilidade	Fase de construção

Usos do indicador no gerenciamento de riscos

O indicador permitirá ao gerente do projeto averiguar a manifestação do risco, associado às questões abaixo, quando os critérios de decisão descritos na Tabela 4-8 se verificarem:

- Está havendo freqüentes alterações de funcionalidade que não ocasionaram revisão de prazo de entrega?
- O retrabalho oriundo de funcionalidade alterada e funcionalidade excluída foi levado em consideração no esforço e custo do projeto?
- O retrabalho oriundo de funcionalidade alterada e funcionalidade excluída foi utilizado para promover revisão da data de entrega do projeto?

4.5.5 Custos estimados e reais em descompasso

Para o tratamento do risco de que o custo real e o custo estimado estejam em descompasso, é preciso reconhecer que, em um projeto de desenvolvimento, a mão-de-obra é o principal componente de custo. Para acompanhar o risco em questão, cabe comparar a utilização real de horas do projeto com a projeção feita durante o planejamento. Essa comparação é feita por ocasião das revisões periódicas do projeto. As sucessivas comparações permitem verificar se o risco está se manifestando.

O indicador proposto responde à questão sobre se há variação significativa, segundo critério de decisão estabelecido, entre a estimativa inicial de esforço e o esforço efetivamente empregado. O indicador consiste de gráfico que anota, a cada ponto de revisão do projeto, a utilização acumulada da mão-de-obra desde o início da fase de construção e a compara com a curva projetada ao início do projeto. Caso se faça replanejamento do projeto, a nova curva de utilização do esforço é utilizada na comparação.

A Tabela 4-9 representa esquematicamente a associação entre variável de risco e indicador que detecta a manifestação do risco.

Tabela 4-9 – Risco de custos estimados e reais em descompasso

Variável de risco		Componente de risco afetado
Custos estimados e custos reais em descompasso		Cronograma e duração
Questões	O esforço empreendido em construção até o momento é compatível com o esforço esperado?	
Indicador	Gráfico comparando o esforço real com o esforço acumulado estimado até o momento.	I-5

Discussão

Importa ao gerente do projeto conhecer, tão logo seja possível, a estimativa de esforço do projeto, que é base para diversas outras estimativas de medidas derivadas entre as quais o cálculo de custo do projeto, o cálculo de alocação de mão-de-obra por fase e a estimativa de esforço acumulado a cada instante.

O esforço estimado do projeto é obtido a partir da medição do tamanho funcional modificado do produto, e sua subsequente transformação em linha de código, como passo intermediário. Utiliza-se a funcionalidade modificada, ao invés da funcionalidade instalada, uma vez que a primeira inclui funcionalidade que foi excluída e modificada, permitindo a inclusão do esforço efetivamente despendido em retrabalho.

Utilizam-se as equações paramétricas de Putnam para chegar-se à estimativa de esforço, utilizando-se os indicadores de produtividade e de *manpower buildup* descritos no Capítulo 3. A estimativa de esforço será refeita a cada variação do tamanho do projeto.

O esforço real é medido através de apontamento de horas efetivamente dedicadas ao projeto. Deve-se tomar o cuidado de não errar por falta ou excesso: por falta, deixando de reportar horas trabalhadas, mesmo em regime de horas extras ou horas não pagas; por excesso, cuidando para não reportar no projeto horas a ele não relacionadas, como reuniões para tratar de assuntos alheios ao projeto, atividades ligadas a outros projetos, compromissos particulares.

O esforço é medido em homens-mês, ou em homens-ano, sendo esta última a forma utilizada nas equações deste trabalho.

Importa ao gerente do projeto comparar o valor acumulado estimado do esforço, em pontos específicos do tempo, com o esforço acumulado efetivamente despendido em homens-hora e reportado pelo pessoal do projeto. A variação entre o esforço acumulado real e a estimativa será considerada significativa quando ultrapassar um desvio padrão da estimativa. Nesse caso, a variação deve ser objeto de análise causal para efeito de determinação de ações corretivas.

Indicador

O indicador da Tabela 4-10, que se descreve através de estrutura de medição, aponta variação entre o esforço real acumulado desde o início da fase de construção e a estimativa inicial do projeto, ou re-estimativa feita em replanejamento do projeto. A estimativa de esforço é obtida na subseção 3.3.1.2. Já a distribuição acumulada do esforço é calculada através das equações descritas em 3.3.2, especialmente pela fórmula [3-7]. O desvio padrão utilizado para determinar se há variação significativa que deva ser analisada se obtém através das equações [3-11] a [3-13]. Os valores reais do esforço despendido são obtidos de relatórios de reporte de horas utilizadas no projeto.

Usos do indicador no gerenciamento de riscos

O indicador permitirá ao gerente do projeto averiguar a manifestação do risco, associado às questões abaixo, quando os critérios de decisão descritos na Tabela 4-10 se verificarem:

- Há atrasos de cronograma, que estão impedindo alocação de mão-de-obra ao projeto?
- Há desvio de recursos para atividades alheias ao projeto?
- A mão-de-obra requerida pelo projeto foi alocada na ocasião adequada?
- As mudanças de escopo e de requisitos estão sendo consideradas para efeito de re-estimativa e seu impacto no esforço do projeto está sendo calculado?

Tabela 4-10 – Comparação entre esforço estimado e despendido

I-5 - Comparação esforço estimado e esforço despendido	
Necessidade de informação	O esforço empreendido em construção até o momento é compatível com o esforço esperado?
Conceito mensurável	Esforço do pessoal do projeto
Entidades relevantes	A fase de construção
Atributos	Esforço despendido
Medidas primitivas	1.Funcionalidade modificada 2. <i>Gearing Factor</i> 3.Parâmetros PP e MBP 4.Tempo decorrido do projeto 5.Esforço despendido até o momento
Método de medição	1.Regras de contagem do IFPUG 4.2 2.Média da indústria 3.Histórico dos projetos anteriores(esforço, duração, tamanho) 4.Medição de tempo calendário 5.Reporte de horas
Tipo de método	1.Objetivo 2.Objetivo 3.Objetivo 4.Objetivo 5.Objetivo
Escala	1.Números inteiros 2.Números reais 3.Números reais 4.Números reais 5.Números reais
Tipo de escala	1.Razão 2.Razão 3.Razão 4.Razão 5.Razão
Unidade de medida	1.Pontos de função 2.Linhas de código por ponto de função 3.Dimensão de PP e MBP 4.Anos 5.Horas
Medidas derivadas	1.Tamanho 2.Esforço estimado (E_{DEV}) 3.Duração estimada (t_d) 4.Estimativa acumulada do esforço ($E(t)$)
Função de medição	1. tamanho = funcionalidade modificada * <i>gearing factor</i> 2. $E_{DEV} = (0,39 \text{ MBP})^{4/7} \text{ PP}^{-9/7} (\text{tamanho } \beta^{1/3})^{9/7}$ 3. $t_d = (0,39 \text{ MBP})^{-1/7} \text{ PP}^{-3/7} (\text{tamanho} * \beta^{1/3})^{3/7}$ 4. $E(t) = (E_{DEV} / 0,39) [1 - \exp(-t^2 / (2 t_d^2))]$
Indicador	Gráfico comparando o esforço real com o esforço acumulado estimado até o momento.
Modelo de análise	Analisar causa e determinar ação corretiva.
Critérios de decisão	Diferença entre estimativa de esforço acumulado e esforço real despendido superior a um desvio padrão da estimativa.
Aplicabilidade	Fase de construção

4.5.6 Utilização de recursos incompatível com data de conclusão do projeto

O risco de que o esforço esteja sendo despendido de forma incompatível com a data de conclusão ou duração do projeto está associado ao componente de risco que diz respeito ao uso de recursos e desempenho do projeto. Para tratar esse risco, propõe-se um indicador que compare, a cada revisão, a produtividade que o projeto vem conseguindo com a produtividade utilizada no planejamento do projeto. As sucessivas comparações permitem verificar se o risco está se manifestando.

O indicador proposto responde à questão sobre se há compatibilidade entre a utilização real de esforço e a geração de resultados até a data de cada revisão. O indicador consiste de gráfico que anota, a cada ponto de revisão do projeto, a produtividade conseguida até então, com aquela prevista para o projeto. Caso se faça replanejamento do projeto, a nova produtividade pode ser utilizada na comparação.

A Tabela 4-11 representa esquematicamente a associação entre variável de risco e indicador que detecta a manifestação do risco.

Tabela 4-11 – Risco de custos estimados e reais em descompasso

Variável de risco		Componente de risco afetado	
Utilização de recursos incompatível com data de conclusão		Uso de recursos e desempenho	
Questões	O esforço despendido pelo projeto até o momento é compatível com os produtos gerados?		
Indicador	Gráfico comparando o coeficiente de produtividade conseguido até o momento com o coeficiente de produtividade estimado para o projeto.	I-6	

Discussão

O indicador de produtividade da organização (PP), expresso através do parâmetro de produtividade definido por Putnam, é um dos elementos usados para estimativa de esforço e duração dos projetos. É uma medida calculada através do histórico dos projetos a partir de relações definidas pelo modelo de Putnam e é descrita na seção 3.3.1.1. Conforme se depreende da leitura da descrição desse parâmetro ele é dependente da velocidade de execução do projeto e, portanto não se confunde com o que (Putnam, 1992) denomina “produtividade convencional”. A produtividade convencional é usualmente medida

dividindo-se o resultado produzido por unidade de esforço. Normalmente, é expressa por pontos de função por homem-mês.

Também se calcula coeficiente de produtividade para o projeto. O coeficiente de produtividade de um projeto é medida derivada definida com base no esforço requerido para a produção de artefatos do projeto. É comumente expresso por horas para produzir um ponto de função ou horas para produzir um milhar de linhas de código. Nos pontos intermediários de revisão de projeto, pode-se medir esse coeficiente com base nas linhas de código dos artefatos efetivamente testados e colocados sob controle de configuração. Cabe observar, que o que denominamos coeficiente de produtividade é dimensionalmente o inverso da produtividade (Stutzke, 2005).

É importante para o gerente do projeto, portanto, determinar se o projeto está alcançando o coeficiente de produtividade previsto para o projeto ao longo do ciclo de vida do desenvolvimento, comparando-o com o valor do coeficiente obtido até o momento.

Em função do gradual aumento da precisão de estimativa do tamanho de projeto (*funnel graph*), a precisão do coeficiente de produtividade também aumentará com o desenrolar do projeto. Mesmo assim, interessa calcular o coeficiente de produtividade que está sendo alcançado pelo projeto, o mais cedo possível, a fim de detectar variações significativas e tomar as medidas corretivas necessárias.

Indicador

O indicador da Tabela 4-12, que se descreve através de estrutura de medição, compara o coeficiente de produtividade que o projeto demonstra alcançar em suas revisões periódicas de progresso com o coeficiente de produtividade previsto ao seu início. A estimativa de esforço é obtida na subseção 3.3.1.2. Os valores reais do esforço despendido são obtidos de relatórios de reporte de horas utilizadas no projeto.

Para efeito do indicador, o valor intermediário de pontos de função desenvolvidos é estimado a partir da divisão da quantidade de linhas de código produzidas pelo *gearing factor*. Essa técnica, conhecida como *backfiring*, embora não seja recomendada para estimativa de tamanho de sistemas (Silveira, sem data), é justificada neste caso, por utilizar o mesmo *gearing factor* usado na estimativa inicial do projeto.

Tabela 4-12 – Indicador de acompanhamento de produtividade

I-6 - Comparação produtividade estimada e real	
Necessidade de informação	O esforço despendido pelo projeto até o momento é compatível com os produtos gerados?
Conceito mensurável	Desempenho do projeto
Entidades relevantes	A fase de construção
Atributos	Produtividade
Medidas primitivas	1.Funcionalidade instalada (instalada) 2. <i>Gearing Factor</i> 3.Parâmetros PP e MBP 4.Linhas de código geradas e testadas sob controle de configuração (geradas) 5.Esforço na fase de construção até o momento (excluindo retrabalho por mudança de requisitos) (esforço)
Método de medição	1.Regras de contagem do IFPUG 4.2 2.Média da indústria 3.Histórico dos projetos anteriores(esforço, duração, tamanho). 4.Contagem de linhas de código 5.Reporte de horas
Tipo de método	1.Objetivo 2.Objetivo 3.Objetivo 4.Objetivo 5.Objetivo
Escala	1.Números inteiros 2.Números reais 3.Numeros reais 4.Números inteiros 5.Números reais
Tipo de escala	1.Razão 2.Razão 3.Razão 4.Razão 5.Razão
Unidade de medida	1.Pontos de função 2.Linhas de código por ponto de função 3.Dimensão de PP e MBP 4.Linhas de código não comentário 5.Horas
Medidas derivadas	1.tamanho 2.Esforço estimado (E_{DEV}) 3.Coefficiente de produtividade do projeto (fase de construção) (hrs/PF) 4.Coeff. de produtividade real do projeto, a cada revisão (real) (hrs/PF)
Função de medição	1. tamanho = funcionalidade instalada * <i>gearing factor</i> 2. $E_{DEV} = (0,39 MBP)^{4/7} PP^{-9/7} (\text{tamanho } \beta^{1/3})^{9/7}$ 3. Coeficiente de produtividade do projeto = $E_{DEV} / \text{instalada}$ 4. Real = esforço / (geradas / <i>gearing factor</i>)
Indicador	Gráfico comparando o coeficiente de produtividade conseguido até o momento com o coeficiente de produtividade estimado para o projeto.
Modelo de análise	Analisar causa de coeficiente de produtividade inferior e determinar ação corretiva.
Critérios de decisão	Coeficiente de produtividade real inferior ao planejado em 20% na primeira metade de fase de transição e 10% na segunda metade.
Aplicabilidade	Fase de construção

Usos do indicador no gerenciamento de riscos

O indicador permitirá ao gerente do projeto averiguar a manifestação do risco, associado às questões abaixo, quando os critérios de decisão descritos na Tabela 4-12 se verificarem:

- A equipe do projeto é capacitada para o desenvolvimento do sistema?
- Há demora dos usuários em esclarecerem pontos não claros dos requisitos?
- As mudanças de escopo e de requisitos estão sendo consideradas para efeito de re-estimativa e seu impacto no esforço do projeto está sendo calculado?
- O retrabalho causado por mudanças de requisitos está sendo alocado de forma separada das horas despendidas na produção dos artefatos que efetivamente constituirão o produto?
- O ambiente de desenvolvimento é um obstáculo para que se consiga produtividade elevada?
- Os fatores ambientais (barulho, iluminação, mobiliário) levam à perda de produtividade do projeto?

4.5.7 Estimativa de necessidade de pessoal

Um dos riscos que é preciso acompanhar durante a execução de um projeto é se a equipe necessária foi alocada de forma adequada e a tempo. Para acompanhar o risco em questão, cabe comparar a previsão de alocação de pessoal do projeto feita durante o planejamento com a real alocação ao longo do tempo. Essa comparação será feita por ocasião das revisões periódicas do projeto. As sucessivas comparações permitem verificar se o risco está se manifestando.

O indicador proposto responde à questão sobre se há variação significativa, segundo critério de decisão estabelecido, entre o número de profissionais alocados a cada momento e o número previsto. Essa verificação pode ser feita em termos mensais ou quinzenais no caso de projetos de médio porte, dos quais se ocupa esse trabalho. O indicador consiste de gráfico que anota, a cada ponto de revisão do projeto, o quadro de mão-de-obra desde o início da fase de construção e o compara com a curva projetada ao início do projeto. Caso se faça replanejamento do projeto, a nova curva de utilização de mão-de-obra poderá ser utilizada na comparação.

A Tabela 4-13 representa esquematicamente a associação entre variável de risco e indicador que detecta a manifestação do risco.

Tabela 4-13 – Risco de estimativa de pessoal

Variável de risco		Componente de risco afetado
Estimativa de necessidade de pessoal		Cronograma e duração Funcionalidades do sistema
Questões	Há variação entre o quadro de pessoal estimado do projeto e o quadro efetivamente alocado?	
Indicador	Gráfico comparando a alocação prevista de pessoal e a alocação real.	I-7

Discussão

Importa ao gerente do projeto conhecer, tão logo seja possível, a estimativa de esforço do projeto, que será base para diversas outras estimativas de medidas derivadas entre as quais o cálculo de custo do projeto, o cálculo de alocação de mão-de-obra por fase e a estimativa de esforço acumulado a cada instante.

O esforço estimado do projeto é obtido a partir da medição do tamanho funcional modificado do produto, e sua subsequente transformação em linha de código, como passo intermediário. Utiliza-se a funcionalidade modificada, ao invés da funcionalidade instalada, uma vez que a primeira inclui funcionalidade que foi excluída e modificada, permitindo melhor comparação com o esforço efetivamente despendido em retrabalho.

Utilizam-se as equações paramétricas de Putnam para chegar-se à estimativa de esforço, utilizando-se os indicadores de produtividade e de *manpower buildup* descritos no Capítulo 3. A estimativa de esforço será refeita a cada variação do tamanho do projeto.

O esforço estimado total do projeto é então distribuído ao longo do período de duração do projeto com a utilização da distribuição de Rayleigh, na forma descrita na seção 3.3.2. A partir da distribuição do esforço no tempo pode-se determinar o quadro de pessoal a alocar ao projeto. O quadro real pode-se determinar das listas de pessoal do projeto, buscando-se a certeza de que todos estejam efetivamente nele ocupados em tempo integral, sem o que o indicador proposto perde efetividade.

Indicador

O indicador da Tabela 4-14, que se descreve através de estrutura de medição, aponta variação entre o quadro de pessoal estimado a cada momento para o projeto e o real quadro alocado. O quadro estimado é obtido a partir da estimativa de distribuição de esforço descrita na subseção 3.3.2, especificamente a fórmula [3-6]. Os valores reais do quadro alocado devem ser obtidos das listas de pessoal do projeto.

A verificação da existência de variação significativa entre o quadro previsto e o efetivamente alocado será feita de forma visual. Há que investigar se a concordância entre a estimativa e o real se realiza desde o início do projeto, uma vez que a falta de pessoal no início poderá levar à compressão de projeto no fim, o que poderá comprometer a conclusão do projeto no prazo.

Usos do indicador no gerenciamento de riscos

O indicador permitirá ao gerente do projeto averiguar a manifestação do risco, associado às questões abaixo, quando os critérios de decisão descritos na Tabela 4-14 se verificarem:

- Há desvio de recursos para atividades alheias ao projeto, embora estejam formalmente alocados ao projeto?
- A mão-de-obra requerida pelo projeto foi alocada na ocasião adequada, ou houve períodos em que esteve abaixo do normal, acarretando atrasos difíceis de recuperar?
- Se há concordância entre as estimativas de alocação de pessoal e a real alocação e ainda assim o projeto se encontra atrasado, segundo os outros indicadores, a equipe tem a qualificação necessária?
- Se há concordância entre as estimativas de alocação de pessoal e a real alocação e ainda assim o projeto se encontra atrasado, segundo os outros indicadores, as mudanças de escopo e de requisitos estão sendo consideradas para efeito de revisão do tamanho da equipe do projeto?

Tabela 4-14 – Indicador de alocação de pessoal

I-7 - Comparação entre alocação prevista de pessoal e a alocação real.	
Necessidade de informação	Há variação entre o quadro de pessoal estimado do projeto e o quadro efetivamente alocado?
Conceito mensurável	Quadro do pessoal do projeto
Entidades relevantes	A fase de construção
Atributos	Quantidade de mão-de-obra
Medidas primitivas	1.Funcionalidade modificada 2. <i>Gearing Factor</i> 3.Parâmetros PP e MBP 4.Tempo decorrido do projeto 5.Quadro alocado no projeto a cada ponto de revisão
Método de medição	1.Regras de contagem do IFPUG 4.2 2.Média da indústria 3.Histórico dos projetos anteriores(esforço, duração, tamanho) 4.Medição de tempo calendário 5.Contagem de pessoal
Tipo de método	1.Objetivo 2.Objetivo 3.Objetivo 4.Objetivo 5.Objetivo
Escala	1.Números inteiros 2.Números reais 3.Números reais 4.Números reais 5.Números inteiros
Tipo de escala	1.Razão 2.Razão 3.Razão 4.Razão 5.Razão
Unidade de medida	1.Pontos de função 2.Linhas de código por ponto de função 3.Dimensão de PP e MBP 4.Anos 5.Pessoas
Medidas derivadas	1.tamanho 2.Esforço estimado (E_{DEV}) 3.Derivada da estimativa do esforço ($E'(t)$) 4.Quadro estimado (Q_{est})
Função de medição	1. tamanho = funcionalidade modificada * <i>gearing factor</i> 2. $E_{DEV} = (0,39 MBP)^{4/7} PP^{-9/7}$ (tamanho $\beta^{1/3}$) ^{9/7} 3. $E'(t) = [E_{DEV} / (0,39 t_d^2)] t \exp(-t^2/(2 t_d^2))$ 4. $Q_{est}(t) \sim E'(t)$
Indicador	Gráfico comparando a alocação prevista de pessoal e a alocação real.
Modelo de análise	Analisar causa e determinar ação corretiva.
Critérios de decisão	Diferença entre quadro de pessoal estimado e quadro real.
Aplicabilidade	Fase de construção

4.5.8 Uso uniforme do tempo do projeto

Para identificar riscos de falta de uso uniforme do tempo do projeto na fase de construção, cabe verificar se a produção de linhas de código ao longo do tempo se comporta da forma estimada no planejamento do projeto. A produção de linhas de código deve se dar de forma proporcional ao consumo do esforço, que por sua vez, se comporta segundo a distribuição de Rayleigh. Essa verificação será feita por ocasião das revisões periódicas do projeto.

O indicador proposto responde à questão sobre se o tempo transcorre sem que haja a produção de linhas de código esperadas. O indicador consiste de gráfico que anota, a cada ponto de revisão do projeto, a produção acumulada de linhas de código desde o início da fase de construção e a compara com a curva projetada ao início do projeto. As linhas de código que devem ser contadas são as que correspondem a artefatos produzidos, testados e colocados sob controle de configuração.

A Tabela 4-15 representa esquematicamente a associação entre variável de risco e indicador que detecta a manifestação do risco.

Tabela 4-15 – Risco de uso uniforme do tempo do projeto

Variável de risco		Componente de risco afetado	
Uso uniforme do tempo de projeto		Gerenciamento de requisitos	
Questões	Qual o desvio entre o número de linhas de código concluídas e as previstas até a data?		
Indicador	Gráfico mostrando o número de linhas de código planejadas e a quantidade efetivamente concluída segundo o critério estabelecido pelo projeto.		I-8

Discussão

A equipe do projeto deve preocupar-se em identificar, desde o começo do desenvolvimento, se o produto está evoluindo de forma consistente com o transcurso do tempo e com o gasto das horas de esforço previstas. O progresso pode ser acompanhado diretamente através da contagem dos artefatos produzidos, a partir do momento em que haja o planejamento detalhado de que artefatos serão produzidos e quando. Também se pode acompanhar o progresso do projeto através do acompanhamento da quantidade de linhas de códigos dos artefatos produzidos, testados e colocados sob controle de configuração. Essa quantidade pode ser contada com ferramentas apropriadas durante o projeto e ao seu final.

Nesse último caso, é necessário criar uma estimativa para o número de linhas de código concluídas ao longo do projeto. Essa previsão pode ser feita com a utilização do modelo paramétrico exposto no capítulo 3, como se descreve a seguir:

O esforço estimado do projeto é obtido a partir da medição do tamanho funcional do produto, e sua subsequente transformação em linha de código, como passo intermediário. As equações paramétricas de Putnam permitem chegar-se à estimativa de esforço, utilizando os parâmetros de produtividade e de *manpower buildup* descritos no Capítulo 3. Em seguida, o esforço é distribuído ao longo do tempo com base nas fórmulas desenvolvidas na seção 3.3.2, notadamente a fórmula [3-7]. A previsão de produção de linhas de código far-se-á de forma proporcional à previsão de horas trabalhadas pela equipe do projeto.

A comparação entre linhas estimadas e reais deve ser feita, eliminando das linhas reais, aquelas que foram excluídas em função de retrabalho, quer em razão de eliminação de defeitos, quer por mudança e exclusão de requisitos. Para que a comparação funcione de forma adequada, a organização deve utilizar processos de gerenciamento de configuração para eliminar do repositório de artefatos do projeto, o código tornado obsoleto por eliminação e alteração de requisitos.

Indicador

O indicador da Tabela 4-16, que se descreve através de estrutura de medição, aponta variação entre as linhas de código previstas e aquelas efetivamente produzidas e testadas.

Para tanto, utilizar-se-á gráfico que mostra, nas ocasiões de revisão de progresso do projeto, a previsão de linhas que devem ter sido produzidas e a quantidade real.

O indicador determina que, caso a quantidade produzida esteja abaixo de 90% da estimada, em qualquer ponto da fase de construção, deve-se buscar a causa raiz e solucionar os problemas que estão ameaçando o progresso do projeto.

Tabela 4-16 – Indicador de produção de linhas de código

I-8 - Comparação entre linhas de código planejadas e prontas	
Necessidade de informação	Qual o desvio entre o número de linhas de código concluídas e as previstas até a data?
Conceito mensurável	Tamanho físico do produto
Entidades relevantes	Código fonte
Atributos	Linhas de código
Medidas primitivas	1. Funcionalidade instalada 2. <i>Gearing Factor</i> 3. Parâmetros PP e MBP 4. Tempo decorrido do projeto (t) 5. Linhas de código dos artefatos desenvolvidos e testados (concluídas)
Método de medição	1. Regras de contagem do IFPUG 4.2 2. Média da indústria 3. Histórico dos projetos anteriores (esforço, duração, tamanho) 4. Medição de tempo calendário 5. Relatório de gerencia de configuração
Tipo de método	1. Objetivo 2. Objetivo 3. Objetivo 4. Objetivo 5. Objetivo
Escala	1. Números inteiros 2. Números reais 3. Números reais 4. Números reais 5. Números inteiros
Tipo de escala	1. Razão 2. Razão 3. Razão 4. Razão 5. Razão
Unidade de medida	1. Pontos de função 2. Linhas de código por ponto de função 3. Dimensão de PP e MBP 4. Anos 5. Linhas de código
Medidas derivadas	1. tamanho 2. Esforço estimado (E_{DEV}) 3. Duração estimada (t_d) 4. Estimativa acumulada do esforço ($E(t)$) 5. Linhas de código planejadas ($L(t)$)
Função de medição	1. tamanho = funcionalidade instalada * <i>gearing factor</i> 2. $E_{DEV} = (0,39 MBP)^{4/7} PP^{-9/7} (\text{tamanho } \beta^{1/3})^{9/7}$ 3. $t_d = (0,39 MBP)^{-1/7} PP^{3/7} (\text{tamanho} * \beta^{1/3})^{3/7}$ 4. $E(t) = (E_{DEV} / 0,39) [1 - \exp(-t^2 / (2 t_d^2))]$ 5. $L(t) = (E(t) * E_{DEV}) * \text{tamanho}$
Indicador	Gráfico mostrando o número de linhas de código planejadas até o momento e a quantidade efetivamente concluída segundo o critério estabelecido pelo projeto.
Modelo de análise	Analisar causa e determinar ação corretiva.
Crítérios de decisão	Número de linhas código efetivamente desenvolvidas menor do que 90% das linhas planejadas.
Aplicabilidade	Fase de construção

Usos do indicador no gerenciamento de riscos

O indicador permitirá ao gerente do projeto averiguar a manifestação do risco, associado às questões abaixo, quando os critérios de decisão descritos na Tabela 4-16 se verificarem:

- Há excessivo retrabalho, em razão de defeitos ou mudança de requisitos, que impede a conclusão dos artefatos previstos?
- Há problemas que estão afetando a produtividade da equipe?
- Houve atraso na alocação de mão de obra em relação à estimativa de montagem da equipe?
- Há suficiente trabalho em execução, que quando colocado sob controle de configuração, recupere o atraso indicado pelo indicador?

4.5.9 Dificuldade em gerenciar a complexidade do projeto.

Há inúmeras razões pelas quais um projeto se pode considerar complexo: dificuldade algorítmica, ambiente tecnológico não suficientemente conhecido, tamanho do aplicativo, grande número de usuários com diferentes necessidades, etc. Dessa forma, um indicador que descreva quão bem um projeto gerencia sua complexidade necessita ser genérico. Adotar-se-á um indicador que reflita, ao longo do tempo, como o projeto está satisfazendo os requisitos de seus usuários. Esse indicador é especificado através da comparação entre a quantidade de casos de testes gerados e a quantidade que passou nos testes de requisitos e desempenho a que foram submetidos.

A Tabela 4-17 representa esquematicamente a associação entre variável de risco e indicador que detecta a manifestação do risco.

Tabela 4-17 – Risco de complexidade do projeto

Variável de risco		Componente de risco afetado	
Dificuldade em gerenciar complexidade do projeto		Cronograma e duração Uso de recursos e desempenho	
Questões	O sistema está atendendo aos requisitos especificados pelos usuários e aos critérios de desempenho dos requisitos não-funcionais?		
Indicador	Gráfico registrando os casos de teste especificados e a quantidade dos casos que efetivamente passaram.		I-9

Discussão

O gerente do projeto deve, desde as fases iniciais do projeto, ocupar-se em descrever e preparar casos de testes para os requisitos funcionais e não funcionais do sistema. Com isso, o projeto tem condição de ir aplicando esses casos de testes na verificação e validação da aplicação produzida. Os testes se devem aplicar tão cedo quanto possível, pois quanto mais tempo transcorre desde a injeção de um defeito até sua eliminação, maior o custo de correção.

O projeto deve, desde o começo, cuidar de quantificar os casos de teste e comparar seu número com o de casos de testes executados com sucesso.

Indicador

O indicador da Tabela 4-18, que se descreve através de estrutura de medição, aponta variação entre o número de casos de teste e a quantidade de casos de teste executados com sucesso.

Essa verificação é feita a cada revisão do projeto, embora nas primeiras revisões seja normal que poucos casos de testes tenham sido executados, ou mesmo nenhum.

O indicador é utilizado para verificar se o número de casos de testes bem sucedidos tende ao número total de casos de teste, à medida que a data de conclusão do projeto se aproxima.

Usos do indicador no gerenciamento de riscos

O indicador permitirá ao gerente do projeto averiguar a manifestação do risco, associado às questões abaixo, quando os critérios de decisão descritos na Tabela 4-18 se verificarem:

- Há problemas não resolvidos nos projetos e que impedem a conclusão dos testes de caso de uso?
- Os casos de teste foram elaborados de forma adequada, e, tanto quanto possível, já nas fases iniciais do projeto?
- Há alguma área que apresente considerável taxa de defeitos?

Tabela 4-18 – Indicador de complexidade do projeto

I-9 - Complexidade do projeto	
Necessidade de informação	O sistema está atendendo aos requisitos especificados pelos usuários e aos critérios de desempenho dos requisitos não-funcionais?
Conceito mensurável	Correção funcional
Entidades relevantes	Casos de teste
Atributos	Sucesso no teste
Medidas primitivas	1.Total de casos de testes 2.Total de casos de testes bem sucedidos
Método de medição	1.Contagem 2.Contagem
Tipo de método	1.Objetivo 2.Objetivo
Escala	1.Números inteiros 2.Números inteiros
Tipo de escala	1.Razão 2.Razão
Unidade de medida	1.Casos de teste 2.Casos de teste
Medidas derivadas	1.Total de casos de testes 2.Total de casos de testes bem sucedidos
Função de medição	1.Contagem 2.Contagem
Indicador	Gráfico registrando os casos de teste especificados e a quantidade dos casos que efetivamente passaram.
Modelo de análise	Análise de causa e re-estimativa de esforço e prazo, deflagrada pelo critério de decisão.
Critérios de decisão	Diferença entre total de casos de teste e total de casos de teste bem sucedidos não tende a zero à medida que se aproxima data de conclusão do projeto.
Aplicabilidade	Fase de Construção

4.6 Enquadramento dos indicadores no ciclo de desenvolvimento de software e modelo CMMI

A Figura 4-5 indica, no âmbito do Processo Unificado, as fases e disciplinas em que podem se aplicar os indicadores definidos no presente trabalho.

Figura 4-5 – Aplicabilidade dos indicadores no Processo Unificado

		Fases			
		Iniciação	Elaboração	Construção	Transição
Disciplinas Centrais	Modelagem de Negócios				
	Requisitos	Aplicável	Aplicável	Aplicável	
	Análise e projeto		Aplicável	Aplicável	
	Implementação		Aplicável	Aplicável	Aplicável
	Teste		Aplicável	Aplicável	Aplicável
	Instalação				
Disciplinas de Suporte	Configuração e gerenciamento de mudanças		Aplicável	Aplicável	Aplicável
	Gerenciamento de projeto	Aplicável	Aplicável	Aplicável	Aplicável
	Ambiente				

Adaptado de (Krutchen, 2003)

Por outro lado, a Figura 4-6 indica as áreas de processo do modelo CMMI, em sua versão 1.2 (Chrissis, 2007), para as quais os conceitos definidos no presente trabalho são relevantes.

Figura 4-6 – Relevância dos conceitos para o modelo CMMI

Nível 2		Nível 3		Nível 4		Nível 5	
Área de Processo	Relevante?	Área de Processo	Relevante?	Área de Processo	Relevante?	Área de Processo	Relevante?
REQM	Sim	RD	Sim	OPP		OID	
PP	Sim	TS	Sim	QPM	Sim	CAR	Sim
PMC	Sim	PI					
SAM		VER	Sim				
MA	Sim	VAL					
PPQA	Sim	OPF					
CM		OPD					
		OT					
		IPM					
		RISKM	Sim				
		DAR					

4.7 Apresentação do estado final da matriz de risco

Nesta seção, concluímos o processo de criação dos indicadores, ilustrado na Figura 4-2, com a apresentação do estado da matriz de riscos, após a sua atualização com a identificação dos indicadores que tratam cada uma das variáveis de risco. Pode-se observar, na Tabela 4-19,

que se definiram indicadores para todas as variáveis de risco cobertas pelo escopo do trabalho, ou seja, aquelas que não foram excluídas em função das limitações definidas.

Tabela 4-19 – Matriz de risco com indicadores

Variáveis de risco	Componentes de risco					
	Cronograma e duração	Funcionalidades do sistema	Subcontratação	Gerenciamento de requisitos	Uso de recursos e desempenho	Gerenciamento de pessoal
Deficiências de pessoal						NA
Problemas de cronograma	I-3					
Utilização de recursos incompatível com data de conclusão					I-6	
Custos estimados e custos reais em descompasso	I-5					
Estimativa incorreta de tamanho	I-2					
Estimativa de necessidade de pessoal	I-7	I-7	NA			
Uso uniforme do tempo de projeto				I-8		NA
Expertise insuficiente em métodos, software e equipamentos						NA
Dificuldade em gerenciar complexidade do projeto	I-9				I-9	
Funções do software desenvolvido em descompasso com necessidades dos usuários		NA				
Software inclui funcionalidade apenas marginalmente necessária ("gold plating")				NA		
Mudança constante de requisitos				I-1		
Cronograma mudando continuamente	I-4			I-4		
Usuários descontentes com interface do usuário implementada		NA				
Componentes adquiridos externamente não atendem suas especificações			NA			
Expectativas irrealistas das habilidades dos membros da equipe do projeto						NA
Requisitos de desempenho (tempo de resposta, etc) estimados incorretamente					NA	NA
Tarefas subcontratadas realizadas de forma insatisfatória			NA			
Estimativa inadequada da infraestrutura de hardware e software		NA			NA	

	Variáveis que contribuem para o componente de risco e respectivo indicador
NA	Não aplicável

Adaptado de (Ropponen, 2000)

Para facilidade de referência, reproduzimos aqui o problema de pesquisa enunciado na Introdução:

Pode um conjunto de indicadores objetivos e quantificáveis usados em avaliações periódicas de status de um projeto evidenciar a manifestação dos riscos mais frequentes com que os projetos se deparam em sua fase de construção?

Como cada um dos indicadores descritos nas estruturas de medição deste Capítulo identifica adequadamente a ocorrência do risco ao qual foi associado, o preenchimento da matriz, na forma da Tabela 4-19, constitui evidência de que o processo apresentado responde afirmativamente ao problema de pesquisa.

5. CONCLUSÃO

Neste Capítulo, apresenta-se breve resumo do trabalho em retrospectiva, discute-se sua contribuição, apontam-se suas limitações, e delineiam-se possíveis desdobramentos em possíveis trabalhos futuros.

5.1 Retrospectiva do trabalho

Este trabalho investigou, de forma sistemática, o processo de criação de indicadores objetivos e quantitativos, definidos de forma a detectar a ocorrência dos principais riscos que afetam os projetos. O trabalho utilizou-se dos seguintes componentes principais:

- Descrição pormenorizada do processo de estimativa de desenvolvimento de software, com ênfase no modelo paramétrico de Putnam;
- Conjunto consolidado de variáveis de riscos de projeto, desenvolvido em (Ropponen, 2000), a partir de lista anterior definida empiricamente por Barry Boehm;
- Princípios e práticas do PSM, que definem as medidas a coletar, a calcular e a analisar, inclusive indicadores, com base nas necessidades de informação do projeto e dos indivíduos e entidades que com ele interagem.

Esses três componentes, usados em conjunto, permitiram a definição dos indicadores a partir do conjunto de variáveis de risco com a utilização das práticas do PSM, servindo o modelo paramétrico de Putnam para detalhar a implementação dos indicadores, de forma a evidenciar a viabilidade de sua efetiva criação.

À vista dos resultados conseguidos, concluiu-se que é viável a definição de indicadores objetivos que permitam aos gerentes de projeto acompanhar os riscos de seus projetos de forma objetiva e quantitativa, respondendo afirmativamente à questão de pesquisa.

5.2 Contribuição do trabalho

Os três principais componentes utilizados no trabalho – modelos de estimativa de software, lista de variáveis de risco e os princípios e práticas do PSM – já estão bastante bem descritos na literatura, tanto em artigos quanto em livros.

A contribuição principal do trabalho foi sua utilização conjunta na resposta à indagação se, a partir de uma lista de riscos, é possível construir um conjunto de indicadores que permitam identificar quantitativamente a manifestação dos riscos. A ampla referência aos modelos de

estimativa permitiu evidenciar a viabilidade de se definirem e criarem os indicadores, de forma precisa, segundo as práticas do PSM.

Do ponto de vista concreto, os indicadores definidos no presente trabalho – ou indicadores semelhantes criados com base nas práticas descritas e a partir de outras listas de risco – podem ser utilizados para efeito de acompanhamento dos projetos de desenvolvimento de forma quantitativa e menos subjetiva. Essa contribuição prática é reforçada com a apresentação, em apêndice, de simulação da utilização dos indicadores no acompanhamento de um projeto fictício, de forma a ilustrar a utilização dos indicadores propostos.

5.3 Limitações do trabalho

A pesquisa limitou as variáveis de risco para as quais se iriam criar indicadores. Dessa maneira, só se consideraram riscos que se relacionam com as atividades da fase de construção, conforme definida pelo Processo Unificado e com as principais disciplinas executadas nessa fase. Também se excluíram da pesquisa os riscos que não possam ser tratados pelo gerente do projeto ou por sua equipe, ou cujo tratamento requeira, principalmente, o envolvimento de pessoas ou entidades externas à equipe de projeto.

Dessa forma, ficaram de fora da análise fatores culturais e os relacionados à comunicação e relacionamento entre os participantes e interessados no projeto. Entretanto, alguns desses riscos tiveram sua manifestação questionada nas seções “Usos do indicador no gerenciamento de riscos”, presentes após a descrição de cada uma das estruturas de medição com a definição do indicador.

A pesquisa também limitou o uso dos indicadores definidos para projetos de médio porte, projetos com duração de 3 a 12 meses e com equipe de desenvolvedores de 5 até 25 profissionais, segundo definição de (McConnell, 2006). Projetos de pequeno porte, em geral, apresentam alocação constante de pessoal, não apresentando a característica curva da equação de Rayleigh.

Devido ao tempo que demanda o desenvolvimento de sistemas de médio porte, foi inviável aplicar os indicadores propostos a um conjunto de projetos em desenvolvimento em uma organização para avaliar sua eficácia na eliminação de riscos.

5.4 Possíveis extensões para o trabalho

O presente trabalho poderá ser estendido em vários aspectos, dentre os quais destacamos:

- Desenvolvimento dos indicadores a partir da lista de riscos mais comuns que se encontram em uma entidade ou empresa. Essa lista de risco mais comum pode ser desenvolvida a partir da análise de revisão dos projetos concluídos;
- Aplicar os indicadores especificados a um conjunto de projetos de uma organização, visando obter os benefícios de sua aplicação;
- Se a aplicação dos indicadores se der por longo tempo, seria possível determinar se a adoção da revisão de projetos com base neles contribuiu para a melhoria dos projetos. Para tanto, pode-se adotar, à semelhança de (Galín, 2006), um conjunto de medidas a serem obtidas antes e após a implementação das avaliações sistemáticas de *status* periódico de projetos, tais como:
 - Densidade de defeitos;
 - Produtividade de desenvolvimento (em horas por ponto de função ou milhares de linhas de código);
 - Porcentagem de retrabalho;
 - Cumprimento de cronogramas.

Isso se faria através de teste de hipótese, com formulação da hipótese estatística, da hipótese nula, determinação do grau de significância adotado e do critério de decisão que refutará ou não a hipótese nula com base no grau de significância determinado (Salomon, 2001);

- Incluir indicadores adicionais. O PSM contém extenso catálogo de requisitos de informação dos projetos, conceitos mensuráveis e medidas de interesse. (Florac, 1999) traz listas de variáveis que os projetos podem considerar para gerenciamento quantitativo, no espírito das práticas de nível 4 e 5 do CMMI, que também podem ser usadas para os objetivos de controle dos projetos. (Fernandes, 2003) apresenta indicadores e medidas operacionais que podem ser usados por uma fábrica de software para planejamento do projeto e gerenciamento do desenvolvimento do produto.

- Adaptar o trabalho para utilizar outros métodos de determinação de tamanho de software e modelos paramétricos de estimativa. São candidatos para essa extensão a contagem de Use Case Points e o modelo COCOMO II;
- Sofisticar a planilha usada na simulação apresentada no Apêndice, transformando-a numa ferramenta generalizada que possa ser usada no acompanhamento de projetos.

REFERÊNCIAS BIBLIOGRÁFICAS

- AHERN, D. M.; CLOUSE, A.; TURNER, R. **CMMI distilled: a practical introduction to integrated process improvement**. Boston: Addison-Wesley. 2003.
- ARMOUR, P. **Software: Hard Data: Seeking solid information on the behavior of modern projects**. Communications of the ACM. September 2006.
- CHRISSIS, M.; KONRAD, M.; SHRUM, S. **CMMI: Guidelines for Process Integration and product improvement**. 2a. Ed. Addison-Wesley. 2007.
- BOEHM, B. W. **Software Engineering Economics**. Prentice Hall PTR. 1981.
- BOEHM, B.; ABTS, C. **Software Cost Estimation with Cocomo II**. Prentice Hall PTR. 2000.
- BROOKS, F. P. **The mythical man-month: essays on software engineering**. Reading, Mass.: Addison-Wesley Pub. Co. 1975.
- DeMARCO, T. **Controlling Software Projects: Management, Measurement, and Estimates**. Prentice Hall PTR. 1982.
- DeMARCO, T.; LISTER, T.. **Waltzing with Bears: Managing Risk on Software Projects**. Dorset House Publishing Company, Incorporated .2003.
- FERNANDES, A.; TEIXEIRA, D.. **Fábrica de Software**. Atlas. 2003.
- FERNANDES, A.; ABREU, V.. **Implantando a Governança em TI: da estratégia à gestão dos processos e serviços**. Brasport. 2006.
- FLORAC, W. A.; CARLETON, A. **Measuring the Software Process: statistical process control for software process improvement**. Reading, Mass.: Addison-Wesley. 1999.
- GALIN, D.; AVRAHAMI, M.. **Are CMM Program Investments Beneficial? Analyzing past studies**. IEEE Software, Vol. 23, No. 6, 2006.
- GARMUS, D.; HERRON, D. **Function Point Analysis: measurement practices for successful software projects**. Boston: Addison-Wesley. 2001.
- GIL, A. C. **Como elaborar projetos de pesquisa**. Atlas. 4a Ed. 2002.
- HUMPHREY, W. S. **Managing the Software Process**. Addison-Wesley Professional. 1989.
- HUMPHREY, W. S. **A Discipline for Software Engineering**. Addison-Wesley Professional. 1995.
- IFPUG – International Function Point Users Group. **Manual de Práticas de Contagem de Pontos de Função Versão 4.2.1**. 2005.
- JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. **The Unified Software Development Process**. Reading, Mass: Addison-Wesley. 1999.

- KRUCHTEN, P. **The Rational Unified Process**: an introduction. Boston: Addison-Wesley. 2003.
- LARMAN, C. **Applying UML and Patterns**: An Introduction to Object-Oriented Analysis and Design and the Unified Process (2nd Edition). Prentice Hall PTR. 2001.
- LARMAN, C. **Agile and Iterative Development**: A manager's guide. Addison Wesley. 2004.
- MCCONNELL, S. **Software Estimation**: Demystifying the Black Art. Microsoft Press. 2006.
- MCGARRY, J. et al. **Practical Software Measurement**: Objective Information for Decision Makers. Addison-Wesley Professional. 2002.
- PAULK, M. **Extreme Programming from a CMM Perspective**. IEEE Software. Vol. 18, No. 6. 2001
- POPPENDIECK, M.; POPPENDIECK, T. **Lean Software Development**: An Agile Toolkit for Software Development Managers. Addison-Wesley Professional. 2003.
- PRESSMAN, R. **Software Engineering**: A Practitioner's Approach. McGraw-Hill. 2001.
- PUTNAM, L. H. **A General Empirical Solution to the Macro Software Sizing and Estimating Problem**. IEEE Transactions On Software Engineering, Vol. Se-4, No. 4, July 1978.
- PUTNAM, L. H.; MYERS, W. **Measures For Excellence**: Reliable Software On Time, Within Budget. Prentice Hall PTR. 1992.
- PUTNAM, L. H.; MYERS, W. **Five Core Metrics**: The Intelligence Behind Successful Software Management. Dorset House Publishing Company, Incorporated. 2003.
- QSM. **Function Point Programming Languages Table**.
<http://www.qsm.com/FPGearing.html>. Pesquisado em 12/11/2006.
- ROPPONEN, J.; LYYTINEN, K. **Components of Software Development Risk: How to address them? A Project Manager Survey**. IEEE Transactions On Software Engineering, Vol. 26, No. 2, 2000.
- ROYCE, W. **Software Project Management**: a unified framework. Reading, Mass.: Addison-Wesley. 1998.
- SALOMON, D. **Como fazer uma monografia**. Martins Fontes. 10a. Ed. 2001
- SEVERINO, A. J. **Metodologia do trabalho científico**. 22a. Ed. Cortez. 2002.
- SEI (SOFTWARE ENGINEERING INSTITUTE). **CMMI® for Development, Version 1.2**. Disponível em www.sei.cmu.edu/cmmi/ (em 23/02/2007). 2006.
- SILVEIRA, M.; AGUIAR, M.. **Backfiring: um Atalho ou uma Estrada Que Não Leva a Lugar Algum?** Pesquisado em 30 de junho de 2007 em:
http://www.bfpug.com.br/Artigos/Backfiring_Marcio_Mauricio.htm

STUTZKE, R. D. **Estimating Software-Intensive Systems**: Projects, Products, and Processes. Addison-Wesley Professional. 2005.

VAZQUEZ, C. E.; SIMÕES, G.; ALBERT, R.. **Análise de Pontos de Função**. Erica. 2003.

APÊNDICE A

SIMULAÇÃO DE ACOMPANHAMENTO DE UM PROJETO

A. SIMULAÇÃO DE ACOMPANHAMENTO DE UM PROJETO

A.1. Introdução

O presente Apêndice descreve os passos seguidos na criação dos indicadores para a simulação de acompanhamento de um projeto hipotético, para demonstração das idéias e da proposta deste trabalho. Essa simulação é feita através da utilização de planilha eletrônica e de suas facilidades para elaboração de gráficos. O Apêndice caracteriza o projeto e, em seguida, determina os parâmetros de produtividade e *manpower buildup*, para depois calcular os indicadores desenvolvidos no corpo do trabalho.

A.2. Descrição do projeto hipotético

O projeto e a organização a que ele se filia descrevem-se a partir das seguintes informações:

Tamanho do projeto: 1.000 pontos de função;

Processo de software utilizado: baseado no Processo Unificado;

Ambiente: J2EE;

Linguagem de programação: Java;

Gearing factor: 60 instruções por ponto de função;

Fatores PP e MBP: calculados a partir do histórico da organização, composto de 7 projetos desenvolvidos na mesma plataforma;

Avaliações periódicas do projeto: realizadas a cada duas semanas.

A.3. Determinação dos parâmetros organizacionais

Para utilização do modelo paramétrico de estimativa de Putnam, devem-se estabelecer médias para o parâmetro de produtividade da organização (PP) e parâmetro de alocação de mão-de-obra, (MBP – *Manpower Buildup Parameter*), utilizando-se o histórico de projetos da organização e as fórmulas [3-2] e [3-8].

As organizações que não tenham histórico de projeto podem calcular parâmetros provisórios, utilizando projetos obtidos de bancos de dados criados para efeito de *benchmark*, como, por exemplo, o do ISBSG (www.isbsg.org). Os projetos escolhidos do banco de dados, para servir de ponto de partida para o processo de estimativa, devem, tanto quanto possível, ser semelhantes ao projeto a ser desenvolvido em termos de:

- Plataforma tecnológica e linguagem de programação;
- Domínio da aplicação;
- Tempo de experiência da equipe de desenvolvimento;
- Velocidade que se pretende imprimir ao projeto;
- Tamanho dos sistemas a desenvolver.

À medida que os dados de projeto da própria organização estejam disponíveis, estes devem substituir gradualmente os dos projetos de bancos de dados tomados como ponto de partida. A atualização deve ser feita a cada novo projeto concluído.

O ideal é que, para a determinação dos parâmetros, se utilizem inúmeros projetos da base histórica para maior confiabilidade dos valores calculados. Contudo, três projetos da própria organização representam melhor o ambiente em que se dá o trabalho de desenvolvimento do que um número bem maior de projetos tomados de outras empresas.

A Figura A-1 ilustra a determinação dos parâmetros organizacionais PP e MBP e índices PI e MBI, que são usados para o cálculo de vários dos indicadores de projeto, na forma descrita nas estruturas de medição.

Os valores medidos – tamanho em pontos de função não ajustados, tamanho em linhas de código, esforço da construção (*Main Build*) em horas e duração da construção em meses – são introduzidos na área colorida da planilha que gera os valores médios para os parâmetros.

Os 7 projetos incluídos para o cálculo dos índices geraram valores médios de PP=25.572 e MBP=25,03. Esses valores geram os valores médios PI=15,54 e MBI=2,85, por interpolação na Tabela 3-6 e na Tabela 3-7, respectivamente.

Figura A-1 – Determinação dos parâmetros organizacionais

Determinação do parâmetro de produtividade da organização (PP) e do parâmetro de alocação de mão de obra (MBP), com base no histórico de projetos da organização

Referencia: PUTNAM, LAWRENCE H.; MYERS, WARE. Measures For Excellence: Reliable Software On Time, Within Budget. Prentice Hall PTR. 1991.

Fórmulas: $PP = \text{tamanho} / [(E_{DEV} / \beta)^{1/3} * t_d^{4/3}]$
 $MBP = E_{DEV} / (0,39 * t_d^3)$

onde:

tamanho é representado pelo número de linhas de código do produto (LOC);

E_{DEV} é o esforço em homens-ano utilizados nas fases de construção principal do produto.

β é fator de ajuste de tamanho.

t_d é a duração, em anos, do início da fase de projeto até a conclusão dos testes integrados.

PP é um parâmetro que caracteriza a produtividade.

MBP é o parâmetro de alocação de mão de obra que indica a velocidade do desenvolvimento.

Ambiente: J2EE/Java Gearing Factor: 60

Histórico de projetos

								PP	PI	MBP	MBI	
								Média	25572	15,54	25,03	2,85
								Desvio Padrão	6577		4,29	
Projeto	Pontos de função não ajustados	Tamanho (linhas de código)	Esforço Main Build (horas)	Duração Main Build (meses)	E _{DEV} (homem-ano)	t _d (anos)	β	PP	PI	MBP	MBI	
P1	1000	54053	13127	10,36	6,84	0,86	0,37	24857	15,44	27,21	3,01	
P2	500	31875	5620	8,75	2,93	0,73	0,28	22220	15,05	19,38	2,38	
P3	600	36332	7674	8,48	4,00	0,71	0,28	23782	15,28	29,01	3,07	
P4	650	37160	7091	9,08	3,69	0,76	0,28	22800	15,13	21,84	2,59	
P5	250	13821	2400	5,88	1,25	0,49	0,16	18050	14,08	27,31	3,01	
P6	1602	94437	19565	12,99	10,19	1,08	0,39	28647	16,00	20,62	2,49	
P7	1690	108661	19696	11,51	10,26	0,96	0,39	38645	17,29	29,84	3,10	

<-----Fim da área de entrada de dados. Insira linhas acima deste ponto ----->

A.4. Formulário de entrada de dados

Para o cálculo dos indicadores, é necessário prover uma série de medidas primitivas que reflitam o progresso do desenvolvimento do sistema. As medidas usadas no cálculo dos indicadores na presente simulação constam do formulário da Figura A-2.

Figura A-2 Formulário de entrada de dados

Quinzena	Data (dd-mm-aa)	Tipo de contagem	Adição (PF)	Eliminação (PF)	Linhas de código desenvolv. & testadas	Esforço real da quinzena anterior (Horas)	Casos de teste especificados	Casos de teste passados
0	02/01/06	Contagem inicial	1000		0	0	200	0
1	16/01/06	Mudança requisitos	30	-5	1483	332	213	0
2	30/01/06	Acerto em milestone	20		3023	349	226	0
3	13/02/06	Mudança requisitos	10	-10	4126	357	235	0
4	27/02/06				5591	366	247	10
5	13/03/06				6964	376	259	25
6	27/03/06	Mudança requisitos	45		8939	426	276	35
7	10/04/06				10447	437	289	45
8	24/04/06				12154	467	298	45
9	08/05/06	Mudança requisitos		-30	13236	462	303	55
10	22/05/06				15268	513	306	85
11	05/06/06				16995	524	310	110
12	19/06/06				19136	535	316	120
13	03/07/06	Acerto em milestone	40		21761	649	321	140
14	17/07/06				23589	719	325	150
15	31/07/06							
16	14/08/06							
17	28/08/06							
18	11/09/06							
19	25/09/06							
20	09/10/06							
21	23/10/06							
22	06/11/06							
23	20/11/06							
24	04/12/06							

No formulário definem-se os seguintes campos:

Quinzena: Identificação seqüencial da quinzena decorrida desde o início do desenvolvimento do projeto. Cada quinzena coincide com uma revisão do projeto. A quinzena identificada como 0 coincide com o início da fase de construção.

Data: Data de cada revisão de projeto.

Tipo de contagem: Menu *drop-down* que indica o tipo de revisão de requisitos e contagem de funcionalidade efetuada. São valores válidos:

- “Contagem inicial”: se estabelece no campo Adição o valor obtido em pontos de função;
- “Acerto em *milestone*”: se estabelece no campo Adição o valor a ser somado à funcionalidade anterior ou dela subtraído (com sinal negativo), resultante de recontagem na qual não houve alteração de requisitos;

- “Mudança requisitos”: coloca-se, no campo Adição, o valor da funcionalidade adicionada em função da mudança, e no campo Eliminação, o valor da funcionalidade eliminada (com sinal negativo);

Adição: Os usos desse campo se descrevem acima em “Tipo de contagem”.

Eliminação: Os usos desse campo se descrevem acima em “Tipo de contagem”.

Linhas de código desenvolvidas e testadas: Preencher nesse campo o total de linhas de código lógicas dos programas produzidos, testados e colocados sob controle de configuração. Esse valor pode ser obtido através de contagem automatizada por programas criados para esse fim.

Esforço real da quinzena anterior: Reporta-se nesse campo o total de horas trabalhadas pela equipe do projeto. Excluem-se desse valor a quantidade de horas gastas com retrabalho resultante de mudanças de requisitos. Essas horas de retrabalho também devem ser registradas, mas à parte, para efeito de ajustes ao projeto, inclusive eventuais renegociações. A exclusão do retrabalho oriundo de mudanças de requisitos é feita para facilitar o cálculo dos indicadores que se relacionam com a funcionalidade instalada (conforme descrita na seção 4.5.1).

Casos de testes especificados: Quantidade existente de casos de testes na data da revisão. No começo da fase de construção, boa parte dos casos de teste já podem ter sido criados nas fases anteriores.

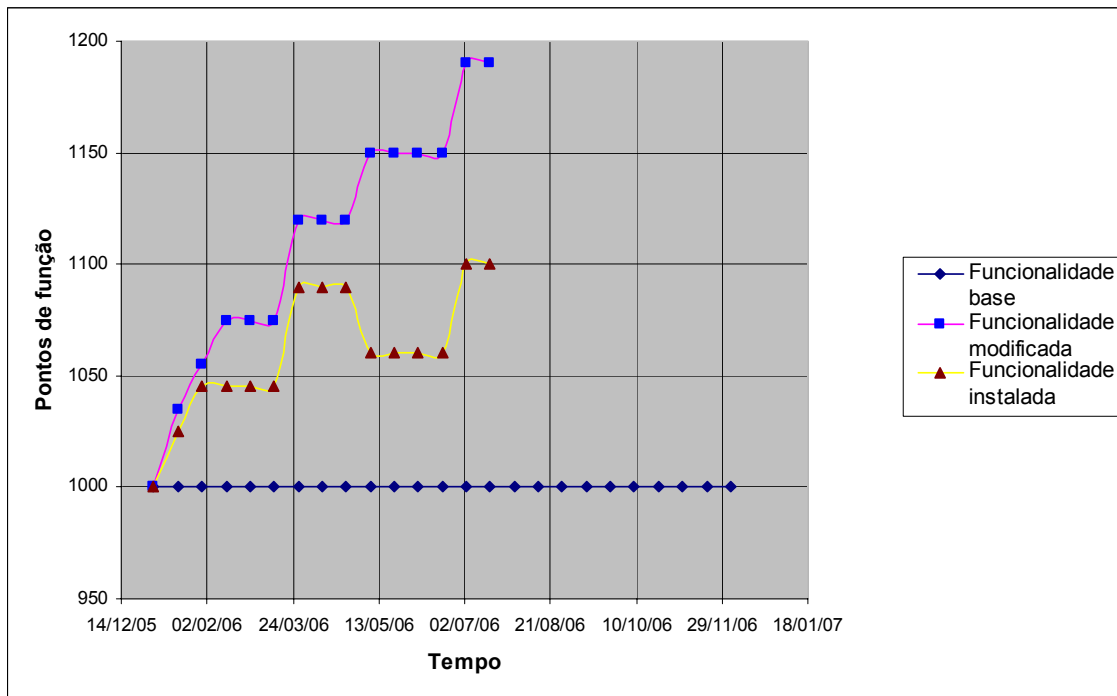
Casos de testes passados: Quantidade existente de casos de testes bem sucedidos na data da revisão.

A.5. Indicador de variação de funcionalidade (I-1)

O projeto deve acompanhar a evolução do tamanho funcional modificado do sistema (definido na seção 4.5.1, e que inclui alteração e exclusão de funcionalidade que acarretam retrabalho) e do tamanho instalado.

A Figura A-3 ilustra uma possível implementação do indicador com base em sua definição através da Tabela 4-2.

Figura A-3 – Indicador de variação de funcionalidade (I-1)



Na figura nota-se que o sistema, ao início da fase de construção, estava estimado com um tamanho de 1.000 pontos de função. Ao longo do desenvolvimento, mudanças de requisitos e recontagens, anotadas no formulário de entrada de dados, alteraram tanto a funcionalidade instalada, quanto a modificada.

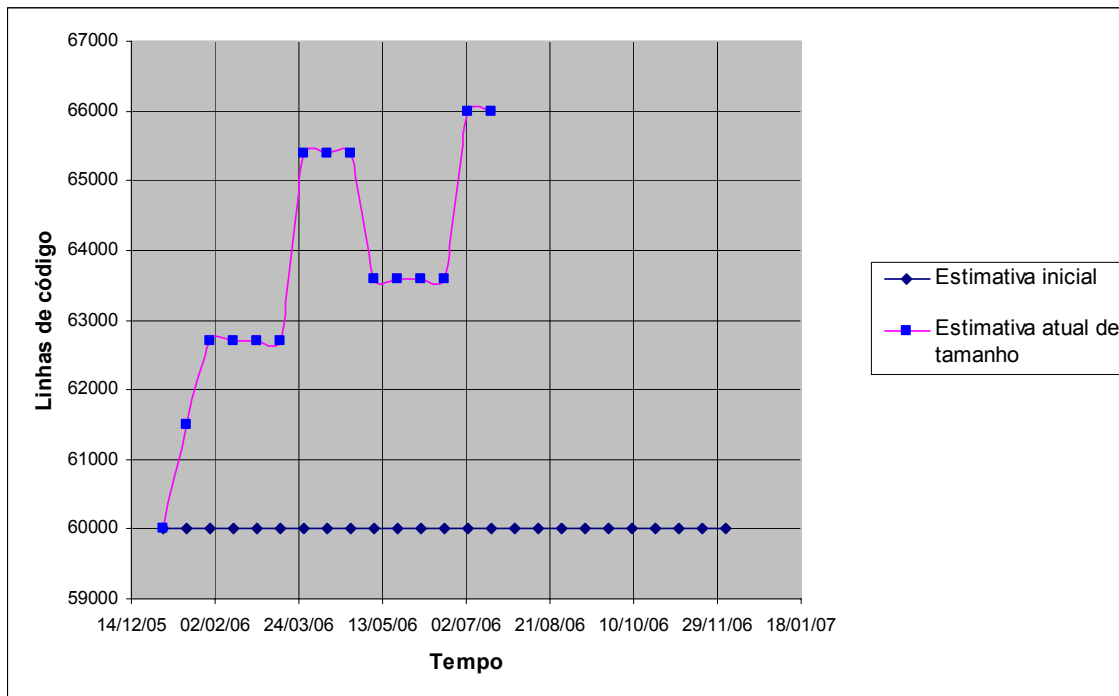
A.6. Indicador de variação de tamanho de produto (I-2)

O projeto deve acompanhar a evolução do tamanho do produto, que é resultante do tamanho funcional instalado do sistema. O tamanho do produto é definido em linhas de código lógicas, excluindo comentários. É calculado segundo descrito na seção 3.2.4.

A Figura A-4 ilustra uma possível implementação do indicador com base em sua definição através da Tabela 4-4.

Na figura, nota-se que o tamanho do produto variou 10% ao longo dos seis meses de projeto, como resultado das alterações de requisitos e recontagens indicadas no formulário de entrada de dados.

Figura A-4 – Indicador de variação de tamanho de produto (I-2)



A.7. Indicador de atraso de projeto (I-3)

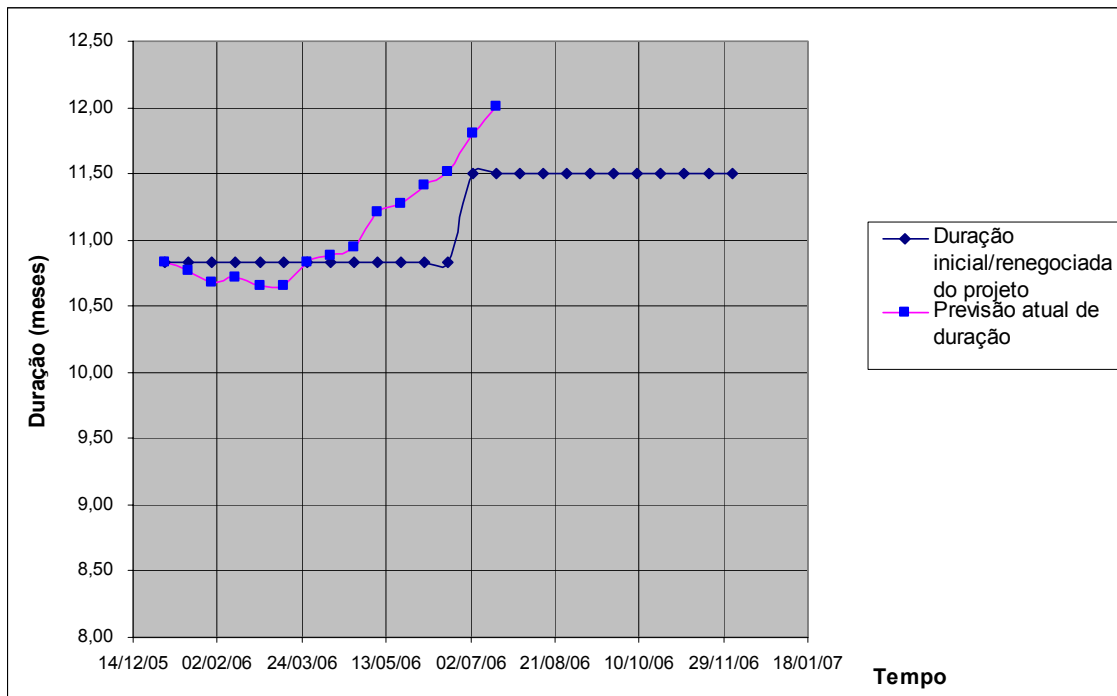
O projeto deve acompanhar, a cada revisão, a duração esperada do projeto comparando-a com a duração inicial estimada ou, ainda, duração renegociada em função das variações de requisitos.

A Figura A-5 ilustra uma possível implementação do indicador com base em sua definição através da Tabela 4-6.

Na figura, nota-se que, inicialmente, as novas previsões de duração do projeto feitas nas revisões apontam para uma redução da duração do projeto. Como se vê dos indicadores que se avaliam a seguir, isso se deve à alocação mais intensa de recursos no início do projeto (Figura A-7) e da resultante geração de linhas de código mais rápida no início (Figura A-10).

Com a variação dos requisitos e a redução do ritmo de geração de linhas de código, as previsões de duração do projeto passam a crescer. O gráfico indica ter havido renegociação da duração do projeto no início do mês de julho. Contudo, as previsões ainda indicam que o projeto está com problemas para concluir dentro da duração renegociada.

Figura A-5 – Indicador de atraso de projeto (I-3)



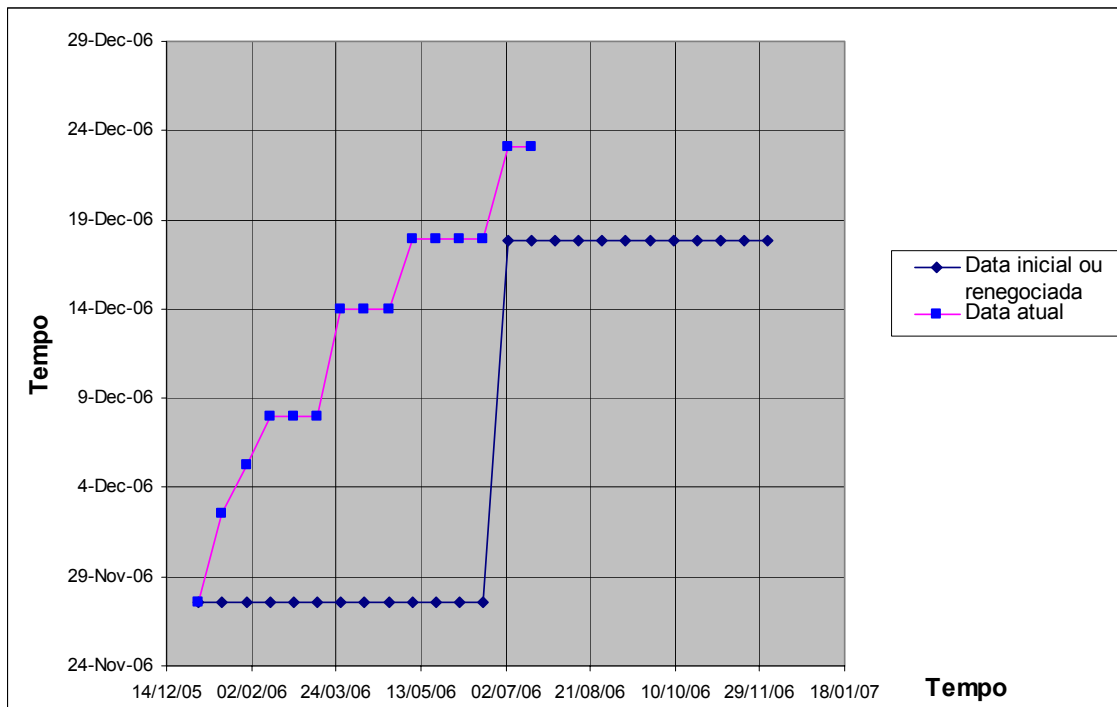
A.8. Indicador de evolução de re-estimativas (I-4)

A cada revisão, o gerente do projeto deve calcular a previsão de conclusão com base nos requisitos do projeto. À medida que se alteram os requisitos, quer por adição, alteração ou eliminação, uma nova data deve ser calculada com base na funcionalidade modificada.

A Figura A-6 ilustra uma possível implementação do indicador com base em sua definição através da Tabela 4-8.

Na figura, nota-se que, à medida que novos requisitos se vão introduzindo, novas estimativas da data de conclusão são feitas, que resultam posterior à data inicial. Com a renegociação da data de conclusão do projeto, a data esperada de conclusão volta a aproximar-se da data comprometida.

Figura A-6 – Indicador de evolução de re-estimativas (I-4)



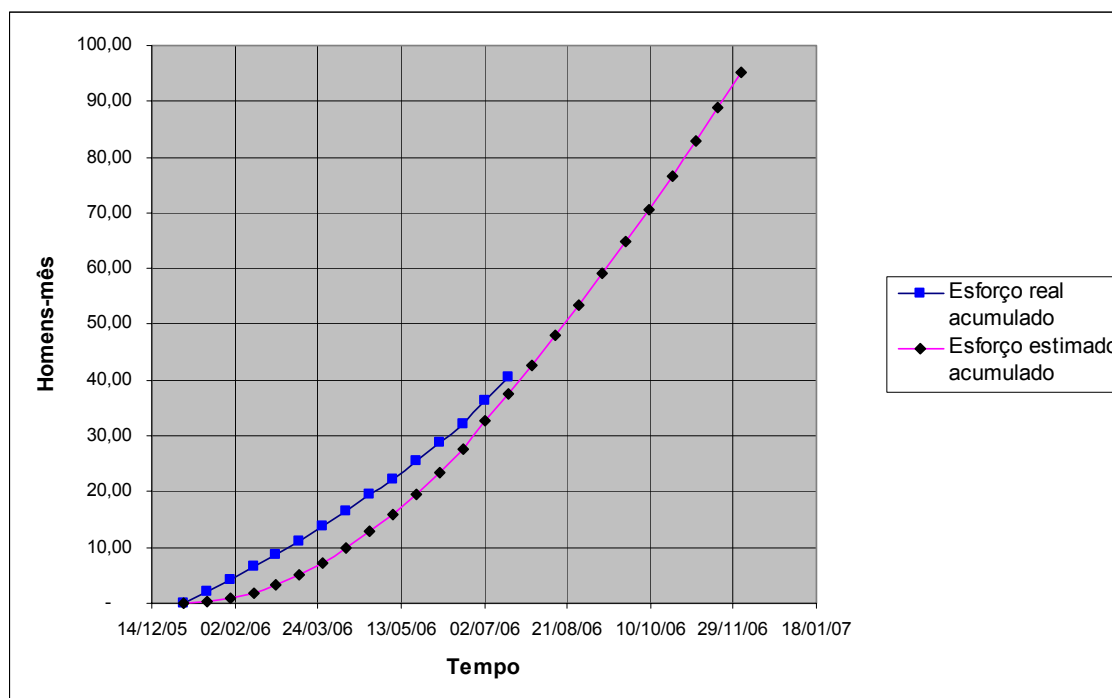
A.9. Comparação entre esforço estimado e despendido (I-5)

O projeto deve acompanhar, a cada revisão, o consumo acumulado de horas de desenvolvimento, comparando-o com a estimativa de esforço.

A Figura A-7 ilustra uma possível implementação do indicador com base em sua definição através da Tabela 4-10.

Na figura, nota-se que, inicialmente, o consumo de recursos excede as estimativas iniciais, indicando maior gasto de horas do projeto. Isso se deve à alocação mais rápida e intensa de recursos no início do projeto, como se verifica na Figura A-9. O consumo real de horas tende a aproximar-se do consumo estimado ao redor do mês de julho.

Figura A-7 – Indicador de comparação entre esforço estimado e despendido (I-5)



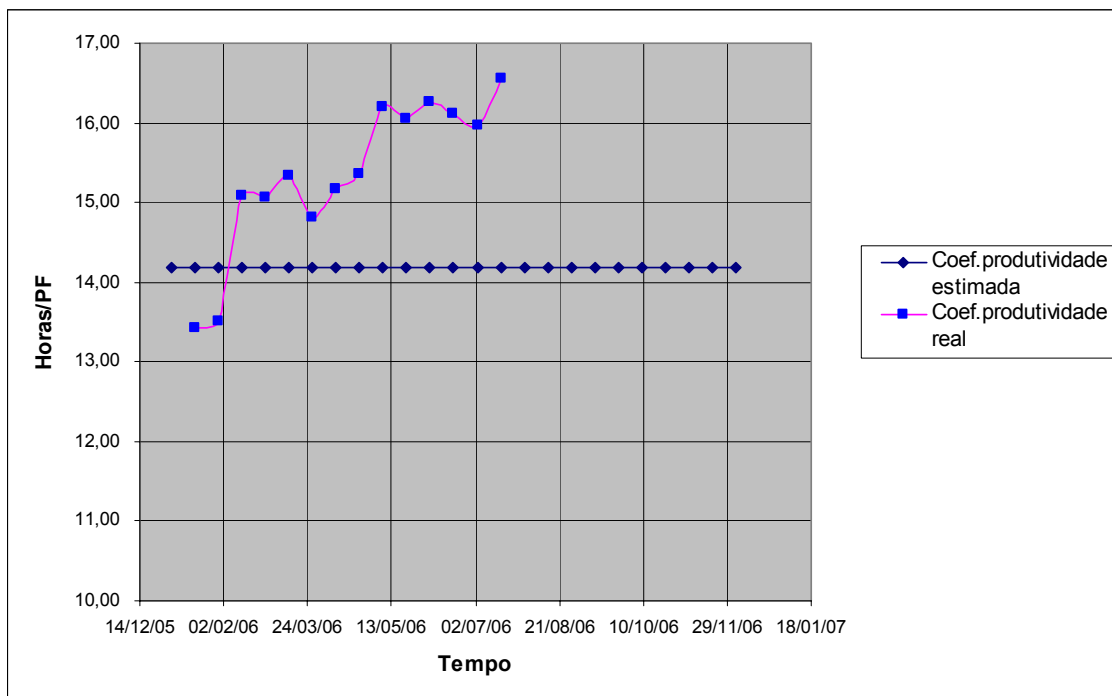
A.10. Indicador de acompanhamento de produtividade (I-6)

O projeto deve acompanhar, a cada revisão, a produtividade que vem sendo conseguida pelo projeto. Para tanto, utiliza-se o coeficiente de produtividade, expresso em horas por ponto de função. O coeficiente de produtividade alcançado em cada revisão resulta da quantidade de linhas de código produzida até o momento e do esforço utilizado.

A Figura A-8 ilustra uma possível implementação do indicador com base em sua definição através da Tabela 4-12.

Na figura, nota-se que, nos primeiros meses do projeto, a produtividade está abaixo da esperada. Isso se deve à maior utilização de mão-de-obra nos primeiros meses do projeto, sem que tenha havido produção adicional suficiente de código, correspondente ao aumento maior de mão-de-obra. Essa explicação é consistente com a Figura A-9.

Figura A-8 – Indicador de acompanhamento de produtividade (I-6)



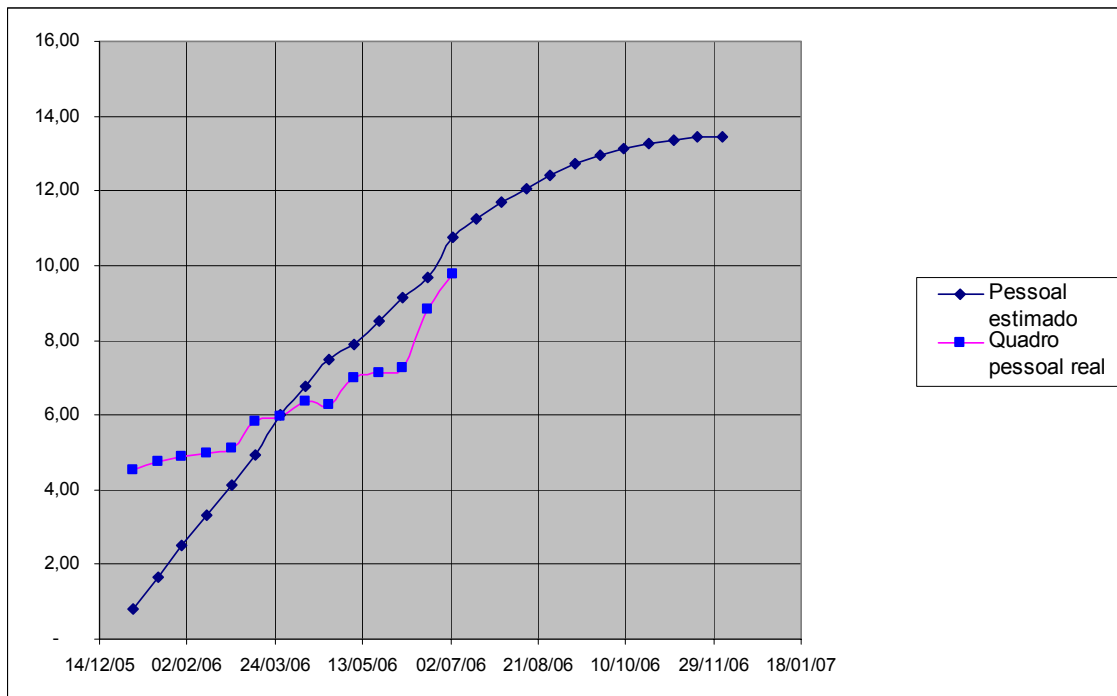
A.11. Indicador de alocação de pessoal (I-7)

O projeto deve acompanhar, a cada revisão, como a alocação real de pessoal ao projeto se compara com a alocação prevista no início, ou reprogramada em função das variações de requisitos.

A Figura A-9 ilustra uma possível implementação do indicador com base em sua definição através da Tabela 4-14.

Na figura, nota-se que, inicialmente, o pessoal alocado ao projeto foi superior à necessidade prevista pelo modelo de estimativa de recursos adotado. A partir do terceiro mês do projeto, ao contrário, o projeto passou a contar com menos recursos do que o previsto. O gráfico indica que se requer ação do gerente do projeto para adicionar o número adequado de profissionais ao projeto o mais cedo possível.

Figura A-9 – Indicador de alocação de pessoal (I-7)



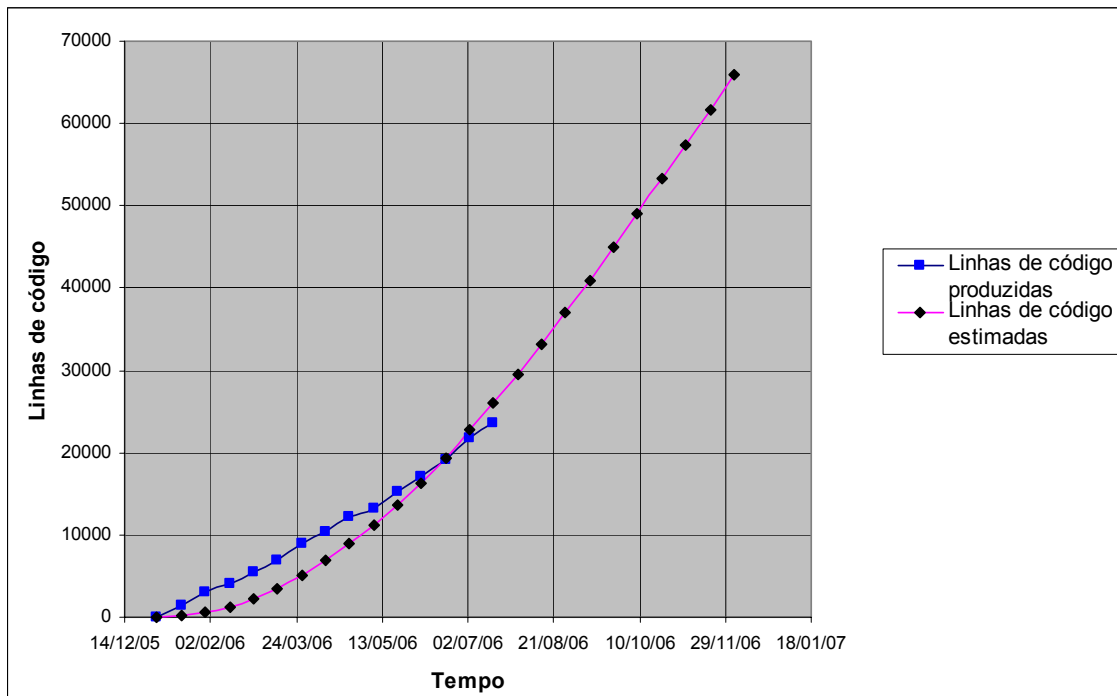
A.12. Indicador de produção de linhas de código (I-8)

O projeto deve comparar, a cada revisão, a quantidade de linhas lógicas de código geradas, testadas e colocadas sob controle de configuração até o momento, com a produção planejada no início do projeto ou em sua reprogramação. A produção de linhas de código deve se dar de forma proporcional ao consumo do esforço segundo a distribuição de Rayleigh. A determinação da quantidade de linhas de código geradas deve ser feita ignorando-se linhas de comentário.

A Figura A-10 ilustra uma possível implementação do indicador com base em sua definição através da Tabela 4-16.

Na figura, nota-se que, inicialmente, o projeto gerou mais linhas de código do que as previstas. Contudo, ao redor da metade do projeto, a produção efetiva igualou-se com a prevista. A inclinação da curva de produção real indica a tendência de que o projeto passe a sofrer atraso na geração dos artefatos. Essa tendência deve ser examinada em conjunto com o déficit de pessoal na equipe de projeto, evidenciado pela Figura A-9.

Figura A-10 – Indicador de produção de linhas de código (I-8)



A.13. Indicador de complexidade do projeto (I-9)

O gerente do projeto deve, a cada revisão, cuidar de quantificar os casos de teste criados e comparar seu número com o de casos de testes executados com sucesso.

A Figura A-11 ilustra uma possível implementação do indicador com base em sua definição através da Tabela 4-18.

Na figura, nota-se que, por volta da metade do projeto, a geração de novos casos de testes diminuiu, enquanto aumenta a velocidade de execução bem sucedida de casos de testes. As duas curvas apresentam tendência a convergir no futuro. O uso desse indicador é mais significativo à medida que se avizinha a conclusão do desenvolvimento.

Figura A-11 – Indicador de complexidade do projeto (I-9)

