



Programa de Pós-Graduação em Engenharia de Software do C.E.S.A.R

**MODELAGEM DE RISCO PARA CÁLCULO DE RESERVA
DE CONTIGÊNCIA E RESERVA GERENCIAL EM
PROJETOS DE SOFTWARE.**

Karline de Valésio Pinto Filho

Recife,
18 de agosto de 2011



Karline de Valésio Pinto Filho

**MODELAGEM DE RISCO PARA CÁLCULO DE RESERVA
DE CONTIGÊNCIA E RESERVA GERENCIAL EM
PROJETOS DE SOFTWARE.**

*ESTE TRABALHO SERÁ APRESENTADO
À PÓS-GRADUAÇÃO EM ENGENHARIA
DE SOFTWARE DO C.E.S.A.R - CENTRO
DE ESTUDOS E SISTEMAS AVANÇADOS
DO RECIFE COMO REQUISITO PARCIAL
PARA OBTENÇÃO DO GRAU DE MES-
TRE.*

Orientador: Jones Albuquerque

Coorientador: Silvana Bocanegra

Recife,

18 de agosto de 2011

Ao meu Pai (in memoriam).

Agradecimentos

- Ao meu orientador, professor Jones Albuquerque, pela competência, bom senso e presteza demonstrados nesses dois semestres em que trabalhamos juntos.
- À minha coorientadora, professora Silvana Bocanegra, pela ternura, apoio e suporte em modelagem matemática.
- Ao professor Sérgio Cavalcante, aos profissionais de T.I Antônio Valença e Gustavo Bastos, pela oportunidade de vivenciar na prática o estudo de caso.
- À analista de sistemas Marcela Guerra, pela presteza no fornecimento das medidas do projeto.
- À professora Herlen Kúrie pelo apoio e incentivo.
- À minha mãe (ex-professora regente do Velho Ginásio Pernambucano) e pai (ex-professor adjunto dos departamentos de Antibiótico e Micologia da UFPE , in memoriam).

Resumo

Esse trabalho apresenta um estudo realizado numa fabrica de software na cidade do Recife. A partir da coleta de dados reais foi realizado uma abstração do modelo de casos de uso, a partir do qual, objetos possam ser derivados. Foram apresentadas especificações para essa derivação em linguagem de notação algébrica para ser inserido no ambiente de otimização matemática para modelagem de riscos em reserva de contingência e gerencial. O objetivo do modelo de simulação é encontrar os parâmetros do processo de desenvolvimento para o projeto de software que resultam, em um custo mínimo, prazo, esforços e riscos. O modelo inclui uma interface de simulação onde é possível configurar os parâmetros manualmente, com uma automática otimização do experimento. Após a simulação o projetista pode constatar a necessidade de se refazer o modelo, ajustar parâmetros, recalculando as possibilidades a cada nova saída e produzindo gradualmente estimativas mais acuradas. O modelo de simulação foi definido no ambiente de modelagem matemática - AIMMS.

Palavras-Chave:

Modelagem Matemática, Otimização Matemática, Programação Linear, Riscos em Projetos de Software, Riscos de Reserva Contingencial e Casos de Uso.

Abstract

This paper presents a case study of the software factory in Recife. From the actual data collection was carried out an abstraction of the model use cases, from which objects can be derived. Specifications were presented this derivation in the language of algebraic notation to be inserted into a mathematical optimization environment for modeling for risk reserve contingency and risk contingency reserve. The goal of the simulation model is to find the development process parameters for the software project that result in minimum cost, time, effort and risks. The model includes a simulation interface where designer can set up the parameters manually, and the automatic optimization experiment. After the simulation the designer can see the need to redo the model, adjust parameters, recalculating the possibilities each new output, and gradually producing more accurate estimates. The mathematical simulation model was specified in AIMMS environment.

Keywords:

Mathematical Modeling, Optimization, Linear Programming, Risks in Software Projects, Risk Management Reserve and Use Cases.

Sumário

1	Introdução	1
1.1	Por quê as reservas contingencial e gerencial são importantes para a gerência de projetos?	4
1.2	Motivação	6
1.3	Objetivos	7
1.4	Limites do Estudo	7
1.5	Organização do Estudo	8
2	Revisão Sistemática	9
2.1	Metodologia da Revisão	15
3	O Modelo Proposto	18
3.1	A Medida de esforço	18
3.2	Registro da medida de esforço	20

- 3.3 Classificação dos membros do time 21
- 3.4 Modelagem ADM sistemas 21
 - 3.4.1 Visão Hierárquica 22
 - 3.4.2 Visão Seqüencial 23
 - 3.4.3 Visão Desenvolvimento 23
 - 3.4.4 Casos de Uso 23
 - 3.4.5 Especificação dos Objetos e Métodos 25

- 4 A SIMULAÇÃO DO MODELO PROPOSTO 27**
- 4.1 Descrição dos dados para a Linguagem de Notação Algébrica 27
- 4.2 Notation 35
 - 4.2.1 Stochastic Integer Linear Programming Formulation 35
- 4.3 System-level Design Exploration and Risk Analysis 41
- 4.4 Bibliografia 43

Lista de Tabelas

4.1	36
4.2	37
4.3	37

Lista de Figuras

Capítulo 1

Introdução

A melhoria de processo de software significa primeiramente compreender os processos existentes para então modificá-los, não significando simplesmente a adoção de métodos padronizados, modelos ou ferramentas. A melhoria de processo deve ser vista como uma atividade específica para cada organização necessitando, portanto, de um período gradual de implantação e adaptação. (SOMMERVILLE, 2003). O crescimento por produtos de software é cada vez maiores, em 2010 o Brasil terminou o ano numa situação de destaque, alcançando a 11^a posição no ranking mundial, tendo movimentado 19,04 bilhões de dólares, equivalente a 1,0No entanto, o cenário típico que geralmente se observa nas organizações produtoras de software mostra-nos que tais técnicas, quando existentes, não são usadas adequadamente em grande parte dos casos. Estimativas equivocadas de prazos e custos, ou produtos que apresentam baixa qualidade, infelizmente constituem situações comuns no desenvolvimento de software. Grande parte dos projetos e produtos é desenvolvida e evoluída em bases na experiência do gerente de projetos, cuja grande preocupação é atingir o objetivo de prazo, negligenciando-se os atributos de qualidade dos produtos. E freqüentemente nem o objetivo de prazo é atingido, resultando em atrasos,

orçamentos ultrapassados e insatisfação geral. Além disso, experiências passadas não são bem aproveitadas, é comum uma organização cometer o mesmo erro diversas vezes. Em uma fábrica de software, o aproveitamento da experiência obtida ao longo do desenvolvimento de cada produto é de extrema importância, na medida em que vai progressivamente compondo uma base de conhecimento que contribuirá para um melhor desempenho em projetos futuros. As lições aprendidas permitem que os processos sejam continuamente ajustados, tornando-se progressivamente mais adequados, estáveis e previsíveis. Reyck (2005) comenta que entregar um projeto no prazo previsto, dentro do orçamento e com a satisfação do cliente, continua sendo uma notória dificuldade. O autor aponta como a causa principal para estas falhas o fato de que todos os projetos estão sujeitos a riscos e saber como tratá-los é hoje o fator principal para o seu sucesso. Algumas definições conceituais de projeto, apresentadas a seguir, ajudam a esclarecer esse ponto. O risco pode ser definido como a combinação da probabilidade de um acontecimento e das suas consequências (ISO/IEC Guide 73). O simples fato de existir atividade, abre a possibilidade de ocorrência de eventos ou situações cujas as consequências constituem oportunidades para obter vantagens no que tange aos aspectos positivos ou negativos no caso de ameaças ao sucesso. As empresas em sua grande maioria, diariamente fazem escolhas, com graus diferenciados de riscos, mas também com um alto grau de oportunidade e benefícios associados. Uma ótima ilustração do que foi dito, encontra-se na matéria: "imbróglio: TRT aumenta o risco para empresas de TICs", blog dia dia, bit a bit do cientista Sílvio Meira em 2009, que afirmou: agrega renda e valor na Índia, onde se criou uma economia de serviços altamente especializada que, juntamente com a de desenvolvimento e manutenção de software, soma US 47 bilhões à economia do país e emprega dois milhões de pessoas. Muitas milhares das quais [principalmente em software] trabalham para empresas brasileiras: no Brasil, terceirizar para o Brasil é um grande risco; para a Índia, não. isto é incrível."Quais os principais benefícios do gerenciamento de riscos quando se

usa a abordagem ágil?*** Na matéria: "Gerenciamento de Riscos em Projetos Ágeis", Cláudio Romero Lucena Rocha, faz a seguinte análise: "... o que acontece em muitos projetos ágeis é o gerenciamento implícito de riscos. Para eles, o custo do gerenciamento explícito de riscos se sobrepõe aos benefícios. Mas, apesar de serem muitos os projetos que podem (e devem) tratar os riscos de tal maneira, outros se beneficiam de gerenciá-los explicitamente. Para isso, John Brothers introduziu uma técnica em 2004 chamada de Risk Burndown Chart, ilustrado na figura abaixo:

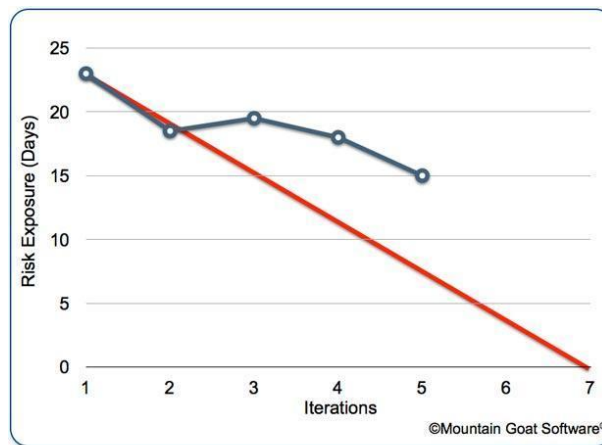


Figura 1. Risk Burndown Chart. A linha vermelha é o parâmetro base da redução do risco ao longo das iterações, enquanto que a azul é a atualização da situação real.

Basicamente, o gráfico mostra a exposição do projeto ao risco ao longo das iterações. A Exposição ao Risco é calculada considerando o tamanho do impacto (em dias) do risco e a sua probabilidade de acontecer. Como exemplo, um risco do atraso do projeto em 15 dias, com 20 Olhando para o gráfico acima, vemos que o risco não tem diminuído a uma taxa apropriada (nem perto disso). Quando isso ocorre, o time pode requisitar dedicar algum tempo no próximo Sprint para trabalhar diretamente com mitigação de riscos. Como podemos ver então, é possível sim levar a sério o gerenciamento de riscos em projetos ágeis a ponto de explicitá-lo. Se for necessário, o Risk Burndown Chart é uma útil ferramenta. Por fim, é bom não confundir "não levar o gerenciamento explícito a

sério" com "não levar o risco a sério". Vimos neste texto que o risco sempre é tratado em projetos ágeis, seja explícita ou implicitamente". (BARBOSA et al., 2008). Afirma que a precisão das estimativas dos custos dos pacotes de trabalho irá influenciar diretamente no orçamento final e nas análises de riscos e de reservas do projeto. Um pacote de trabalho é uma descrição quantitativa e qualitativa de uma operação a ser executada no projeto. Considerando o modelo desenvolvido neste estudo, a mesma é inserida num contexto que envolve simulações. A utilização do processo automatizado viabiliza um grande número de análises, contribuindo para minimizar os riscos de projetos no processo de avaliação do risco.

1.1 Por quê as reservas contingencial e gerencial são importantes para a gerência de projetos?

Em gerência de projetos, muitas decisões deverão ser tomadas, e nem sempre estas são fáceis. A informação reveste-se de matéria prima fundamental no qual reside subsídios de como trataremos ou reagiremos aos riscos e incertezas do projeto. Após a identificação de um risco, o gerente de projeto pode encontrar vários caminhos que sirvam de pista, para responder de forma satisfatória aos impactos do risco. Mas, qual o melhor caminho? Em resposta a esta pergunta, fica claro que a tomada de decisão do gerente de projetos se encaixa dentro da análise de riscos. Em projetos nem sempre podemos confiar o resultado apenas em sentimentos ou experiências, a decisão deverá partir de parâmetros que possa embasar a análise e interpretação, garantindo uma maior probabilidade de sucesso. Eventualmente, ocorrerão eventos de riscos que são totalmente desconhecidos. Como poderemos lidar com estes imprevistos? Além dos riscos conhecidos, é usual incluir uma reserva de contingência distribuída ao longo do projeto para eventos desse tipo -

Riscos desconhecidos. Vargas (2003, p. 104) descreve: "[...] as reservas serão consumidas com base nas solicitações de mudanças provenientes dos outros planos e dentro da autonomia do gerente do projeto". No contexto desta pesquisa, o processo de modelagem é usado para auxiliar o gerente de projeto no processo decisório dentro de sua autonomia. [ARENALES 2007] define que a formulação (modelagem) são as variáveis e as relações matemáticas para descrever o comportamento do projeto de software ou problema do mundo real. A dedução (análise) aplica técnicas matemáticas e tecnologias para resolver o modelo e visualizar as devidas conclusões. A interpretação (inferência) argumenta que as conclusões retiradas do modelo tem significado suficiente para inferir conclusões ou decisões para o problema do mundo real. Frequentemente, uma avaliação (julgamento) dessas conclusões ou decisões inferidas mostra que elas não são adequadas e que a definição do problema e sua modelagem matemática precisam de revisão e, portanto, o ciclo é repetido. A Figura 2 ilustra um processo simplificado da abordagem de solução de um problema usando modelagem matemática.

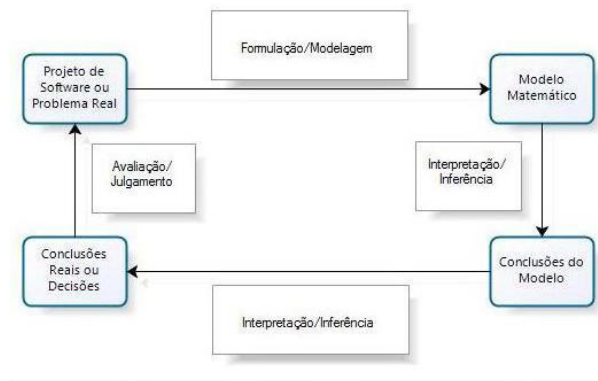


Figura 2: Processo de Modelagem adaptado de [ARENALES 2007].

1.2 Motivação

Ainda são poucos os estudos específicos no que tange à gestão de riscos em projetos de software. Machado (2002) afirma que o no "Brasil, muito pouco se sabe sobre o desempenho dos projetos de software e sobre os fatores de riscos que interferem no atendimento ao prazo dos projetos, pois não existem muitos estudos disponíveis sobre esse tema". O desafio desta pesquisa é aplicar o método descrito no capítulo 4 para o modelo de otimização matemática, quantificando riscos de reservas de contingência e gerencial. Segundo Barki [BARKI et al., 1993], "...a quantificação é um dos passos mais importantes no processo de avaliação de risco...", contribuindo para um entendimento homogêneo e menos subjetivo entre todos os envolvidos com os riscos do projeto. Evidenciamos um pequeno número de estudos mencionados na literatura: conforme pode ser evidenciado na revisão da literatura, conforme exposto no capítulo 2. Portanto, o aprofundamento da pesquisa foi realizada com base em projeto real de uma fábrica de software, resultado do modelo, descrito no capítulo 4. O modelo proposto de otimização é inédito no Brasil, não existindo estudos que tratem especificamente sobre esse tema.

O resultado dessa pesquisa evidencia a importância do processo de modelagem matemática, sobre o tema em tela, fornecendo informações valiosas para gerentes e equipes de projeto, fábrica de software, clientes e pesquisadores. No caso específicos dos gerentes de projetos, esse estudo irá possibilitar o uso desse método durante o processo de desenvolvimento do software, comprovando se as ações executadas para quantificações dos riscos têm o efeito esperado sobre o projeto. Outra dificuldade seria a escolha de uma arquitetura de processos que seja difundida o suficiente para que o modelo possa ser potencialmente aplicável a um grande grupo de organizações. Uma arquitetura que divida o processo em uma estrutura matricial, é sugerido e destacado.

1.3 Objetivos

O trabalho do modelo de simulação proposto tem como objetivos em nível: Geral:

"Disponibilizar um modelo de simulação voltado para otimização dos parâmetros dos processos de desenvolvimento de software para cálculo de riscos em reservas de contingência e gerencial em projetos de software.

Específico:

"Fazer uma revisão sistemática da literatura em relação a riscos em reservas de contingência e gerencial; "Desenvolver um modelo a partir da análise dos principais artefatos de um projeto de software real, com base na arquitetura matricial de uma fábrica de software, envolvendo riscos de reservas de contingência e gerencial; "Escrever a especificação da pesquisa em linguagem de modelagem algébrica (AML); "Utilizar a ferramenta de modelagem matemática AIMMES; "Proceder a realização do processo de otimização matemática do modelo proposto; "Apresentar os resultados do modelo de otimização e sugestão para estudos futuros.

1.4 Limites do Estudo

Tão importante quanto enumerar os objetivos do trabalho é esclarecer os limites do mesmo, visando delimitar mais precisamente o escopo que está sendo pesquisado. É importante ressaltar que a pesquisa propõe um modelo que simule aspectos do desenvolvimento de software, envolvendo riscos de reserva de contingência e gerencial, inerentes a projeto de software. Foi realizada uma coleta básica das medidas do projeto ADM sistemas que permitiu extrair informações relevantes para um melhor conhecimento dos

dados para a modelagem.

1.5 Organização do Estudo

A pesquisa seguiu uma seqüência de atividades, que foi utilizada como estruturação ao longo deste documento. A introdução discorre sobre o porque dos riscos de reservas, a inspiração da motivação, os objetivos, limites e organização da pesquisa. O Capítulo 2, consiste na revisão sistemática da literatura com o intuito de encontrar e analisar o maior número de trabalhos primários relevantes e reconhecidos na área de riscos de reservas de contingência e gerencial, de acordo com o guia PMBOK. A descrição complementar encontra-se nos apêndices D, E, F e G,. O Capítulo 3, descreve a aplicação do estudo em projeto real por uma fábrica de software, os passos que foram utilizados para guiar essa fase até a formalização das especificações para a linguagem de notação algébrica. O modelo de otimização matemática foi construído no Capítulo 4. Finalmente, o Capítulo 5, apresenta a análise dos resultados do modelo de simulação proposto, assim como, as conclusões observadas ao longo da pesquisa, além das sugestão de estudo futuro que podem ser desenvolvidos com base nos resultados aqui obtidos. O documento vem acompanhado ainda de seis apêndices, com informações que complementam alguns aspectos apontados ao longo do texto. Apresenta o referencial teórico que embasa as questões de estudo, referentes aos conceitos relacionados às seguintes áreas: pesquisa operacional especificamente no tocante a conceitos de programação linear; apresentação do AIMMS como ferramenta de modelagem matemática; uma breve apresentação da arquitetura matricial - PRAXIS e riscos de reservas de contingência e gerencial, de acordo com o guia PMBOK.

Capítulo 2

Revisão Sistemática

Uma revisão sistemática da literatura foi conduzida no intuito de encontrar e analisar o maior número de trabalhos primários relevantes e reconhecidos na área que possa responder as questões de estudo.

Questões do Estudo

A estudo parte para quatro questões de investigação mais específicas que possam responder essas perguntas na busca por uma abordagem que apóiem com práticas e ferramentas eficazes no gerenciamento de projetos.

(Q1) Quais os principais desafios no gerenciamento de reservas de contingência e gerencial em riscos de gerencia de projetos?

(Q2) Quais as melhores práticas a serem adotadas no gerenciamento de projetos de reservas de contingência e gerencial?

(Q3) Existem técnicas de otimização matemática que apóiem as atividades de geren-

ciamento de projetos de reservas de contingência e gerencial?

(Q4) Quais modelos existem para apoiar as atividades de gerenciamento de projetos de reservas de contingência e gerencial?

Estrutura das Questões

Kitchenham (2007) recomenda considerar as questões de estudo a partir da seguinte estrutura PICOC (Population, Intervention, Context, Outcomes, e Comparison) que traduzida para o português seria: População, Intervenção, Contexto, Resultados e Comparação. Para cada pergunta de estudo, os elementos PIO (Population, Interventaion, e Outcome) são apresentados a seguir:

Q1:

População (P): Projetos de desenvolvimento distribuído de software. Intervenção (I): Gerenciamento de projetos. Resultado (O): Desafios no gerenciamento de projetos.

Q2:

População (P): Projetos de desenvolvimento distribuído de software. Intervenção (I): Práticas de gerenciamento de projetos. Resultado (O): Melhor gerenciamento de projetos.

r

Q3: População (P): Projetos de desenvolvimento distribuído de software. Intervenção (I): Ferramentas. Resultado (O): Apoiar o gerenciamento de projetos .

Q4: População (P): Projetos de desenvolvimento distribuído de software. Intervenção (I): Modelos. Resultado (O): Apoiar o gerenciamento de projetos

A Comparação e o Contexto da estrutura PICOC não foram utilizados, uma vez que os objetivos do trabalho não incluem nenhum contexto específico e não buscam a comparação

entre os tópicos investigados.

Estratégia de Busca

A construção dos termos de busca foi realizada seguindo a estratégia composta pelos seguintes passos:

1. A partir das estruturas das questões de investigação (PIO) definidas anteriormente, os principais termos são identificados;

2. É realizada a tradução desses termos para o inglês por ser a língua utilizada nas bases de dados eletrônicas estudadas e nas principais conferências e jornais dos tópicos de investigação;

3. Sinônimos são identificados com a orientação de um especialista no tema de investigação para cada um dos principais termos;

4. As strings de busca são geradas a partir das estruturas das questões e da combinação dos termos chave e sinônimos. São usados OR (ou) entre os sinônimos identificados e AND (e) entre os termos chaves. Algumas adaptações são necessárias de acordo com as necessidades específicas de cada base de dados. Possíveis peculiaridades das bibliotecas digitais e adaptações mediante a isso são registradas.

O Apêndice G apresenta as informações quanto às buscas de cada fonte. Os termos e sinônimos identificados são apresentados abaixo:

Reservas de contingência e gerencial em projetos: Management and contingency reserves in project management, Project risks, Risk control, Contingency reserves, Management reserves;

Gerenciamento de Projetos: Project Management;

Desafios: Challenge, Difficult, Critical Factor, Problem;

Melhores Práticas ou Lições Aprendidas : Practice, Best practice, Good Practice e, Lesson Learned, Success Factor;

Ferramentas: Tool, Software, Program, System, Mathematical optimization; Modelos: Analytical and simulation models, Analytical models for risks, Process, Framework, Method, Technique, Methodology; Os Quadros xy a x'y' apresentam as strings de busca geradas para cada questão de estudo do trabalho. Quadro 1. Strings para as quatro questões de estudo Strings do estudo Para Q1 ("management and contingency reserves in project management«OR» "project risks«OR» "risk control«OR» "contingency reserves«OR» "management reserves") AND ("project management") AND (challenge* OR difficult* OR "critical factor*"OR problem*) Para Q2 ("management and contingency reserves in project management«OR» "project risks «OR» "risk control«OR» "contingency reserves«OR» "management reserves") AND (Practice* OR "Best practice*"OR "Good Practice*"OR "Lesson* Learned"OR "Success Factor*") Para Q3 ("management and contingency reserves in project management«OR» "project risks «OR» "risk control«OR» "contingency reserves«OR» "management reserves") AND ("Project Management") AND (Tool* OR Software*) Para Q4 ("management and contingency reserves in project management«OR» "project risks «OR» "risk control«OR» "contingency reserves«OR» "management reserves") AND ("Project Management") AND (Model* OR Process* OR Framework* OR Method* OR Technique* OR Methodolog*) Fontes de Buscas:

IEEEExplore Digital Library (<http://ieeexplore.ieee.org/>)

ACM Digital Library <http://portal.acm.org>

PMI - Project Management Institute <http://search.pmi.org/Default.aspx/>

<http://www.periodicos.capes.gov.br>

GOOGLE http://www.google.com/advanced_search?hl=en

Critérios de Inclusão e Exclusão dos Estudos

A inclusão de um trabalho é determinada pela relevância (acredita -se que o trabalho é um potencial candidato a tornar -se um estudo primário) em relação às questões de investigação, determinada pela análise do título, palavras-chave, resumo e conclusão. Os seguintes critérios de inclusão foram definidos:

a) Estudos que tratem primária ou secundariamente Dificuldades, Fatores Críticos, Desafios e Problemas em projetos de Desenvolvimento Distribuído de Software relacionados ao gerenciamento;

b) Estudos que apresentem primária ou secundariamente Boas Práticas, Lições Aprendidas e Fatores de Sucesso em projetos de Desenvolvimento Distribuído de Software relacionados ao gerenciamento;

c) Estudos que apresentem primária ou secundariamente Modelos, Processos, Técnicas, Metodologias e Ferramentas de apoio ao Gerenciamento de Projetos no Desenvolvimento Distribuído de Software.

A partir também da análise do título, palavras-chave, resumo e conclusão, são excluídos os estudos que se enquadrem em qualquer dos casos abaixo:

a) Estudos que não estejam disponíveis livremente para consulta na web ou Portal da Capes;

b) Estudos claramente irrelevantes para a estudo, de acordo com as questões de investigação levantadas;

c) Estudos que não respondam nenhuma das questões de estudo;

d) Estudos Repetidos: se determinado estudo estiver disponível em diferentes fontes de busca, a primeira estudo será considerada;

e) Estudos Duplicados: caso dois trabalhos apresentem estudos semelhantes, apenas o mais recente e/ou o mais completo será incluído, a menos que tenham informação complementar;

f) Estudos que apresentem texto, conteúdo e resultados incompletos, ou seja, trabalhos com resultados não concluídos não serão aceitos. Processo de Seleção dos Estudos Primários De acordo com Kitchenham (2007), as buscas iniciais retornam uma grande quantidade de estudos que não são relevantes, não respondendo às questões ou mesmo não tendo relação com o tópico em questão. Logo, estudos totalmente irrelevantes são descartados no início. O Quadro x1 apresenta as etapas do processo de seleção dos estudos primários

Etapas do Processo de Seleção dos Estudos Primários Etapa 1 Dois estudos inicialmente realizam as buscas para identificar os potenciais estudos primários e a partir da leitura dos títulos dos trabalhos que o estudo retorna e palavra-chave, excluem trabalhos que claramente são irrelevantes para as questões investigadas. Etapa 2 Cada estudo chega então a uma lista de potenciais estudos primários. As duas listas são então comparadas e os estudos chegam a uma única lista de potenciais candidatos. Se houver qualquer discordância na inclusão ou exclusão de um estudo, o mesmo deve ser incluído. Etapa 3 A partir da lista unificada com os potenciais candidatos a estudos primários, todos os trabalhos são avaliados por dois ou mais estudos, mediante a leitura do resumo e conclusão, considerando-se os critérios de inclusão e exclusão, para então se chegar a uma lista final de estudos primários. Etapa 4 Os estudos incluídos são documentados através de formulários, assim como todos os trabalhos excluídos e o critério que definiu sua exclusão. Posteriormente, cada estudo primário é lido e através de formulários a extração dos dados

e avaliação da qualidade dos trabalhos é realizada

Avaliação da Qualidade (FALTA)

Procedimento para Análise dos Resultados (FALTA)

2.1 Metodologia da Revisão

2.1 Metodologia da Revisão

Os resultados apresenta a análise desse estudo e serão apresentados levando-se em consideração duas partes distintas. Esses componentes constituem as principais contribuições do estudo e serão detalhados da seguinte maneira.

Análise descritiva da revisão sistemática - apresenta dados gerais da revisão, como: quantidade de trabalhos retornados nas buscas, processo de seleção com o número final de estudos primários, distribuição ao longo dos anos, locais de publicação, os tipos dos estudos, modelo de negócio envolvido, o foco, e por fim, avaliação da qualidade.

Discussão sobre os resultados obtidos - apresenta uma análise dos principais resultados obtidos pelo estudo.

Análise descritiva da Revisão Sistemática

A revisão sistemática foi executada de acordo com o que foi definido no protocolo que se encontra no Apêndice C. A partir das strings e fontes definidas, as buscas primárias retornaram um total de 73 trabalhos, no qual, 06 trabalhos foram identificados no IEEE, 05 na ACM, 07 no PMI, 09 no GOOGLE ADVANCED e por fim, 46 na CAPES. O Gráfico 3 mostra a quantidade de trabalhos retornados por cada engenho de busca. A

revisão foi realizada no período de 22/12/2010 a 31/03/2011. Embora 74 dos estudos retornados tenham sido da CAPES e GOOGLE ADVANCED, essas fontes tiveram uma representatividade pequena se comparada às demais, 16,67 dos estudos primários incluídos na estudo. Já o PMI, retornou sozinho 16,67 dos estudos primários incluídos, o portal IEEE apresentou o maior número de estudos, 66,66.

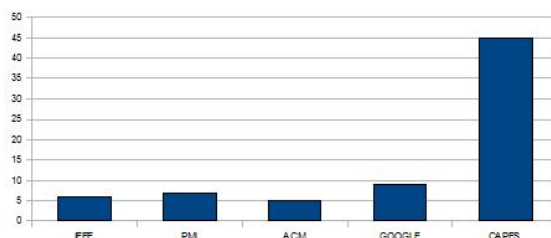


Gráfico 3: Número de trabalhos retornados

A Como se pode verificar, o número de estudos retornados na busca primária foi baixo, porém a partir do processo de seleção, esse número foi menor ainda. O Quadro 4.1 apresenta a evolução em números do processo de seleção de estudos primários. O quadro mostra os valores na busca primária para cada string, no total, 73 estudos retornados, e que a partir da primeira seleção por título e palavra chave, foram identificados 16 estudos potencialmente relevantes para a estudo. Com a leitura do resumo e conclusão dos estudos potencialmente relevantes, e utilizando-se os critérios de inclusão e exclusão, chegou-se a 06 estudos primários, disponíveis no Apêndice A deste trabalho. Assim, 10 trabalhos considerados potencialmente relevantes na primeira seleção foram excluídos e os principais motivos para exclusão foram:

- Não respondiam a nenhuma questão de estudo ; - Identificados duas vezes por fontes diferentes, isto é, repetido ou duplicado; - Não apresentavam texto completo. Os estudos

excluídos e as razões que levaram a sua exclusão estão disponíveis no Apêndice B deste trabalho.

Embora não se tenha limitado por ano a realização das buscas, todos os estudos primários foram publicados entre 1978 e 2010. Isso demonstra e confirma que poucos autores destacam estudos na área de RESERVAS DE CONTINGÊNCIA E GERENCIAL.

Resultados

Nesta revisão sistemática, ficou evidenciada uma pequena contribuição de trabalhos científicos sobre a relevância da RESERVAS DE CONTINGÊNCIA E GERENCIAL EM RISCOS DE PROJETO DE SOFTWARE. Não houve ensaios ou outra manifestação que venha evidenciar o fato ou indícios de um sistema computacional em uso. O tema portanto, torna-se fecundo para pesquisa na área de gerencia de projetos de software.

Capítulo 3

O Modelo Proposto

Por motivo de confidencialidade, o nome da fábrica de software e o sistemas definido não é real, sendo designada nesta estudo genericamente como sendo fabrica de software e ADM sistema. Em alguns casos os artefatos foram mostrados parcialmente ou com pequenas alterações.

3.1 A Medida de esforço

O ADM sistema, objeto de estudo deste trabalho, é baseado no processo de arquitetura matricial, mencionado no item 2.3, que divide o ciclo de desenvolvimento em fases (que por sua vez são divididas em iterações) e fluxos. De forma que, nossa estudo detalhe um segmento de iteração durante a fase de desenvolvimento. A estudo em tela destaca o uso da medida de esforço. O esforço é uma das medidas mais aceitas e utilizadas para compreender e gerenciar processos e projetos de software, por vários motivos. Em primeiro lugar, o esforço empregado em um projeto de software constitui um dos principais

indicadores de seu custo, já que uma fração significativa dos orçamentos dos projetos é dedicada à remuneração da equipe envolvida. Mesmo não sendo suficientes, por si só, para determinar com grande precisão o valor investido, os indicadores de esforço seguramente possuem uma forte correlação com esse custo [DOD00]. Outro fator importante é que o conhecimento do esforço dedicado aos projetos consiste, juntamente com as informações sobre o tamanho dos produtos desenvolvidos, em um dos ingredientes indispensáveis para o cálculo da produtividade das equipes. Essa informação sobre a produtividade será essencial para o conhecimento do desempenho da organização e para a realização de estimativas de prazos e custos em projetos futuros. O planejamento de projetos é outra aplicação direta da caracterização quantitativa do esforço. É necessário conhecer a distribuição típica do esforço ao longo dos passos do processo para que seja possível estimar os recursos humanos que provavelmente serão requeridos em cada etapa dos projetos, e com isso dimensionar as equipes de desenvolvimento. Tal conhecimento só pode ser obtido através da análise de dados históricos de esforço coletados em projetos já desenvolvidos pela organização. O conhecimento da distribuição do esforço ao longo dos passos do processo é útil também para as atividades de melhoria do próprio processo. Os passos que consomem a maior parte dos recursos poderão ser escolhidos como os primeiros alvos das atividades de melhoria, por serem potencialmente capazes de trazer maior impacto sobre o desempenho geral da organização. Como qualquer outra medida, a operacionalização da coleta de dados de esforço exige a definição prévia de algumas regras. Em particular, um aspecto deve ser esclarecido: A unidade de medida escolhida para o esforço, neste estudo - dia trabalhado.

3.2 Registro da medida de esforço

O esforço produzido afeta todos os envolvidos em projetos de software da organização. Ao final de cada dia de trabalho, o gerente de projeto ou responsável imediato deve registrar o esforço empregado nas atividades desempenhadas naquele dia. Os dados de esforço deverão ser registrados, como ilustrado na Figura 15. Nele, deverão ser registradas as seguintes informações sobre cada esforço registrado: -Desenvolvedor - Membro da equipe do time de desenvolvimento a quem se refere o esforço registrado. -Data - Data em que ocorreram as atividades cujo esforço está sendo registrado. Este campo é importante por dois motivos: -Facilitar o controle do registro das horas pelos membros da equipe, auxiliando-os a identificar atividades omitidas e replicadas; -Permitir a identificação da iteração à qual o esforço deve ser associado, a partir dos marcos de início e término de cada uma delas. -Projeto - Projeto de desenvolvimento de software em andamento na organização ao qual se referem as atividades realizadas. Cada esforço registrado deve estar associado a um e somente um projeto. -Fluxo - Fluxo ao qual pertencem as atividades realizadas no período de tempo que está sendo registrado. As opções possíveis são definidas pelo processo de desenvolvimento adotado no projeto em questão. -Duração - Período de tempo dedicado às atividades cujo esforço está sendo registrado. Pelos motivos já formalizado, a duração deve ser expressa em dias trabalhados.

Registro de Esforço Fábrica de Software XYZ LTDA Período: 01 a 30/05/2011
Desenvolvedor Data Projeto Fluxo Duração Engenheiro de Software Pleno SEX 12/03/10 ADM Sistema Implementação 3 dias Engenheiro de Software Júnior QUA 17/03/10 ADM Sistema Implementação 5 dias Engenheiro de Software Sênior SEX 12/03/10 ADM Sistema Implementação 8 dias

FIGURA x - Registro de esforço

FIGURA x - Registro de ...

3.3 Classificação dos membros do time

[ALBUQUERQUE 1998] ressalta que a estimativa da equipe para uma determinada tarefa pode ser capturada por uma tripla (m, M, c) onde "m" é o valor mínimo, "M" o valor máximo e "c" é o grau de confiança para uma determinada métrica. O valor do grau de confiança pode ser alterado de acordo com o projeto (ou mesmo durante um projeto para refletir a precisão da estimativa da equipe durante o mesmo) e são baseadas em dados históricos. (tradução nossa). Uma equipe de desenvolvimento, principalmente em projetos de médio e grande porte, é formada por desenvolvedores de diferentes níveis de formação e experiência. Em nosso estudo adotaremos a classificação para esses diferentes níveis de engenheiros de software, como sendo: júnior, pleno e sênior. Conhecendo a classificação de cada desenvolvedor é possível obter, o esforço total dedicado aos projetos. Essa informação será útil e permite o planejamento de equipes em projetos futuros, levando em consideração a quantidade necessária e o nível de experiência, reduzindo assim, o risco de se alocar equipes muito inexperientes.

3.4 Modelagem ADM sistemas

Nessa seção abordamos o ADM sistemas com base em três modelos de visões: Hierárquica, Seqüencial e Desenvolvimento.

3.4.1 Visão Hierárquica

[ALBUQUERQUE 1998] descreve o modelo de visão hierárquica do sistema como sendo um conjunto de objetos composto por outros objetos, formando uma relação de dependência. Cada objeto também contém métodos de acesso (operações ou interface) que executam diferentes funcionalidades previsto para esse objeto. Neste sentido, a visão hierárquica é um modelo orientado a objetos, onde as relações de dependência são obtidos através de hierarquia. A visão hierárquica apresenta uma visão estática do sistema que é usada durante a atribuição das tarefas e alocação das mesmas. Qualquer objeto no sistema hierárquicas pode ser subdividido em refinamento sucessivos ou durante sua execução, o que pode requerer a atribuição de novos objetos para as mesmas equipes. Destaca ainda, que do ponto de vista formal a visão hierárquica pode ser representado pelo grafo (M, R) , onde M é o conjunto vértices que representam os objetos do projeto, com $R \subseteq M \times M$ que é o conjunto de arestas que representa uma composição sucessivas de um objeto. Nesta visão, a cada dois objetos que não estão na relação transitiva uns dos outros são assumidos como concorrentes. (tradução nossa).

Com base na figura x é possível subdividi-la nos objetos: Manter Diárias, Manter Destino e Manter Função. E ainda subdividindo cada objeto, os mesmos designam os seguintes métodos: Manter Diárias: pés, inc, alt e com ; Manter Destino: pes1,inc1,alt1, exc1,des1, rea1 e con1 ; Manter Função: pes2, inc2, alt2, exc2,des2,rea2 e con2 totalizando três objetos e dezoito métodos.

FIGURA x - visão hierárquica do sistema ADM

3.4.2 Visão Seqüencial

[ALBUQUERQUE 1998] salienta que o modelo de visão seqüencial permite ao projetista especificar os aspectos dinâmicos do sistema. Geralmente consiste de caminhos seqüencial executado pelas interfaces de acesso dos objetos. A Figura apresenta uma visão seqüencial do método: pes do objeto: Manter Diárias.

FIGURA x - visão hierárquica do sistema ADM

3.4.3 Visão Desenvolvimento

[ALBUQUERQUE 1998] salienta que a visão de desenvolvimento pode conter mais objetos que a visão hierárquica. A Figura apresenta o grafo de desenvolvimento do ADM sistema, com distribuição do tempo máximo para cada objeto.

FIGURA x - Grafo de desenvolvimento ADM do sistema.

3.4.4 Casos de Uso

Um caso de uso especifica a seqüência de interações entre um sistema e os agentes externos, e define o uso de uma parte da funcionalidade de um sistema, sem revelar a estrutura e o comportamento interno desse sistema. É usado para identificar como o sistema se comporta em várias situações que podem ocorrer durante sua operação. O CMMI indica que casos de uso podem ser usados na elicitação e análise de requisitos para estabelecer os cenários operacionais do sistema [1]. Então, os casos de uso também descrevem uma solução em alto nível. A Figura x representa o relacionamento de comunicação do ator ADM sistema.

Figura x Ator associado aos casos de uso.

Este estudo utiliza o formato de caso de uso definido pela Rational [2], que compreende os fluxos básicos e sub-fluxos de execução. [WAZLAWICK, 2011, p.39] explica que uma boa prática é agrupar operações tipo CRUD (sigla do inglês que significa: Create, Retrieve, Update e Delete) em casos de uso do tipo "manter", que segue um padrão bem definido, variando apenas as regras de negócio. A figura XXX mostra o indicador do esforço do time no desenvolvimento dos casos de uso ao longo da execução do projeto. A cada ciclo de desenvolvimento, um conjunto tratável de casos de uso é considerado.

O padrão de caso de uso do ADM sistema Manter foi utilizado para documentar os requisitos de manutenção que são caracterizados por operações de Inclusão, Consulta, Alteração e Exclusão, Reativar e Desativar. A Figura x mostra o caso de uso Manter Função.

CDU001 - Manter Classe de Função Descrição: Este caso de uso tem como finalidade manter as possíveis classes de função que serão usadas no sistema de controle de viagens. Atores: Administrador do Sistema. Entradas e Pré-condições: O usuário deve estar autenticado no sistema com o perfil de Administrador do Sistema. Fluxo de Eventos Fluxo Básico:

1. O caso de uso inicia quando o usuário seleciona manter classe de função e solicita a opção Erro: Origem da referência não encontrado.
2. O sistema apresenta tela de filtro de pesquisa conforme Erro: Origem da referência não encontrado.
3. O usuário informa o critério de pesquisa a ser utilizado.
4. O sistema exibe uma lista (Erro: Origem da referência não encontrado) que satisfaça

aos parâmetros de pesquisa informados [Erro: Origem da referência não encontrado] e as opções de Erro: Origem da referência não encontrado, Erro: Origem da referência não encontrado, Erro: Origem da referência não encontrado, Erro: Origem da referência não encontrado ou Erro: Origem da referência não encontrado.

5. O usuário seleciona uma das opções apresentadas.
6. O sistema aciona o sub-fluxo correspondente a seleção do usuário.

FIGURA x - Grafo de desenvolvimento ADM do sistema.

3.4.5 Especificação dos Objetos e Métodos

Embora existam uma grande variedade de representações disponíveis, o método proposto emprega um estilo informal. As informações dos casos de uso são descritas e mapeadas em forma de esquema. A estratégia consiste em identificar o número de objetos e métodos quantificando o número de sub-fluxos e o número de atributos dos sub-fluxos, respectivamente. A tabela x mostra a relação dos objetos instanciados.

Caso de uso Sub-Fluxos objetos Métodos Manter diárias pesquisar pes met1 Manter diárias incluir inc met2 met3 met4 Manter diárias alterar alt met5 met6 met7 met8 met9 met10 Manter diárias consultar con Met11 met12 Manter função pesquisar pes1 met13 Manter função incluir inc1 Met14 met15 met16 met17 met18 met19 met20 Manter função alterar Alt1 Met21 met22 met23 met24 met25 met26 met27 Manter função consultar con1 met28 met29 met30 met31 met32 Manter função desativar des1 met33 met34 met35 met36 Manter função reativar rea1 met37 met38 met39 met40 Manter função excluir exc1 met41 met42 Manter destino pesquisar pes2 met43 Manter destino incluir inc2 met44 met45 met46 Manter destino alterar alt2 met47 met48 met49 met50 met51 met52 Manter

destino Consultar con2 met53 met54 met55 met56 met57 Manter destino desativar des2
met58 met59 met60 met61 Manter destino reativar rea2 Met62 met63 met64 met65 Manter
destino excluir exc2 met66 met67 Tabela x - Grafo de desenvolvimento ADM do sistema.

Capítulo 4

A SIMULAÇÃO DO MODELO PROPOSTO

4.1 Descrição dos dados para a Linguagem de Notação Algébrica

```
data;  
  //Relação do conjunto de objetos de acordo com casos de uso: Manter Diárias, Manter  
set OBJECTS:= initial_object pes inc alt con pes1 inc1 alt1 exc1 des1 real con1 pes2 in  
  // Relação do conjunto de métodos de acordo com casos de uso: Manter diárias, Manter  
set METHODS_NAMES:= met1 met2 met3 met4 met5 met6 met7 met8 met9 met10 met11 met12 met1  
set LINKS: = initial_object pes  
pes inc  
inc alt
```

```
alt con
con final_object
initial_object pes1
pes1 inc1
inc1 alt1
alt1 con1
con1 final_object
initial_object pes2
pes2 inc2
  inc2 alt2
  alt2 con2
con2 final_object;
set METHODS: = initial_object met1
pes met2
inc met3 met4 met5
alt met6 met7 met8 met9 met10 met11
con met12 met13
pes1 met14
inc1 met15 met16 met17 met18 met19 met20 met21
alt1 met22 met23 met24 met25 met26 met27 met28
exc1 met29 met30 met31 met32 met33
des1 met34 met35 met36 met37
rea1 met38 met39 met40 met41
con1 met42 met43
pes2 met44
inc2 met45 met46 met47
```

```
alt2 met48 met49 met50 met51 met52 met53
exc2 met54 met55 met56 met57 met58
des2 met59 met60 met61 met62
rea2 met63 met64 met65 met66
con2 met67 met68
final_object met69;
// Foram escolhidos para efeito de amostragem três casos de uso, que foram desenvolvidos
set TEAMS:= tm1 tm2 tm3;
// Período de tempo igual a dezesseis dias, sendo: três dias (CDU003); cinco dias (CDU004)
set DAYS:= 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16;
param LAST_OBJECT:= final_object;
param num_seq_view:= 3;
set PATHSEQ:= 1 initial_object met1
1 pes met2
1 inc met3 met4 met5
1 alt met6 met7 met8 met9 met10 met11
1 con met12 met13
  1 final_object met69
2 initial_object met2
2 pes1 met14
2 inc1 met15 met16 met17 met18 met19 met20 met21
2 alt1 met22 met23 met24 met25 met26 met27 met28
2 exc1 met29 met30 met31 met32 met33
2 des1 met34 met35 met36 met37
2 rea1 met38 met39 met40 met41
2 con1 met42 met43
```

```
2 final_object met69
3 initial_object met1
3 pes2 met44
3 inc2 met45 met46 met47
3 alt2 met48 met49 met50 met51 met52 met53
3 exc2 met54 met55 met56 met57 met58
3 des2 met59 met60 met61 met62
3 rea2 met63 met64 met65 met66
3 con2 met67 met68
3 final_object met69;
param num_dev_view:= 2;
set PATHDEV:= 1 initial_object pes
1 pes inc
1 inc alt
1 alt con
1 con final_object
2 initial_object pes1
2 pes1 inc1
  2 inc1 alt1
  2 alt1 exc1
  2 exc1 des1
  2 des1 rea1
  2 rea1 con1
  2 con1 final_object
// Custo do projeto para desenvolvimento de três casos de uso do módulo ADM do sistema
param COST:= 4.651,36
```



```
// Tempo de execução.
Param DI := 1 4.0 2 4.5 3 4.8 ;
// Tempo de desenvolvimento. O prazo previsto foi de dezesseis dias, considerando que c
param DD:= 1 8 2 8;
// Carga de cada time
param LAMBDA:= tm1 8 tm2 5 tm3 3;
//Assumindo qualificação profissional superior e disponibilidade inferior.
Param C := tm1 tm2 tm3;
Initial_object 0,0,0
Pes 2.3 2.3 2.0
inc 2.3 2.2 1.9
alt 1.8 2.2 2.1
con 1.8 1.8 1.9
pes1 1.6 1.7 1.7
inc1 1.2 1.3 1.9
alt1 1.7 1.8 1.9
exc1 1.2 1.8 1.9
des1 1.3 1.3 1.8
rea1 1.4 1.8 1.5
con1 1.4 1.9 1.6
pes2 1.6 1.8 1.7
inc2 1.3 1.8 1.5
alt2 1.3 1.9 1.9
exc2 1.6 1.8 1.6
des2 1.6 1.7 1.5
rea2 1.8 2.0 2.1
```

```
con2 2.1 1.9 1.9
final_object 0,0,0;
// Estimativa de custo para cada objeto.
Param T := tm1 tm2 tm3;
Initial_object 0,0,0
pes 0,6 1,1 2,2
inc 0,4 1,1 1,3
alt 0,5 1,2 2,1
con 0,5 1,3 1,9
  pes1 0,4 0,6 1,1
  inc1 0,2 0,6 1,2
  alt1 0,2 0,7 1,0
  exc1 0,2 0,7 0,9
  des1 0,3 0,3 0,9
  rea1 0,3 0,5 1,2
  con1 0,2 0,5 0,9
  pes2 0,7 0,8 1,5
  inc2 0,2 0,5 0,9
  alt2 0,3 0,6 0,7
  exc2 0,7 0,7 0,9
  des2 0,2 0,7 0,9
  rea2 0,3 0,4 0,6
  con2 0,2 0,9 0,9
final_object 0,0,0 ;
// Tempo de desenvolvimento estimado para cada objeto.
Param L := tm1 tm2 tm3;
```

```
Initial_object 0,0 0,0 0,0
pes 0,7 1,0 1,6
inc 0,9 1,0 2,1
alt 0,7 1,2 1,1
con 0,7 1,2 1,1
pes1 0,4 0,6 1,0
inc1 0,3 0,6 1,0
alt1 0,3 0,7 0,6
exc1 0,4 0,7 0,9
des1 0,4 0,3 0,6
rea1 0,4 0,5 1,2
con1 0,3 0,5 0,9
pes2 0,3 0,8 1,1
inc2 0,3 0,6 0,9
alt2 0,3 0,6 0,7
exc2 0,8 0,6 0,9
des2 0,3 0,4 0,9
rea2 0,3 0,4 0,6
con2 0,4 0,7 0,9
final_object 0,0,0 ;
```

... (continuidade após a simulação do modelo)

$$\min\left\{\sum_{i,j} c_{ij}x_{ij}\right\}$$

s.t.

$$Prob\left\{\sum_j a_{ij}x_{ij} \leq b_i\right\} \geq 1 - \alpha_i,$$

where $x_{ij} \in \{0, 1\}$, a_{ij} , b_i and c_{ij} are random variables, and $0 \leq \alpha_i \leq 1$ represents the uncertainty that the equation will be satisfied.

It has been shown in the literature that SILP problems are not linear at all, being strongly dependent on the distribution of the random variables [1]. Computationally, using conventional sequential computers, it is not possible to solve very large problems, because the model size becomes very large. Fortunately, some approximation techniques [?, ?, ?, ?, ?, ?, ?, ?, ?, ?] and parallel algorithms [?, ?] have been presented in literature to solve stochastic programming problems.

The simplest approach is to solve an Integer Linear Programming instance where all the probabilistic data have been replaced with their *expected values*. In this way the problem can be written as a large deterministic problem (Expected Value Formulation [?]). The resulting *deterministic equivalent* problem can be solved using any general purpose optimization package. However, the explicit scenario representation of recourse problems can result in large problems that would fall outside the scope of general purpose solver packages [?].

On the other hand, in Section 4.3 we present a more quantifiable approximation for the SILP problem. The expected value formulation is extended to consider not only the expected value, but design scenarios, where the risk can vary according to the equations (as mentioned before), and in simulation of several design cases. Moreover, in our formulation, the coefficients a_{ij} and c_{ij} can be either normally distributed random variables or actual

values. In this way the number of variables produced by the expected value formulation are reduced.

We present in the following section the notation and how constraints and team estimates are modeled in a mathematical SILP to solve the partitioning problem. We observe that the formulation is a general one, i.e. the cost function and constraints can be customized to specific partitioning problems.

4.2 Notation

The general symbols are presented in Table 4.1, the probabilistic parameters (estimates) in Table 4.2, and decision variables in Table 4.3.

In the following formulation, let i range over objects, m range over methods, j range over teams, p range over paths on graphs, and k range over time units of the project, such as weeks or months.

We assume that if a team implements object o , it will implement all methods $m_k \in M_o$ defined by object o .

4.2.1 Stochastic Integer Linear Programming Formulation

We present below the SILP formulation for the partitioning and risk analysis algorithm in the presence of uncertainty.

1. The *probabilistic parameters of each node* x_{ij} are a 4-tuple $(c_{ij}, d_{mj}, t_{ij}, \lambda_{ij})$ with non-negative numbers for each possible technology option. Each parameter is represented

Tabela 4.1: Symbols Used in the SILP Formulation

Symbol	Description
<i>Objects</i>	set of the objects
<i>Methods_i</i>	set of methods pertaining to object <i>i</i>
<i>Teams</i>	set of available development teams
<i>Weeks</i>	ordered set of weeks (or any other time metric: days, months, ...)
<i>PathsD</i>	set of ordered development dependencies (objects) defined in each path from INITIAL-OBJECT to FINAL-OBJECT in <i>DG</i> development graph
<i>PathsE</i>	set of ordered execution dependencies (methods) defined in each path from initial_method to final_method in <i>SG</i> sequential graph
<i>C</i>	cost desired for the project
<i>DI_p</i>	execution time desired for path <i>p</i> in <i>SG</i> graph
<i>DD_d</i>	development time desired for path <i>d</i> in <i>DG</i> graph
Λ_j	maximum load of team <i>j</i> (manpower)

Tabela 4.2: Probabilistic Parameters Used in the SILP Formulation.

Parameter	Descripton
c_{ij}	estimated cost for the task i by team j (US\$, area, KLOC, ...)
d_{mj}	estimated execution time of method m when implemented by team j ($ms, \mu s, \dots$)
t_{ij}	estimated (development) time taken by team j to implement object i ($hours, weeks, \dots$)
λ_{ij}	estimated load of team j to implement object i ($manhours, manweeks, \dots$)

Tabela 4.3: Decision Variables Used in the SILP Formulation.

Variable	Descripton
x_{ij}	binary variable that assumes value 1 if object i is implemented by team j
γ_{ijk}	binary variable meaning that at week k , team j begins the implementation of object i
Γ_{ijk}	binary variable meaning that at week $k - 1$, team j ends the implementation of object i
a_{ijk}	binary variable meaning that at week k , team j is implementing object i

as the 3 - tuple (m, M, c) (minimum, Maximum, confidence degree).

2. Every object i must be implemented by only one team j , i.e.

$$\forall i \in Objects \quad \sum_{j \in Teams} x_{ij} = 1.$$

3. There is only one initial (γ_{ijk}) and final (Γ_{ijk}) week for each object i implemented by team j , i.e.

$$\forall i \in Objects, j \in Teams \quad \sum_{k \in Weeks} \gamma_{ijk} = x_{ij};$$

$$\forall i \in Objects, j \in Teams \quad \sum_{k \in Weeks} \Gamma_{ijk} = x_{ij}.$$

4. Path execution time. For each path (p) in *SG* Graph (sequencing view), the estimated execution time of the methods implemented by teams (d_{mj}) in the path must satisfy a maximum execution time (DI_p) with a defined probability of certainty ($1 - \alpha_{path}$). Each method m of object i in path $p \in PathsE$ is represented by a 3-tuple (i, p, m) , i.e.

$$\forall P \in PathsE, j \in Teams$$

$$Prob\left\{ \sum_{(i,p,m) \in P} d_{mj} x_{ij} \leq DI_p \right\} \geq 1 - \alpha_{path}.$$

5. Objects developed sequentially are implemented in that order, i.e.

$$\sum_{j,k} k \gamma_{i_2jk} - \sum_{j,k} k \Gamma_{i_1jk} \geq 1$$

for each edge (p, i_1, i_2) of a path p in $PathsD$, i.e. the edge defines a dependency in the development view model p , $j \in Teams$ and $k \in Weeks$. In this way the implementation of object i_2 must begins (γ_{i_2jk}) at least in the week after the end of implementation of object i_1 (Γ_{i_1jk}). Because the variables γ and Γ are binaries, we need multiply by factor k to capture the *week* number.

6. Development time. For each path (p) in DG Graph (development view), the estimated development time of the objects implemented by teams (t_{ij}) in the path must satisfy a maximum development time (DD_p) with a defined probability of certainty ($1 - \alpha_{develop}$). Each object i_1 in path $p \in PathsD$ is represented by a 3-tuple (p, i_1, i_2) , where i_2 is its successor, i.e.

$$\forall P \in PathsD, j \in Teams$$

$$Prob\left\{ \sum_{(p, i_1, i_2) \in P} t_{i_1j} x_{i_1j} \leq DD_p \right\} \geq 1 - \alpha_{develop}.$$

7. Maximum load for team j (k_1 is the first week). At week k_1 : team j can be implementing an object i or not, i.e.

$$\forall i \in Objects, j \in Teams$$

$$a_{ijk_1} = \gamma_{ijk_1} - \Gamma_{ijk_1}$$

where, $\forall_{ij} \Gamma_{ijk_1} = 0$. At other weeks: team j can be implementing object i if it begins in this week ($\gamma_{ijk} = 1$) or it is already implementing it ($a_{ij(k-1)} = 1$) and it did not end yet ($\Gamma_{ijk} = 0$), i.e.

$$\forall i \in Objects, j \in Teams, k \in Weeks$$

$$a_{ijk} = a_{ij(k-1)} + \gamma_{ijk} - \Gamma_{ijk}.$$

At each week, the maximum load of a team (Λ_j) must be satisfied, i.e. the entire effort wasted on all objects by a team at a week must be less or equal to the maximum load of this team with a defined probability of certainty ($1 - \alpha_{load}$). Thus,

$$\forall j \in Teams, k \in Weeks$$

$$Prob\left\{ \sum_{i \in Objects} \lambda_{ij} a_{ijk} \leq \Lambda_j \right\} \geq 1 - \alpha_{load}.$$

8. Object assignment. The final week (Γ_{ijk}) of an implementation is greater or equal the development time (t_{ij}) plus the initial week (γ_{ijk}) with a defined probability of certainty ($1 - \alpha_{assignment}$), i.e.

$$\forall i \in Objects, j \in Teams$$

$$Prob\left\{ \sum_{k \in Weeks} k \Gamma_{ijk} \geq t_{ij} x_{ij} + \sum_{k \in Weeks} k \gamma_{ijk} \right\} \geq 1 - \alpha_{assignment}.$$

9. Cost of project is the summation of all object costs with a defined probability of certainty ($1 - \alpha_{cost}$), i.e.

$$Prob\left\{ \sum_{i \in Objects, j \in Teams} c_{ij} x_{ij} \leq C \right\} \geq 1 - \alpha_{cost}.$$

A designer can select upon the following objective functions to obtain a partitioning into teams and technologies:

1. Minimum development cost;

2. Minimum execution time;
3. Maximum team load.

For a selected objective function from the set presented above, the respective constraint can be easily converted into a objective function to be optimized.

The system partitioning is the problem of finding a mapping $map : Objects \rightarrow (HW, SW) \times Teams$ in such a way that all performance and constraints are fulfilled and the objective function is minimized/maximized.

4.3 System-level Design Exploration and Risk Analysis

As mentioned in the previous section, SILP problems are not linear at all. Therefore, an approximation technique is usually employed to make the solution computationally feasible. We propose an approach based on one of the most common approximation methods suggested in the literature, i.e. the approximation by expected value [?].

In order to understand the implications behind our approximation by expected value, let us restate the general SILP formulation below.

$$\min\left\{\sum_{i,j} c_{ij}x_{ij}\right\}$$

s.t.

$$Prob\left\{\sum_j a_{ij}x_{ij} \leq b_i\right\} \geq 1 - \alpha_i$$

The approximation by expected value considers the probability of the sum of random variables to be equivalent to the sum of the probabilities of the random variables. In this way, the equations above can be rewritten as the following equations:

$$\min\left\{\sum_{i,j} Prob\{c_{ij} \geq 1 - \alpha_o\}x_{ij}\right\}$$

s.t.

$$\sum_j Prob\{a_{ij} \geq 1 - \alpha_i\}x_{ij} \leq b_i$$

In order to analyze how the approximation works, let us assume the constraint $Prob\{a_1x_1 + a_2x_2 \leq b\} \geq 0.5$, where x_1 and x_2 are binary decision variables, and $a_1 = (\mu_{a_1}, \sigma_{a_1})$ and $a_2 = (\mu_{a_2}, \sigma_{a_2})$ are random variables. The approximation by the expected value states that $Prob\{a_1x_1 + a_2x_2 \leq b\} \geq 0.5$ can be approximated by $\mu_{a_1}x_1 + \mu_{a_2}x_2 \leq b$, given that the equation must be satisfied on average half of the cases.

Extending the approximation for design scenarios, and considering a_1 and a_2 normally distributed, the constraint $Prob\{a_1x_1 + a_2x_2 \leq b\} \geq 1 - \alpha$ can be replaced by $(\mu_{a_1} + F^{-1}(1 - \alpha)\sigma_{a_1})x_1 + (\mu_{a_2} + F^{-1}(1 - \alpha)\sigma_{a_2})x_2 \leq b$, where $F^{-1}(1 - \alpha)$ is the inverse distribution function of a standard normal random variable.

In this way, this approximation imposes the restriction that all parameters a_{ij} of a single problem constraint will be over/under estimated in the same way.

4.4 Bibliografia

Referências Bibliográficas

- [1] Biswal98, engenharia, 2001
- [2] SOMMERVILLE, I. Engenharia de software. 6. ed. São Paulo: Addison Wesley, 2003.
- [3] PROJECT MANAGEMENT INSTITUTE, PMI. Um Guia do Conjunto de Conhecimentos em Gerenciamentos de Projetos: Guia PMBOK. Quarta Edição. Local Pennsylvania: Four Campus Boulevard, 2004. 388p.
- [4] REYCK, Bert de. Tools to keep projects on the rails. Financial Times : Mastering Risk, Sept. 8 2005 Disponível em: <<http://financialtimes.printthis.clickability.com/pt/>>. Acesso em: 02 maio 2006.
- [5] BARBOSA, Cristina et al. Gerenciamento de custos em projetos. 2. Ed. Rio de Janeiro: FGV, 2008, 148p. (Publicações FGV Management, série Gerenciamento de Projetos).
- [6] VARGAS, Ricardo. Plano projeto novas fronteiras. Disponível em : <<http://virgilioolano.files.wordpress.com/2011/02/termo-de-abertura-projeto-novas-fronteiras.pdf> >. Acesso em : 22/07/11.
- [7] PAULA FILHO, Wilson de Pádua: Processo Praxis. Disponível em: <http://homepages.dcc.ufmg.br/wilson/praxis/>. Acesso em 11 mar. 2010.

- [8] HARREL, Charles R.; GHOSH, Biman K.; BOWDEN, Royce. Simulation Using ProModel®. McGraw-Hill, 2000.
- [9] KRUCHTEN, Philippe. Rational Unified Process: An Introduction. Reading, MA: Addison-Wesley, 2000.
- [10] AMBLER, Scott W. More Process Patterns. Cambridge: Cambridge University Press, 1999. 369 p.
- [11] BOOCH, Grady. Object-Oriented Analysis and Design with Applications 2nd ed. Redwood City: Benjamin/Cummings, 1994.
- [12] JACOBSON, Ivar. Object-Oriented Software Engineering. Reading, MA: Addison-Wesley, 1994.
- [13] HUMPHREY, Watts S. A Discipline for Software Engineering. Reading, MA: Addison Wesley, 1995.
- [14] PMBOK, Project Management Institute (PMI). PMBOK - Project Management Body of Knowledge. 3ed. 2004. Disponível em: <<http://www.pmimg.org.br>>, Acessado em: 11 out. 2010.
- [15] SEIBERT, Werner. Estudo de caso sobre Gerência de Projetos com foco em Gerência de Riscos, 2004. Monografia - Universidade Luterana do Brasil
- [16] MULCAHY, R. PMP Exam Prep. 5. ed. USA: RMC, 2005, 443p. PMBOK guide. 4. ed. Newton Square, PA. 2008.
- [17] ALBUQUERQUE, Jones: A System-level Design Model for Hardware/Software Co-design. Tese de doutorado em ciência da computação, UFMG, Novembro 2002.
- [18] Rational Unified Process®, Version 2002.05.00. Rational Software Corporation, 2001

- [19] Chrissis, M. B., Konrad, M., Shrum, S. CMMI Guidelines for Process Integration and Product Improvement. Addison-Wesley, 2004.
- [20] WAZLAWICK, Raul Sidnei. Análise e projeto de sistemas de informação orientados a objetos. Rio de Janeiro: Elsevier, 2011.
- [21] MACHADO, CRISTINA FILIPAK. A-Risk: Um método para identificar e Quantificar Riscos de Prazo em Projetos de Desenvolvimento de Software. 2002. 239p. Dissertação (Mestrado em Informática Aplicada) - Pós-Graduação em Informática Aplicada - PPGIA, Pontifícia Universidade Católica do Paraná PUCPR, Curitiba.
- [22] BARKI HENRI; RIVARD SUZANNE; TALBOT, JEAN. Toward an Assessment of Software Development Risk. Journal of Management Information Systems, v. 10, n. 2, 1993, p. 203-225.
- [23] Pesquisa Operacional / Marcos Arenales...[et al.]. Rio de Janeiro: Elsevier, 2007