

**JHONEY DA SILVA LOPES**

**UM MODELO PARA APOIO À DECISÃO EM AVALIAÇÃO DE RISCOS EM  
PROJETOS DE SOFTWARE UTILIZANDO SIMULAÇÃO COM DINÂMICA DE  
SISTEMAS**

**Dissertação apresentada à Universidade  
Federal de Viçosa, como parte das  
exigências do Programa de Pós-  
Graduação em Ciência da Computação,  
para obtenção do título de *Magister  
Scientiae*.**

**VIÇOSA  
MINAS GERAIS - BRASIL  
2014**

**Ficha catalográfica preparada pela Biblioteca Central da Universidade  
Federal de Viçosa - Câmpus Viçosa**

T

L864m  
2014

Lopes, Jhoney da Silva, 1987-  
Um modelo para apoio à decisão em avaliação de risco em  
projetos de software utilizando simulação com dinâmica de  
sistemas / Jhoney da Silva Lopes. – Viçosa, MG, 2014.  
xi, 94f. : il. (algumas color.) ; 29 cm.

Inclui apêndices.

Orientador: José Luis Braga.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f.90-94.

1. Software - Desenvolvimento. 2. Gestão de projetos.  
3. Avaliação de riscos. 4. Tomada de decisões. I. Universidade  
Federal de Viçosa. Departamento de Informática. Programa de  
Pós-graduação em Ciência da Computação. II. Título.

CDD 22. ed. 005.12

*Dedico essa dissertação aos meus pais  
Luzia e Jorge*

*À minha irmã  
Maria Clara*

*Ao meu mentor  
José Luis Braga*

*E aos meus amados amigos.*

## **AGRADECIMENTOS**

Agradeço à minha família: minha mãe, irmã e meus amigos.

Agradeço à minha mãe e irmã, por me fazerem acordar todo dia e lutar para ser alguém melhor, por me apoiarem nas minhas loucuras e por acreditarem em mim.

Agradeço aos meus amados amigos, por serem uma extensão de mim, agradeço aos que estiveram comigo, na saúde, na doença, na alegria, na tristeza e que acima de tudo, me aturam até hoje.

Agradeço às pessoas que passaram pela minha vida e que não estão mais presentes para compartilhar comigo as minhas conquistas e fracassos.

Agradeço ao MEJ UFV, por ter me ensinado que as pessoas podem ser muito maiores do que pensam e por ter me dado a oportunidade de sonhar a Catálise Social.

Agradeço à No Bugs, por ter colocado pessoas incríveis na minha vida e por ter lapidado quem eu sou hoje.

Agradeço à todos do DPI, que de alguma forma contribuíram para a realização deste trabalho.

Agradeço ao meu orientador de pesquisa e mentor de vida, por todo o conhecimento e carinho.

Agradeço à todas as pessoas, que de alguma forma me amaram e eu amei.

Agradeço à Deus, ao universo, à energia e tudo mais, por terem me dado a oportunidade de ter ao o que e a quem agradecer.

À Universidade Federal de Viçosa.

À CAPES pelo apoio financeiro.

## **BIOGRAFIA**

JHONEY DA SILVA LOPES, filho de Luzia da Silva Lopes e Jorge Alves Lopes, brasileiro, nascido em 21 de Novembro de 1987 na cidade de Ubá, no estado de Minas Gerais.

Em 2005, decidiu optar pela ideologia vegetariana, a qual mudou a sua forma de pensar e ver o mundo.

Em 2006, exerceu o serviço militar obrigatório, tendo recebido os prêmios: - Praça Mais Distinta, por liderança e destaque; - Honra ao Mérito.

No ano de 2008, trocou a opção de medicina por computação e ingressou no curso de Ciência da Computação na Universidade Federal de Viçosa – UFV, e graduou em quatro anos, mesmo tempo que atuou na empresa júnior de informática – No Bugs, na qual exerceu as atividades de: analista de projetos em 2008; gerente de projetos em 2009; diretor de projetos em 2010; e presidente em 2011. Em 2012, a No Bugs recebeu o reconhecimento de ser uma das vinte melhores empresas juniores do Brasil, destaque por toda diretoria de 2011, ano no qual teve a honra de ser presidente.

Em 2012, foi aprovado na seleção do programa de pós-graduação do Departamento de Informática – DPI, onde, em Março de 2012, iniciou o mestrado em Ciência da Computação na Universidade Federal de Viçosa – UFV, defendendo a dissertação em Junho de 2014.

Em 2012, participou do concurso nacional realizado pela Fundação Estudar e no meio de seis mil jovens de todo o Brasil, foi um dos dez finalistas do Programa Jovens Inspiradores, único representante de Minas Gerais e consecutivamente da UFV.

Em 2012, foi aprovado como um dos Embaixadores de Negócios Sociais do Brasil, pelo Movimento CHOICE, em que atuou por um ano e meio e exerceu os papéis de: embaixador; mentor; e equipe CHOICE.

Em 2013, fundou juntamente com um amigo a Catálise Social, um projeto com o objetivo de “catalisar” pessoas; transmitindo conhecimento nas mais diversas formas, textos em blog, palestras, workshops e consultorias.

Em 2013, participou do Programa Mineiro de Inovação na Pós-graduação e juntamente com uma equipe multidisciplinar de cinco pessoas, desconhecidos até então, fez parte da única equipe finalista da UFV e recebeu juntamente com sua equipe o prêmio de terceiro lugar pelo projeto.

Em 2013, foi um dos três organizadores principais que conquistaram o direito de organizar pela primeira vez na UFV, o TEDx, a maior conferência independente de Tecnologia, Entretenimento e Design do mundo, que tem como objetivo transmitir ideias de valor que merecem ser espalhadas.

Alguém que sua maior conquista são seus amigos. Uma pessoa que possui mais fracassos que sucessos, que já falhou em três tentativas de empreender e ainda não desistiu.

## SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>vii</b>
<b>LISTA DE TABELAS.....</b>	<b>ix</b>
<b>RESUMO .....</b>	<b>x</b>
<b>ABSTRACT .....</b>	<b>xi</b>
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1 O Problema e sua importância .....	2
1.2 Hipótese.....	4
1.3 Objetivos .....	4
1.4 Método de Desenvolvimento do Trabalho.....	5
1.5 Organização do trabalho .....	7
<b>2 REVISÃO BIBLIOGRÁFICA.....</b>	<b>8</b>
2.1 Riscos em projeto de software .....	8
2.2 Dinâmica de sistemas – DS.....	12
2.3 Estimativa de produção .....	20
2.4 Trabalhos correlatos .....	23
<b>3 GERENCIAMENTO DE RISCOS COM DINÂMICA DE SISTEMAS PARA APOIO À DECISÃO.....</b>	<b>27</b>
3.1 Seleção de variáveis do modelo .....	27
3.2 Modelo de dinâmica de sistemas.....	34
3.3 Ajuste do modelo e estimativas.....	48
<b>4 SIMULAÇÕES DO MODELO.....</b>	<b>56</b>
<b>5 CONCLUSÃO .....</b>	<b>62</b>
5.1 Trabalhos futuros .....	63
<b>APÊNDICE A .....</b>	<b>64</b>
5.2 Equações do modelo .....	64
<b>APÊNDICE B .....</b>	<b>67</b>
5.3 Aplicativo de auxílio à estimativa de software .....	67
5.4 Decisões em projeto utilizando o aplicativo de estimativa .....	77
<b>APÊNDICE C .....</b>	<b>87</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>90</b>

## LISTA DE FIGURAS

Figura 1. Método para a construção de modelos de dinâmica de sistemas.....	6
Figura 2. Visão geral do gerenciamento de riscos do projeto.....	10
Figura 3. Momento da reação diante dos riscos.....	11
Figura 4. Cone de incerteza, melhoria da estimativa de custo ao longo do tempo. ...	12
Figura 5. Diagrama de causalidade do equilíbrio da temperatura do corpo humano em ambiente frio. ....	14
Figura 6. Cone de incerteza, melhoria da estimativa de custo ao longo do tempo. ...	15
Figura 7. Simulação de juro composto de poupança bancária. ....	18
Figura 8. Cálculo realizado pelo simulador para gerar a poupança bancária.....	19
Figura 9. Gráfico do rendimento do modelo dinâmico de uma poupança bancária. .	19
Figura 10. Modelo alto nível de reusabilidade de software. ....	25
Figura 11. Produção e retrabalho em software.....	25
Figura 12. Possível diagrama de causalidade para as seis variáveis presentes. ....	34
Figura 13. Diagrama de causalidade simbolizando o retrabalho em produção de software. ....	35
Figura 14. Modelo de dinâmica de sistemas para avaliação de risco no desenvolvimento de software.....	37
Figura 15. Segmento do modelo responsável por definir o esforço na produção de software.....	38
Figura 16. Segmento do modelo responsável por definir a função de aprendizado. .	43
Figura 17. Comportamento da função de aprendizado. ....	44
Figura 18. Segmento do modelo responsável por definir a rotatividade de pessoal..	45
Figura 19. Segmento do modelo responsável por definir pessoal contratado.....	46
Figura 20. Segmento do modelo responsável por definir o retrabalho. ....	47
Figura 21. Gráfico da variação do impacto dos riscos em projetos de tamanhos diferentes. ....	50
Figura 22. Gráfico da variação do impacto dos riscos em projetos de tamanhos diferentes. ....	51
Figura 23. Risco de atraso do projeto baseado na relação tamanho do projeto e retrabalho.....	52
Figura 24. Redução do prazo de desenvolvimento de um projeto com o aumento do conhecimento. ....	53
Figura 25. Conhecimento contra tamanho do projeto.....	53
Figura 26. Impacto do tamanho da equipe no prazo de desenvolvimento de um projeto de 520 PF. ....	56
Figura 27. Impacto da variação de conhecimento no prazo de desenvolvimento em um projeto de 520 PF. ....	57
Figura 28. Impacto da variação do tamanho do projeto no prazo de desenvolvimento em um projeto de 520 PF. ....	58
Figura 29. Impacto gerado no prazo, a partir da variação do conhecimento da equipe em projetos de tamanhos diferentes. ....	60
Figura 30. Tela principal do aplicativo eProj.....	68

Figura 31. Tela de informações.....	69
Figura 32. Tela de estimativa de prazo. ....	70
Figura 33. Tela de estimativa de tamanho de equipe. ....	71
Figura 34. Tela de estimativa de tamanho de projeto. ....	72
Figura 35. Matriz de risco. ....	73
Figura 36. Tela de introdução da Avaliação do Grau de Risco. ....	74
Figura 37. Tela de introdução da Avaliação do Grau de Risco. ....	75
Figura 38. Tela de resultado da avaliação do grau de risco. ....	76
Figura 39. Tela de exibição do botão de ajuda.....	77
Figura 40. Estimando o tamanho do projeto. ....	79
Figura 41. Estimando o tamanho da equipe. ....	80
Figura 42. Estimando o prazo do projeto. ....	81
Figura 43. Questionário avaliação de risco em tamanho. ....	82
Figura 44. Questionário avaliação de risco em estrutura. ....	83
Figura 45. Questionário avaliação de risco em tecnologia. ....	84
Figura 46. Resultado da avaliação do grau de risco do projeto. ....	85

## LISTA DE TABELAS

Tabela 1. Elementos, notações e descrições de um diagrama de DS.....	16
Tabela 2. Ferramentas de aplicação de dinâmica de sistemas. ....	20
Tabela 3. Coeficientes de uma Cobb-Douglas para produtividade em software. ....	22
Tabela 4. Tabela com informações gerais de projetos de software do Grupo Internacional de Padrões de Benchmarking em Software.....	22
Tabela 5. Tabela de análise e gerenciamento de risco. ....	24
Tabela 6. Levantamento dos fatores de risco do desenvolvimento de software. ....	28
Tabela 7. Fatores de risco organizados por representatividade e relevância. ....	30
Tabela 8. Tabela comparativa entre os grupos propostos na literatura .....	31
Tabela 9. Agrupamento dos riscos por contexto .....	33
Tabela 10. Correlação entre as variáveis dos grupos de risco e os da Figura 15 .....	39
Tabela 11. Ajuste geral do modelo baseado no tamanho do projeto .....	48
Tabela 12. Estimativa de tamanho de equipe baseado no tamanho do projeto.....	54
Tabela 13. Definições do projeto. ....	78
Tabela 14. Variáveis de riscos propostas por Schmidt et al. (2001).....	87
Tabela 15. Variáveis de riscos propostas por Leopoldino (2004).....	88
Tabela 16. Variáveis de riscos propostas por Barki et al. (1993) .....	89
Tabela 17. Variáveis de riscos propostas por Boehm (1991).....	89

## **RESUMO**

LOPES, Jhoney da Silva, M. Sc., Universidade Federal de Viçosa, Junho de 2014.  
**Um modelo para apoio à decisão em avaliação de riscos em projetos de software utilizando simulação com dinâmica de sistemas.** Orientador: José Luis Braga.

A gerência de projetos é amplamente estudada e possui diversas ferramentas que auxiliam na eficiência e eficácia da produção de software. Entretanto, na gerência de projetos, existem inúmeros fatores de risco que envolvem o desenvolvimento de software. Esses fatores de risco interagem de forma dinâmica e não linear, dificultando as tomadas de decisões. O objetivo desse trabalho é criar um modelo de dinâmica de sistemas, que é uma técnica descritiva utilizada para modelagem e simulação de sistemas, envolvendo fatores de risco que influenciam fortemente no desenvolvimento de software. O modelo gerado deve ser uma ferramenta que auxilie os gerentes de projetos em suas tomadas de decisões. Os relacionamentos entre as variáveis foram extraídos a partir de experimentos reais ou empíricos disponibilizados na literatura, com isso aproximando os resultados do modelo aos do mundo real. O modelo possibilita reproduzir cenários de riscos em projetos de software, nos quais seria oneroso ou perigoso testar na realidade, possibilitando a análise dos impactos dos riscos no prazo de desenvolvimento dos projetos.

## **ABSTRACT**

LOPES, Jhoney da Silva, M. Sc., Universidade Federal de Viçosa, June 2014. **A model for decision support in risk assessment in software projects using system dynamics simulation.** Adviser: José Luis Braga.

Project management is widely studied and has several tools that aid in the efficiency and effectiveness of software production. However, in project management, there are numerous risks involved in software development. These risk factors interact in a dynamic and nonlinear way, hindering decision making. The aim of this work is to create a system dynamics model, which is a descriptive technique used for modeling and simulation of systems involving risk factors that strongly influence the development of software. The generated model should be a tool that helps project managers in their decision making. The relationships between variables were extracted from actual or empirical experiments available in literature, thus approaching the model results to the real world. The model allows to reproduce scenarios risks in software projects where it would be costly or dangerous to test in reality, allowing the analysis of the impacts of risks within the project development.

# 1 INTRODUÇÃO

Nas últimas décadas, a indústria de software tem sofrido inúmeros prejuízos com extrapolação de cronograma e excessos de custos, bem como a má qualidade dos produtos entregues. Além disso, existe o aumento da demanda dos clientes por desenvolvimentos “melhor, mais rápido, mais barato” e a constante ampliação da complexidade dos softwares, exigindo maior desempenho no processo de desenvolvimento (KELLNER et al. 1999).

O pensamento humano tem-se modificado com a utilização de software, quanto mais o tempo passa, as pessoas anseiam cada vez mais por soluções automatizadas que facilitem o dia a dia. Com isso, a complexidade desses sistemas e a dificuldade de desenvolvimento, cresceram a níveis não planejados (DAIBERT, 2010).

Histórias de acontecimentos mal sucedidos de software podem ser encontrados em vários estudos de casos (CHARETTE, 1996). Embora a engenharia de software tenha evoluído nos últimos tempos, muitos projetos de software ainda utilizam mais recursos do que o planejado. Uma série de estudos relacionam essas falhas aos riscos presentes na gerência de projetos, como: problemas de comunicação da equipe, equipe de desenvolvimento insuficiente ou inapropriado e falta de habilidade do gerente para gerenciar os projetos (BARROS, 2002).

O risco de projeto de software pode ser definido como o produto da incerteza associada aos fatores de risco do projeto e a magnitude da perda potencial devido ao fracasso do projeto (BARKI et al. 1993). Risco pode ser visto de várias formas. Primeiro, risco gira em torno dos acontecimentos futuros. A questão é: pode-se mudar ações hoje, a fim de criar oportunidades para uma melhor situação amanhã? Segundo, risco envolve mudanças, tais como mudança de mentalidade, opinião, ações, ou lugares. Em um mundo estático, não existe risco, logo, esse é o terceiro aspecto do risco. Risco envolve escolhas, e todas as incertezas que essas escolhas trazem (CHARETTE, 1989).

Riscos podem ser considerados positivos ou negativos (CHARETTE, 1989). Alguns são oportunidades gerenciais, assim encontrar essas oportunidades e

maximizar seus efeitos gera uma vantagem ao gerente. Por exemplo, uma equipe pequena de desenvolvimento acarreta riscos, a saída de um desenvolvedor possui grande impacto negativo ao projeto, mas em contrapartida, uma equipe desse tamanho é mais fácil de ser alinhada, gerenciada e treinada; nessa situação cabe o gerenciamento otimista para manter a equipe motivada e coibir saídas de colaboradores.

Existem diversos riscos negativos no desenvolvimento de software, estudos mostram que 25% dos sistemas de software de larga escala em desenvolvimento são cancelados. Em média, projetos ultrapassam o prazo de desenvolvimento em mais da metade do tempo planejado, projetos maiores geralmente são piores. Apenas 25% dos projetos de larga escala não são considerados “falhas operacionais”, os outros 75%, ou não funcionam como previsto ou não são usados na íntegra (GIBBS, 1994).

Este trabalho apresenta um modelo de dinâmica de sistemas, que permite avaliar os impactos de alguns contextos de riscos, para o auxílio à tomada de decisão no processo de desenvolvimento de software. O modelo permite simular e analisar os resultados de fatores presentes no gerenciamento de software, como tamanho da equipe, retrabalho, tamanho do projeto e outros, e aplicar esses resultados em projetos reais. É possível com o modelo simular diversos cenários de acordo com as características da empresa, equipe e projeto, sem que seja necessário realizar experimentos em projetos reais, economizando tempo e dinheiro.

No modelo foram incluídas variáveis que influenciam diretamente na produtividade de software. Considerando risco como uma possível anomalia, positiva ou negativa, em um processo, entendeu-se que o risco impacta diretamente a produtividade. Foi definido o prazo de desenvolvimento como atributo principal para entender a consequência da interação entre as variáveis do sistema. As variáveis foram identificadas e calibradas com base em experimentos reais ou empíricos publicados na literatura, o que garante a consistência do modelo e o torna coerente com os conhecimentos difundidos pela Engenharia de Software.

## **1.1 O Problema e sua importância**

A tomada de decisão é uma atividade importante e complexa na engenharia de software. Ela é envolvida por problemas classificados em: estruturados, semi-estruturados e não-estruturados (desestruturados) (SPRAGUE, 1995). Estruturados são aqueles que podem ser completamente determinados, porque possuem soluções

amplamente conhecidas. Para lidar com esse tipo de problema, é possível utilizar listas predefinidas, como por exemplo, ações operacionais padronizadas, em que qualquer funcionário ou máquina pode tomar decisões seguindo um fluxograma. Semi-estruturados estão em uma posição intermediária, possuem interseções com problemas estruturados, mas alguns pontos do problema não são bem conhecidos; são desafios que necessitam de um nível médio de julgamento humano e experiência para serem solucionados. Já problemas desestruturados são mais complexos, porque ocorrem ocasionalmente, suas estruturas não são bem definidas e exigem maior julgamento e experiência para solucioná-los. Ou seja, não existem soluções prontas para este tipo de problema; experiências acumuladas com problemas similares, conhecimentos e ferramentas de apoio à decisão, são instrumentos utilizados para este tipo de desafio.

É fundamental para a tomada de decisão, obter a capacidade de enxergar o sistema como um todo. Quanto maior a habilidade dos gestores de visualizarem o projeto de maneira integrada, maior será a capacidade para tomadas de decisão. Muitas vezes os gerentes realizam ações reativas, baseadas em problemas locais e presentes, sem analisar causalidades e correlações entre os vários fatores que envolvem seus problemas (SENGE, 2012).

O gerente de projetos precisa descobrir quais tipos de desafios está enfrentando. Modelos, tabelas e planilhas são ótimas ferramentas para problemas estruturados, para os demais não é possível usar ferramentas que forneçam soluções diretas. São necessárias tecnologias que auxiliem a tomada de decisão, como sistemas de apoio à decisão, simuladores, modelos dinâmicos entre outros (SPRAGUE, 1995). Nessa perspectiva, métodos de controle dos possíveis riscos no desenvolvimento de software são ferramentas indispensáveis (HERMSDORF, 2011).

O impacto mais óbvio da utilização de ferramentas de apoio à decisão é a melhoria da tomada de decisão. Informações e ferramentas de apoio à decisão podem trazer diversos benefícios: fornecer uma visão dos diversos cenários possíveis; munir os gerentes de alternativas para lidar com os riscos do projeto; aumentar a confiança nas tomadas de decisão; por fim e não menos importante, intensificar a velocidade com que as decisões são tomadas (SPRAGUE, 1995).

Grande parte dos riscos inerentes ao desenvolvimento de software estão relacionados a problemas de decisão no nível gerencial. Uma decisão tomada de maneira equivocada pode comprometer todo um projeto levando a grandes prejuízos,

ocasionalmente induzindo à produção de softwares com falhas críticas. Softwares com falhas críticas podem significar grandes perdas no futuro, tanto financeiras como até mesmo de vidas humanas (PETERSON, 1996).

Uma vez entendida a importância do gerenciamento de risco e a complexidade das relações entre as variáveis dinâmicas envolvidas no desenvolvimento de software, o uso de ferramentas que permitam enxergar o sistema como um todo, torna-se indispensável (AMBRÓSIO, 2011). A dinâmica de sistemas (MADACHY, 2007; STERMAN, 2000) é uma dessas ferramentas, e será explorada neste trabalho para permitir uma melhor visualização do contexto e auxiliar às tomadas de decisão relativas aos riscos em software. A dinâmica de sistemas pode ser usada para realizar simulações, e essas por sua vez, servem para auxiliar o gerente de software nas tomadas de decisão (DAIBERT, 2010).

## **1.2 Hipótese**

Simulações com modelos de dinâmica de sistemas que utilizem variáveis representando riscos em desenvolvimento de software e suas inter-relações, podem auxiliar gerentes de projeto em suas decisões relativas a riscos, sua prevenção e gerenciamento.

## **1.3 Objetivos**

O objetivo principal deste trabalho é produzir um modelo de dinâmica de sistemas que inclua as principais variáveis de risco envolvidas no processo de desenvolvimento de software.

Como objetivos específicos pretende-se:

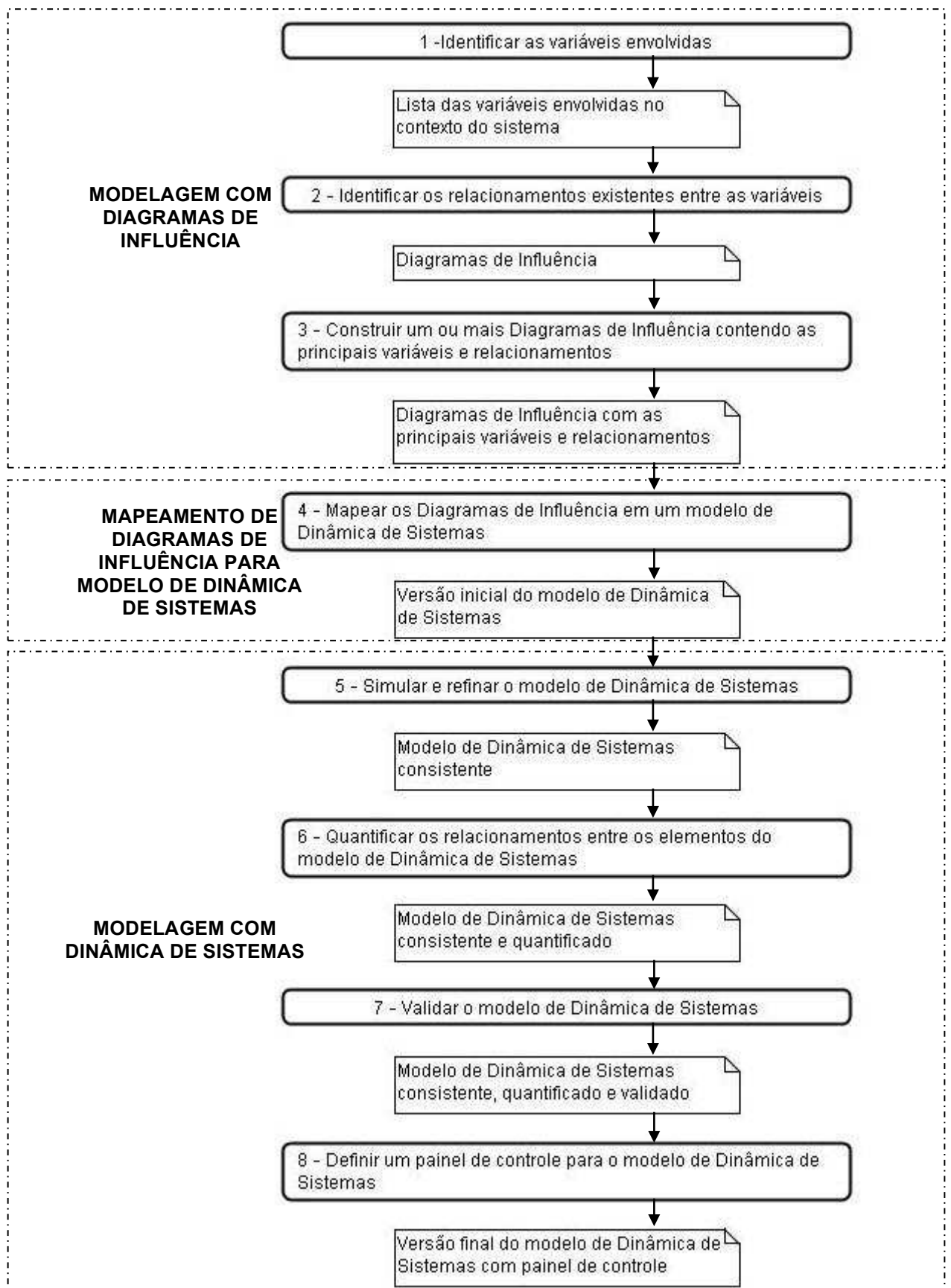
- Identificar as principais variáveis e contextos que têm impacto no processo de desenvolvimento de software;
- Obter um modelo de estoque-fluxo, alimentando-o com parâmetros e valores do ambiente do problema, para que as simulações sejam o mais próximo do real quanto possível;
- Calibrar e testar o modelo obtido utilizando dados disponíveis na literatura técnica sobre riscos;
- Tornar públicos os resultados da pesquisa, na forma de artigos científicos e disponibilização do modelo para execução;

- Tornar o modelo acessível de modo que ele possa ser usado como uma ferramenta de ensino e aprendizagem em disciplinas de engenharia de software e gerência de projetos.

#### **1.4 Método de Desenvolvimento do Trabalho**

Em todos os estágios do trabalho, buscou-se realizar a pesquisa com embasamentos comprovados na literatura. O objetivo dessa abordagem é garantir maior confiabilidade quanto aos resultados obtidos e conclusões alcançadas.

Visando alcançar os resultados que satisfaçam os objetivos do trabalho, foram realizadas as atividades que seguem, de maneira iterativa e não sequencial para a construção do modelo. Desse modo, o método a seguir não visa a definir a construção cronológica do modelo de estoque fluxo, mas sim apresentar os passos que foram realizados. O método é proposto por Ambrósio (2011).



Fonte: (AMBRÓSIO, 2011).

Figura 1. Método para a construção de modelos de dinâmica de sistemas.

## 1.5 Organização do trabalho

Os resultados obtidos neste trabalho estão fortemente embasados em dados e informações publicadas na literatura estudada, tendo sido realizado um levantamento de trabalhos científicos que estivessem relacionados ao contexto do problema proferido. Este trabalho está organizado como se segue.

O Capítulo 2 apresenta o referencial teórico sobre os riscos em projeto de software e dinâmica de sistemas. Também são apresentados trabalhos correlatos que relatam como os fatores intangíveis influenciam e afetam os projetos, as pessoas e a produtividade.

O Capítulo 3 apresenta o processo utilizado para construção do modelo de dinâmica de sistemas e explica as regras que foram utilizadas para seleção das variáveis, como também para obter o coeficiente de relacionamento de forma quantificada entre elas, a fim de aplicá-las às equações do modelo. Além disso, apresenta também o modelo de dinâmica de sistemas construído e descreve as variáveis e os relacionamentos que formam a sua estrutura.

O Capítulo 4 apresenta cenários e simulações, permite aos gerentes visualizar informações que podem extrair do modelo. Exibe também gráficos para auxiliar os gerentes nas tomadas de decisão. Por fim, são apresentados os resultados alcançados com o trabalho.

O Capítulo 5 apresenta as conclusões obtidas e as principais contribuições disponibilizadas por esta pesquisa. São disponibilizadas também sugestões para trabalhos futuros e a conexão dos trabalhos realizados pelo Laboratório de Engenharia de Software – LES da Universidade Federal de Viçosa –UFV.

Ao final do documento são apresentados três apêndices. O Apêndice A contém as equações e valores das variáveis que não são ajustadas pelos usuários do modelo, essas informações estão apresentadas na sintaxe do software utilizado para as simulações. O Apêndice B fornece um resultado extra, nele é apresentado o desenvolvimento de um aplicativo que foi feito a partir da pesquisa, são apresentadas as particularidades e formas de utilização. O aplicativo possui funcionalidades de estimativa de risco e um questionário para avaliar o grau de risco de um projeto em: baixo, médio ou alto. Ele pode ser utilizado tanto por gerentes de software, quanto por estudantes. No Apêndice C são apresentadas as variáveis de risco utilizadas neste trabalho segmentadas pelos autores.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 Riscos em projeto de software

Risco é falta de informação; a incerteza é a base do risco, por essa razão ele está no futuro. O passado é conhecido, logo é possível saber quais incertezas nos afetaram positivamente ou negativamente. Por mais que cada pessoa seja única e viva experiências individuais, o cotidiano é cercado de influências comuns, que nos fornecem informações que são mapeadas e utilizadas por todos, como por exemplo, o melhor caminho para ir ao trabalho, o melhor tipo de investimento, quanto tempo gasta-se de casa até o destino final e outros.

Risco não é associado com certeza, sendo assim, quando uma ação é de risco, não se pode definir a mesma sendo ruim ou boa. Existem diversos esportes de alto risco, como por exemplo o paraquedismo. Durante um salto, o paraquedas pode desdobrar no ar da melhor maneira possível, mas em outros casos isso pode não acontecer. As pessoas não deixaram de praticar o esporte por causa desse perigo, elas mapearam as formas indesejadas da abertura do paraquedas e desenvolveram técnicas para lidar com cada uma delas.

Criar soluções para lidar com as incertezas não se restringe ao esporte. Inovar é assumir riscos e é necessário arriscar para quebrar o “status quo”. Sejam inovações disruptivas<sup>1</sup>, em processos, em modelos ou outros, a intenção é aproveitar uma oportunidade para melhorar algo. Essa busca pela melhoria constante nos obriga a esbarrar continuamente com o desconhecido. Todo caminho novo tem incertezas associadas, mas muitas vezes possui características que interceptam com outros já conhecidos. Gerenciar riscos é utilizar de conhecimentos prévios para minimizar efeitos indesejados. Este trabalho fornece um conjunto de informações para auxiliar o gerente de projetos a enxergar os pontos positivos e negativos das decisões em projetos de software.

O gerenciamento dos riscos em áreas como administração, economia, estatística e matemática financeira, além de ser muito empregado, é tratado como elemento chave para tomada de decisão (DAIBERT, 2010). O risco envolvido em projetos é uma condição ou evento incerto que, se ocorrer, terá um efeito positivo ou

---

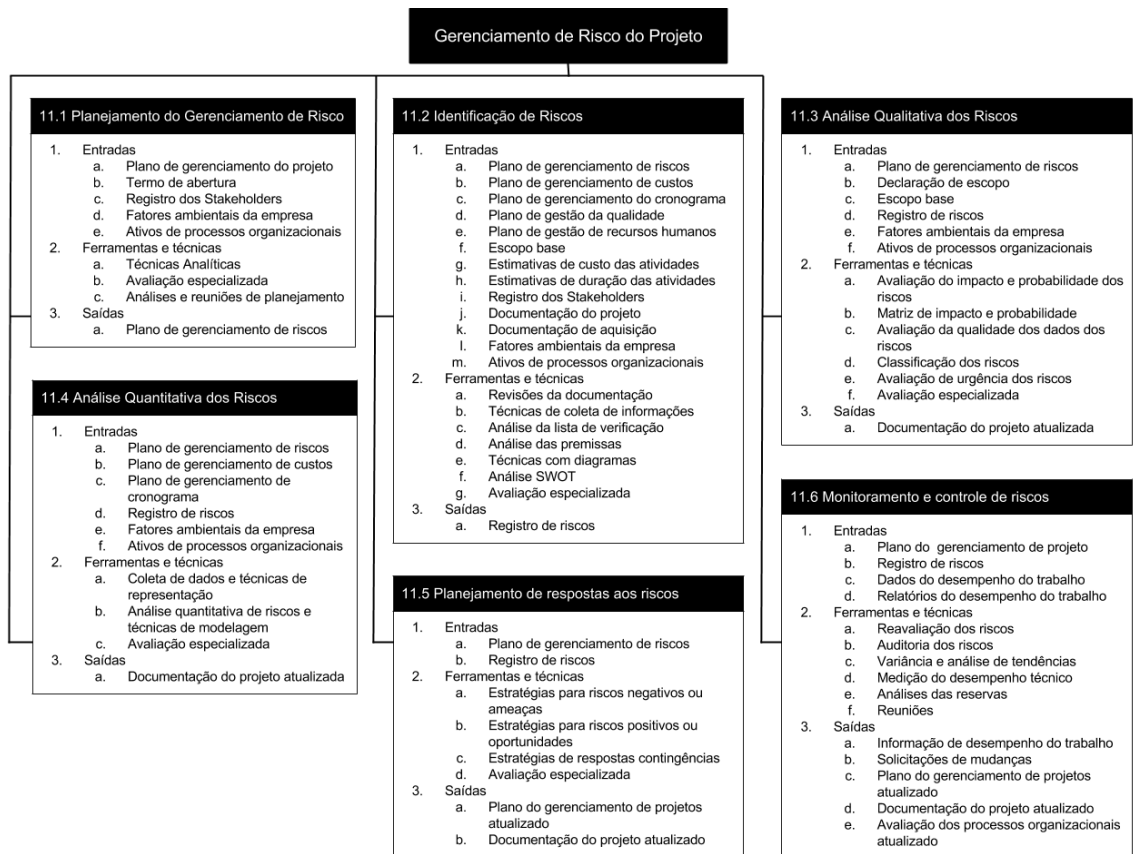
<sup>1</sup> Disruptivo é algo que quebra conceitos, ruptura ou rompimento de algo. Uma inovação disruptiva é um produto, serviço ou tecnologia que rompe ideias preestabelecidas, ou seja, algo que extrapola o até então conhecido, como foi o caso do avião, internet, fotografia digital, wi-fi, iPhone e outros

negativo. Esses efeitos podem ser no prazo, custo, esforço e qualidade. Um risco pode ter diversas causas, e, se ele acontecer, pode ter diversos impactos. Uma causa é uma condição que favorece o acontecimento de resultados positivos ou negativos. Por exemplo, a causa pode ser uma mudança no escopo, a falta de conhecimento ou pessoas insuficientes para executar um projeto, etc (PMBOK, 2013).

O processo de manipulação dos riscos é realizado com a análise e gerenciamento. Na análise estão as etapas de identificação, estimativa e avaliação, já o gerenciamento é entendido como controle do planejamento e o monitoramento do sucesso dos mecanismos de controle. Este processo tem como objetivo básico observar o que pode dar errado e auxiliar nas tomadas de decisão (CHARETTE, 1989).

A **Figura 2** apresenta a visão geral do gerenciamento de riscos proposta no PMBOK (2013), ela é constituída pelos seguintes itens:

- **Planejamento do gerenciamento de risco:** O processo de definição de como conduzir as atividades do gerenciamento de risco;
- **Identificação dos riscos:** Processo de determinação de quais riscos podem afetar o projeto e documentação de suas características;
- **Análise qualitativa dos riscos:** Processo de priorização dos riscos para análise ou ação adicional através da avaliação e combinação de sua probabilidade de ocorrência e impacto;
- **Análise quantitativa dos riscos:** O processo de analisar numericamente os efeitos dos riscos identificados nos objetivos gerais do projeto;
- **Planejamento de respostas aos riscos:** Processo de desenvolvimento de opções e ações para aumentar oportunidades e reduzir ameaças aos objetivos do projeto;
- **Monitoramento e controle de riscos:** Processo de implementação de planos de respostas aos riscos, acompanhamento dos riscos identificados, monitoramento dos riscos residuais, identificação de novos riscos, e avaliação da eficácia do processo de risco durante todo o projeto.



**Fonte:** Elaborado pelo autor e adaptado de (PMBOK, 2013).

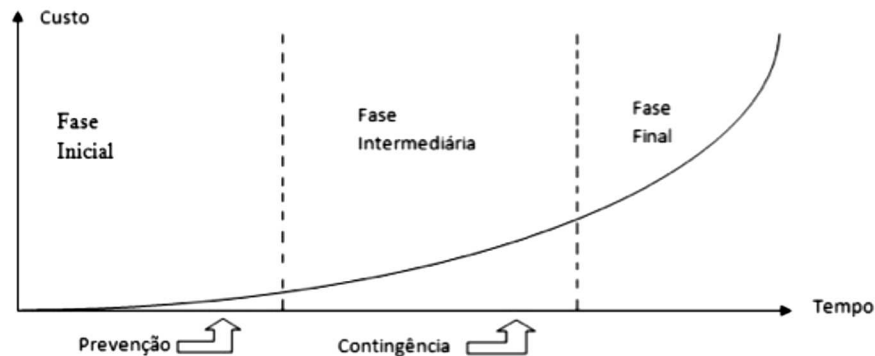
**Figura 2. Visão geral do gerenciamento de riscos do projeto.**

A **Figura 2** apresenta as entradas, ferramentas técnicas e saídas dos principais processos de gerenciamento de risco. Para obter sucesso, uma organização deve estar compromissada em realizar gerenciamento de risco de forma proativa e consciente. Uma escolha consciente deve ser feita em todos os níveis da organização, para identificar e seguir um efetivo gerenciamento de risco durante todo o ciclo de vida de um projeto (PMBOK, 2013). O gerente de projetos pode utilizar a **Figura 2** para organizar um plano de ação de gerenciamento de risco.

A gestão de risco tornou-se uma preocupação global, os efeitos indesejados dos riscos se transformaram em uma apreensão. Por esse motivo em 2009 saiu a norma ISO 31000, lançada no Brasil em 30/11/2009 a ABNT NBR ISO 31000 - Gestão de Riscos, Princípios e Diretrizes. A norma ISO 31000 tem reconhecimento internacional, não tem finalidade de certificação, mas é uma ferramenta de auxílio às empresas, trazendo uma forte vantagem competitiva<sup>2</sup>.

<sup>2</sup> Portal com informações sobre ABNT NBR ISO 31000, acesso em: <http://www.iso31000qsp.org/>

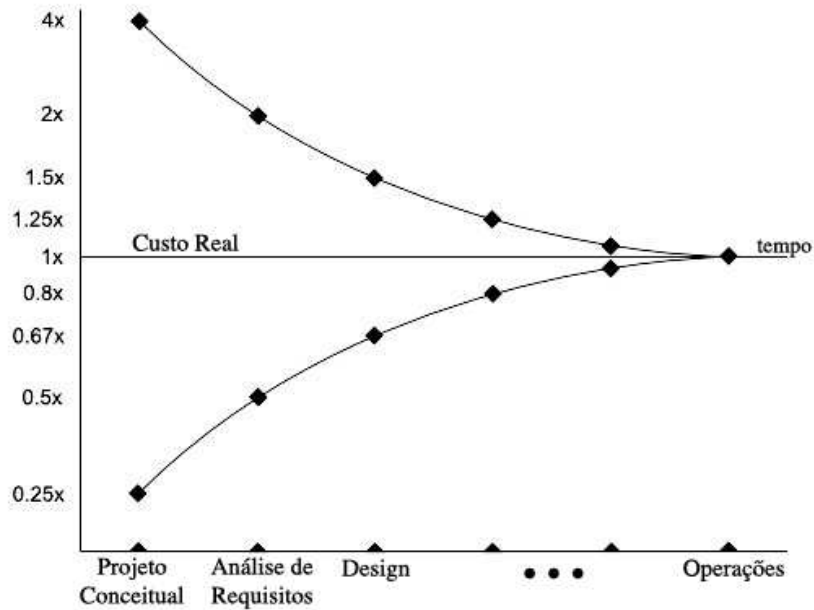
Observa-se que os projetos de desenvolvimento de software, em geral, apresentam atrasos de cronogramas, custos além do planejado e funcionalidades aquém das expectativas. Esses problemas, embora considerados inerentes ao desenvolvimento de software por muitos autores, podem ser minimizados e controlados pelo contínuo gerenciamento de riscos em projetos (SENGE, 2012).



**Fonte:** Elaborado pelo autor e adaptado de (DAIBERT, 2010).

**Figura 3. Momento da reação diante dos riscos.**

O tempo de reação aos riscos é um fator preponderante para a economia, de acordo com Charette (1989), como pode ser visto na **Figura 3**. A reação imediata, feita no momento da identificação e da análise dos riscos, fase inicial, é chamada de reação de prevenção, e acontece antes da decisão final sobre o projeto, alterando as principais variáveis de impacto no projeto, como escopo, qualidade, tempo ou condições financeiras (DAIBERT, 2010). O quanto antes ocorrerem a identificação, prevenção e contingência relacionadas aos riscos, menores serão os custos de impacto ao projeto. Em contraposição, a estimativa de custo do projeto possui maior nível de acerto com o passar do tempo, como pode ser visto a seguir.



Fonte: Adaptado de (CHARETTE, 1989).

**Figura 4. Cone de incerteza, melhoria da estimativa de custo ao longo do tempo.**

A **Figura 4** é conhecida como cone de incerteza, sua leitura nos mostra que no início do projeto a estimativa possui maior taxa de erro. Estimativas realizadas na fase conceitual podem ter um erro de quatro vezes ou um-quarto do valor estimado; o erro se reduz com o aumento do conhecimento sobre o projeto. A **Figura 4** apresenta, de modo simples, o direcionamento para o melhor momento da determinação do custo real do projeto, embora seja compreensível que o erro decresce com o tempo, esse gráfico não leva em consideração a necessidade do cliente em saber o mais rápido possível o valor do projeto. Essa distância entre a assertividade e os interesses do mercado é que fazem surgir diversos riscos, e também indicam momentos em que decisões precisam ser tomadas.

## 2.2 Dinâmica de sistemas – DS

A disciplina de Dinâmica de Sistemas – DS (termo em Inglês: System Dynamics) foi desenvolvida por Jay W. Forrester, um engenheiro elétrico que inventou o acesso aleatório a memória magnética. Em 1956, Forrester, professor no Massachusetts Institute of Technology – MIT, iniciou o “System Dynamics Group”, juntamente com o grupo o campo de dinâmica e sistemas (FORRESTER, 1961; FORRESTER, 1968).

Primeiro é necessário entender os seguintes conceitos: sistema, processo, modelo, simulação e pensamento sistêmico. A partir dessa introdução, o entendimento acerca de dinâmica de sistemas ficará mais claro.

Sistemas podem ser compreendidos como um conjunto de componentes, que sozinhos não são capazes de realizar proezas que juntos são aptos a fazer. Sistemas são compostos por parâmetros independentes e também por variáveis que são dependentes de parâmetros e outras variáveis. Estático e dinâmico, são classificações dadas aos sistemas no mundo real. Se o estado do sistema não altera com o tempo, o mesmo é classificado como estático, se acontece variação, ele é dito dinâmico. Dentro dessa última categoria ele pode ser subclassificado em contínuo, discreto ou combinado. Sistemas contínuos, e.g. experiência pessoal, motivação etc, possuem variáveis que se alteram constantemente, mas sem alterações bruscas ou irregularidades, ponto em que difere de sistemas discretos, e.g. número de defeitos, pedidos de demissão etc, porque estes não possuem um padrão na dinamicidade das variáveis. Já sistemas combinados, como o próprio nome diz, são sistemas híbridos, ou seja, existem as duas classificações dentro de um mesmo sistema (MADACHY, 2007).

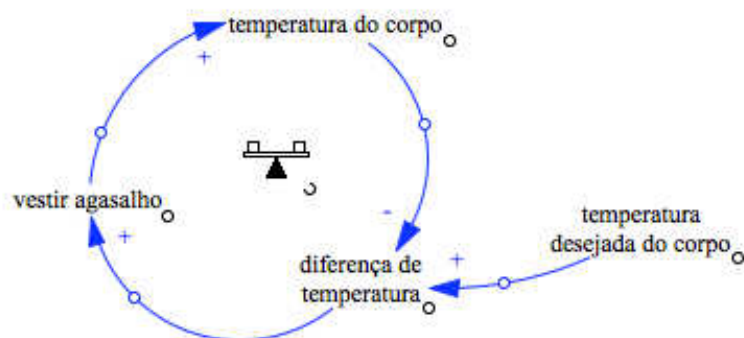
Um processo de software é um conjunto de atividades, métodos, práticas, e transformações usados por pessoas para desenvolver software (CMMI, 2011). Modelos são abstrações de sistemas reais ou conceituais utilizados como substitutos para a experimentação de baixo custo e estudo. Modelos permitem-nos compreender um processo, dividindo-o em partes e analisar como eles estão relacionados (MADACHY, 2007). Os modelos podem ser determinísticos, i.e., sem componentes probabilísticos, ou estocásticos, i.e., com componentes aleatórios; muito poucos processos de softwares são determinísticos.

A simulação é a avaliação numérica de um modelo matemático que descreve um sistema de interesse. Muitos sistemas são muito complexos para soluções analíticas fechadas, portanto, a simulação é usada para exercitar modelos com entradas dadas para avaliar como o sistema se comporta. A simulação pode ser usada para explicar o comportamento do sistema, melhorar os sistemas existentes, ou para projetar novos sistemas muito complexos para serem analisados por planilhas ou fluxogramas (CHARETTE, 1989; MADACHY, 2007; SENGE, 2012). Existem diversas técnicas para realizar simulações com variáveis dinâmicas. Para este trabalho foi escolhida a dinâmica de sistemas, mas não há objeção em utilizar outras

técnicas, análises probabilísticas, Método de Monte Carlo, árvores de decisão ou outros; a escolha por dinâmica de sistemas baseia-se no fato de ser uma ferramenta poderosa e por utilizar o pensamento sistêmico como base para a modelagem, o qual despertou interesse do autor em se aprofundar mais.

O pensamento sistêmico é um conjunto de conhecimentos e ferramentas desenvolvido nos últimos cinquenta anos, o qual visa nos auxiliar a reconhecer padrões e fornecer mecanismos para modificá-los efetivamente (SENGE, 2012). Ele é livre, não possui uma tecnologia obrigatória associada e tem como objetivo enxergar o todo, observar padrões, estruturas, conexões e realizar uma reestruturação das inter-relações de forma mais harmoniosa. É possível conectar variáveis e analisar suas interferências através de um diagrama de causalidade.

Senge (2012) forneceu o entendimento acerca do pensamento sistêmico, desde os impactos nas mais diversas áreas, quanto na construção de diagramas de causalidade e arquétipos. Como o pensamento sistêmico compreende um corpo de métodos, ferramentas e princípios que não possui uma tecnologia associada obrigatória, o seu estudo forneceu uma valiosa compreensão do comportamento de sistemas e da importância das “organizações que aprendem”. Senge (2012) mostra como sistemas complexos possuem algumas vezes milhares de outros sistemas em sua composição, como por exemplo, o corpo humano.

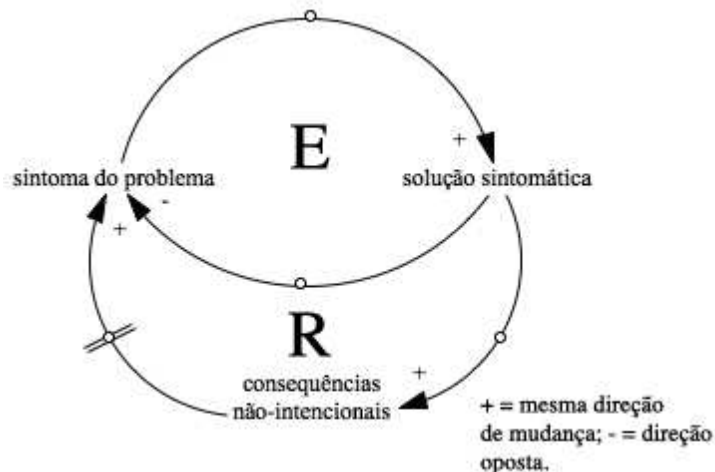


Fonte: (SENGE, 2012).

**Figura 5. Diagrama de causalidade do equilíbrio da temperatura do corpo humano em ambiente frio.**

Esse diagrama de feedback de equilíbrio, apresenta o comportamento do corpo humano para equilibrar a temperatura. Na biologia, sistemas assim são conhecidos como homeostase, capacidade de manter o organismo em equilíbrio, mesmo em ambientes dinâmicos.

Os diagramas de causalidade, também conhecidos como diagramas de influência, constituem a ferramenta principal do pensamento sistêmico. De acordo com esta visão, o mundo opera em circuitos de retroalimentação de reforço e balanceamento. O movimento desses ciclos em conjunto é considerado o comportamento geral do fenômeno ou evento sendo investigado (MADACHY, 2007; SENGE, 2012). Na **Figura 6**, pode ser visto um exemplo de diagrama de causalidade.



**Fonte:** Elaborado pelo autor, adaptado de (SENGE, 2012).



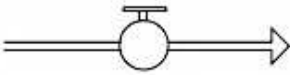

**Figura 6. Cone de incerteza, melhoria da estimativa de custo ao longo do tempo.**


Os símbolos “+ e -” estão representando que quando se altera determinada variável, esta causa um impacto na variável adjacente (ligada a ela) no mesmo sentido (sinal de mais) ou em sentido oposto (sinal de menos). Na modelagem anterior, quando se aumenta *solução sintomática*, seta com “+”, aumentamos o efeito de *consequências não-intencionais* e diminuímos, seta com “-”, *sintoma do problema*. As *consequências não-intencionais* quando crescem, aumentam os efeitos de *sintoma do problema*. O arco cortado por duas linhas paralelas tem o significado de que o efeito produzido não é imediato, ou seja, demanda um tempo até ocorrer, possui um “delay”. As letras “R” e “E” significam, respectivamente, que o caminho

fechado tende a um reforço ou equilíbrio, ou seja, *sintoma do problema* estimula *solução sintomática* e *solução sintomática* desestimula *sintoma do problema*, com isso temos um equilíbrio entre eles.

A partir dos diagramas de influência é possível identificar e visualizar os relacionamentos que existem entre as diversas variáveis do sistema. Esse entendimento acerca da estrutura do problema facilita a construção dos modelos de dinâmica de sistemas, que podem ser usados para simulações. Os modelos de dinâmica de sistema são compostos pelos seguintes elementos: estoques, fluxos e variáveis também conhecidas como conversores (HERMSDORF, 2011). Na **Tabela 1** podem ser vistos detalhadamente os elementos de um modelo de dinâmica de sistemas:

**Tabela 1. Elementos, notações e descrições de um diagrama de DS.**

Elemento	Notação	Descrição
Estoque (Level/Box <i>Variable</i> )		Elementos que sofrem acúmulo ou perda ao longo do tempo. Representam funções vindas dos fluxos.
Fonte/Sumidouro (Source/Sink)		Fontes ou sumidouros são repositórios ou fontes infinitas, e representam fluxos de entrada e saída externos ao processo que não foram especificados no modelo.
Fluxo (Rate)		Fluxos representam as mudanças nos estoques (acúmulo ou perda) e podem indicar as declarações políticas e decisões. Indicam fluxo de material.
Variável Auxiliar ou Constante (Auxiliary/Constant)		Auxiliares permitem a elaboração de detalhes nos fluxos e estoques, como: representar e exercer a influência sobre os valores dos fluxos e estoques por meio dos elos de informação.

<p>Elo de informação (Information Link)</p>		<p>Elos de informação são usados para representar fluxos de informações entre parâmetros do processo aos fluxos e variáveis auxiliares, além de permitir a representação de loops de realimentação.</p>
---	---	---

**Fonte:** Adaptado de (Madachy, 2007).

Dinâmica de Sistemas – DS é um método para a modelagem de sistemas dinâmicos e utilizado fortemente para realizar simulações.

Esse método de modelagem permite representar o comportamento dinâmico dos problemas, o que possibilita analisar, compreender e visualizar de maneira integrada e interconectada como as políticas adotadas, ou a própria estrutura do sistema, afeta ou determina o seu comportamento dinâmico, deixando claras as relações existentes entre as variáveis de decisão de maneira a antecipar colapsos (AMBRÓSIO, 2008).

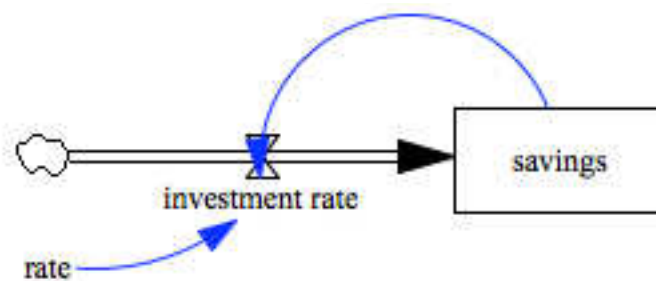
Villela (2005) define uma série de aplicações para a dinâmica de sistemas. Primeiro, aplicações consideradas gerais:

- **Relações de causa e efeito:** A Dinâmica de Sistemas propõe a utilização de diagramas para compreender as relações entre inúmeras variáveis, e com isso realizar aproximações, na tentativa de achar respostas aos problemas levantados.
- **Tempos de resposta:** Para toda ação existe uma reação, mas estas podem não ser instantâneas. Quanto mais longa a resposta para uma determinada ação, mais difícil é a avaliação para descobrir se a ação está realizando o efeito desejado. Os modelos de Dinâmica de Sistemas, auxiliam no estudo deste tempo de resposta.
- **Efeitos de realimentação:** Também conhecido como “feedback”, é o nome dado ao processo pelo qual a saída de um sistema realimenta a entrada, como objeto de controle, com o objetivo de diminuir, amplificar ou estabilizar a saída do sistema.

Após as aplicações gerais, Villela (2005) também cita aplicações específicas para o uso da dinâmica de sistemas:

- **Ciências Sociais:** A Dinâmica de Sistemas tem sido muito utilizada nas Ciências Sociais (Economia, Administração de Empresas, Políticas Públicas, etc). Desde avaliação de risco até gestão de pessoas e tomadas de decisão acerca de investimentos públicos.
- **Ciências Físicas, Químicas e Biológicas:** Estas áreas se caracterizam por sistemas altamente dinâmicos. Efeitos climáticos em um ambiente, aplicação de drogas em organismos, e até mesmo a relação de elementos químicos em uma solução.
- **Engenharias:** O Pensamento Sistêmico atrelado com a DS cria uma ligação forte, na construção e controle de sistemas fechados. Uma rede de alimentação elétrica, a resistência de materiais, etc.

Para entender melhor o funcionamento de um modelo estoque fluxo, a seguir um exemplo de rendimento de dinheiro na poupança bancária.



**Fonte:** Elaborado pelo autor.

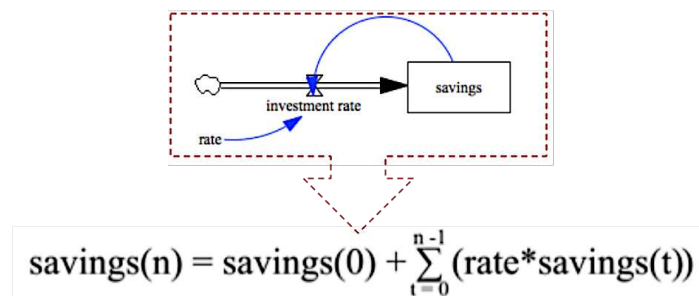
**Figura 7. Simulação de juro composto de poupança bancária.**

Por trás de todo modelo de estoque fluxo, existe uma série de equações matemáticas. Os softwares que realizam as simulações utilizam essas equações para traçar o comportamento do sistema modelado. Para o exemplo da **Figura 7** considere os dados a seguir: taxa de 0,5% ao mês (rate); e investimento de mil reais na poupança (savings) no mês inicial ( $t = 0$ ). Além dos dados citados anteriormente,

necessita-se de uma equação simples para o simulador calcular a taxa de crescimento financeiro mensal.

$$\text{investment\_rate} = \text{rate} * \text{savings} \quad (1)$$

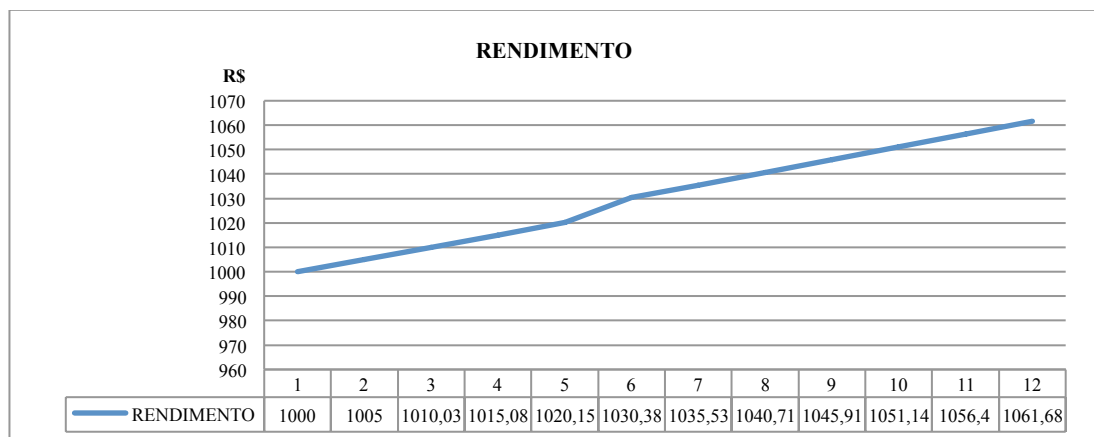
Como o estoque (savings) armazena a soma dos rendimentos, o fluxo da taxa é aplicado dinamicamente sobre a poupança inicial (mil reais), invocando a função de modo recursivo, pode ser entendido que o simulador realiza o seguinte cálculo:



Fonte: Elaborado pelo autor.

**Figura 8. Cálculo realizado pelo simulador para gerar a poupança bancária.**

O gráfico a seguir mostra o rendimento mensal do modelo apresentado.



Fonte: Elaborado pelo autor.

**Figura 9. Gráfico do rendimento do modelo dinâmico de uma poupança bancária.**

O entendimento acerca do pensamento sistêmico e da dinâmica de sistemas, fornece o direcionamento inicial para compreender as ferramentas utilizadas neste trabalho. O exemplo apresentado anteriormente fornece uma visão geral do modelo de DS, da forma que o simulador realiza os cálculos e dos gráficos que podem ser gerados a partir de simulações com o modelo.

Para ajudar nas aplicações de DS existem uma série de softwares que auxiliam na construção de modelos e fornecem um painel para os usuários ajustarem as variáveis e realizarem simulações. Na **Tabela 2**, pode ser visto o nome de algumas das ferramentas mais conhecidas e os tipos de distribuição.

**Tabela 2. Ferramentas de aplicação de dinâmica de sistemas.**

<b>Ferramenta</b>	<b>Distribuição</b>
Powersim <sup>3</sup>	Proprietário
Sphinxes SD Tools <sup>4</sup>	Open Source e gratuito
Stella <sup>5</sup>	Proprietário
Vensim <sup>6</sup>	Proprietário / Versão educacional gratuita
Viasim Solution <sup>7</sup>	Proprietário

**Fonte:** Elaborado pelo autor.

Para a realização deste trabalho foi utilizado o software Vensim, por atender as necessidades da pesquisa e oferecer uma distribuição gratuita para uso educacional.

### 2.3 Estimativa de produção

Para Pendharkar et al. (2008) existem diversos modelos de estimativas que podem ser aplicados à produtividade de software como, COCOMO e Cobb-Douglas.

O COCOMO (do inglês, COConstructive COSt MOdel) é um modelo de estimativa do tempo de desenvolvimento de um produto. Ele é baseado no cálculo das linhas de código gastas para desenvolver um software (BOEHM, 1984). Essa é uma técnica comumente utilizada na engenharia de software (HUSTON, 2001). O COCOMO foi construído para utilizar linhas de código como estimativa de tamanho de projeto. A quantidade de linhas de código necessárias para escrever um software é

<sup>3</sup> <http://www.powersim.com/main/business-simulation/>

<sup>4</sup> <http://www.sphinxes.org/>

<sup>5</sup> <http://www.iseesystems.com/software/Education/StellaSoftware.aspx>

<sup>6</sup> <http://vensim.com/>

<sup>7</sup> <http://www.viasimsolutions.com/>

o fator principal para estimar a produtividade com COCOMO. Por outro lado, utilizar linhas de código para estimar produtividade possui muitas falhas, uma vez que cada empresa pode utilizar um padrão diferente para escrever os seus códigos e, até mesmo, a variação da quantidade de linhas por linguagem de software. Deste modo, a métrica escolhida para estimar tamanho de software nesta pesquisa foi a Análise de Ponto Função<sup>8</sup> - APF.

A APF tornou-se uma técnica muito utilizada na estimativa de tamanho de projeto no mercado de software, a análise de ponto função adquiriu mais força nas últimas décadas (VAZQUEZ, 2009). Com o aumento da força desta técnica, usuários do método de estimativa COCOMO começaram a substituir o tamanho em linhas de código por tamanhos funcionais com APF. Utilizar a análise de ponto função como métrica principal desta pesquisa aumenta a relevância do trabalho para o momento atual da engenharia de software e do mercado de software.

O COCOMO não leva em consideração o tamanho de equipe para estimar o esforço, pelo menos não diretamente na equação, este foi um dos motivos por não ter sido utilizado. Além deste motivo, a função de produção Cobb-Douglas fornece maior liberdade na utilização de sua equação.

A função de produção neoclássica Cobb-Douglas, foi a escolha para esta pesquisa (PENDHARKAR et al. 2008; PINDYCK et al. 2010).

$$y = A * (x)^b * (z)^c \quad (2)$$

Esta função pode ser usada para estimar a produtividade. No trabalho de Pendharkar et al. (2008), os autores mostram que ela pode ser utilizada para realizar um relacionamento entre produção e custo de software. Para ajustes iniciais dos coeficientes da função, foi utilizada a **Tabela 3** fornecida no estudo:

---

<sup>8</sup> A Análise de Ponto Função – APF é uma técnica que visa medir o tamanho funcional de um software para, a partir destes dados, oferecer mecanismos para estimar esforço, prazo e custo (LOPES, 2012).

**Tabela 3. Coeficientes de uma Cobb-Douglas para produtividade em software.**

Coeficientes	Média	Intervalo	
		Mínimo	Máximo
A	10,08	4,31	15,85
b	0,788	0,719	0,857
c	0,613	0,542	0,684

**Fonte:** Elaborado pelo autor, adaptada de (PENDHARKAR et al. 2008).

A função adaptada para esta pesquisa será melhor descrita no decorrer do trabalho. Além da equação, o estudo de Pendharkar et al. (2008) forneceu uma base de dados com informações importantes de projetos de software do Grupo Internacional de Padrões de Benchmarking em Software (International Software Benchmarking Standards Group - ISBSG).

**Tabela 4. Tabela com informações gerais de projetos de software do Grupo Internacional de Padrões de Benchmarking em Software.**

Variáveis	Média	Mínimo	Máximo
Tamanho do Projeto	521,04 Pontos Função	11 PF	9803 PF
Esforço	5.217,7 homem-horas	17,0 h-h	150.040,0 h-h
Tamanho da Equipe	8 pessoas	1 pessoa	77 pessoas

**Fonte:** Elaborado pelo autor, adaptada (PENDHARKAR et al. 2008).

Na **Tabela 4** é apresentada a sigla Pontos de Função – PF, para o tamanho do projeto. Esta é uma medida do tamanho funcional de um projeto de software, ela é produzida a partir da análise de ponto função do projeto. Na **Tabela 4**, o esforço é dado na unidade de homem-horas, uma medida que significa a quantidade de horas que um homem teria que trabalhar para realizar o projeto. Tamanho da equipe é o número de pessoas envolvidas na realização do projeto. Além desses dados, foram obtidos com uma microempresa de software de Viçosa que utiliza a métrica em pontos função, os dados de um projeto da empresa. Para um projeto de duzentos e setenta pontos função realizado por uma equipe de três pessoas, foi fornecida a informação de que a empresa gasta o prazo de três meses para o desenvolvimento. Esses foram dados importantes para os ajustes das simulações do modelo de DS.

Além de utilizar a APF, combinar equações de produtividade da economia com métodos de estimativas já utilizados na engenharia de software, fornecem equações que podem ser facilmente computáveis e transformadas em código, o que facilita a

distribuição da solução apresentada por este trabalho, em qualquer plataforma de software, e não somente como modelo de DS.

## **2.4 Trabalhos correlatos**

Entre os trabalhos que utilizam a visão de risco no desenvolvimento de software, foram importantes para esta pesquisa os trabalhos de (BARKI et al. 1993; BOEHM, 1984; BOEHM, 1991; CHARETTE, 1989; LEOPOLDINO, 2004; PENDHARKAR, 2008; SCHMIDT et al. 2001). Esses trabalhos apresentam o impacto dos riscos no desenvolvimento de software, na forma de listas de riscos, estudos empíricos e até mesmo equações de produtividade no desenvolvimento de software. Alguns trabalhos são estudos empíricos, e outros utilizaram de banco de dados de projetos de software para realizarem suas pesquisas, os dados capturados nesses estudos foram de total importância para a elaboração do modelo de DS, ajustes das variáveis e comparação dos resultados das simulações.

Entre os diversos trabalhos relacionados à aplicação de dinâmica de sistemas no desenvolvimento de software, destacam-se para esta pesquisa os trabalhos (MADACHY, 2007; SENGE, 2012; STERMAN, 2000). Dissertações anteriores do laboratório de engenharia de software da Universidade Federal de Viçosa, também foram muito importantes, destacando-se (COELHO, 2013; COSTA, 2012; AMBRÓSIO, 2011; DAIBERT, 2010).

Segundo os estudos feitos por Barki et al. (1993, 2001), mais da metade dos projetos analisados ultrapassam os valores previstos no orçamento e apenas 58% conseguem entregar o projeto no prazo estimado. Além desses dados, foram retirados do autor uma lista de riscos em projetos de software e um estudo de distribuição desses riscos em grupos de contexto. Para Boehm (1984, 1991) as consequências negativas dos riscos impactam várias dimensões: gerentes, clientes, desenvolvedores e usuários. Desses estudos foi obtida uma lista generalista de riscos em projeto de software, a qual foi utilizada nesta pesquisa.

Schmidt et al. (2001) teve grande impacto na construção deste trabalho. Para os autores, a literatura estava cheia de listas de riscos, sendo a maioria inadequada por serem relativamente velhas e/ou constituírem estudos empíricos rasos que, ou não proporcionavam a realidade dos gerentes de projetos, ou apresentavam a visão limitada de apenas um grupo. Os autores consideram sua pesquisa importante, por apresentarem um estudo empírico realizado em três países de culturas diferentes, e

contextos socioeconômicos distintos, sendo esses: Hong Kong, Finlândia e Estados Unidos da América. Utilizando a pesquisa de Schmidt et al. (2001), Leopoldino (2004) aplicou-a no Brasil, com isso, este estudo foi composto por variáveis de risco de trabalhos empíricos aplicados em quatro países diferentes.

Charette (1989) forneceu uma visão geral sobre risco e seus impactos no mundo e no desenvolvimento de software, além de apresentar as fases do processo de manipulação dos riscos, análise e gerenciamento. Esse processo tem como objetivo básico observar o que pode dar errado e tomar decisões acerca disso. A **Tabela 5** representa dois processos de gerenciamento de risco.

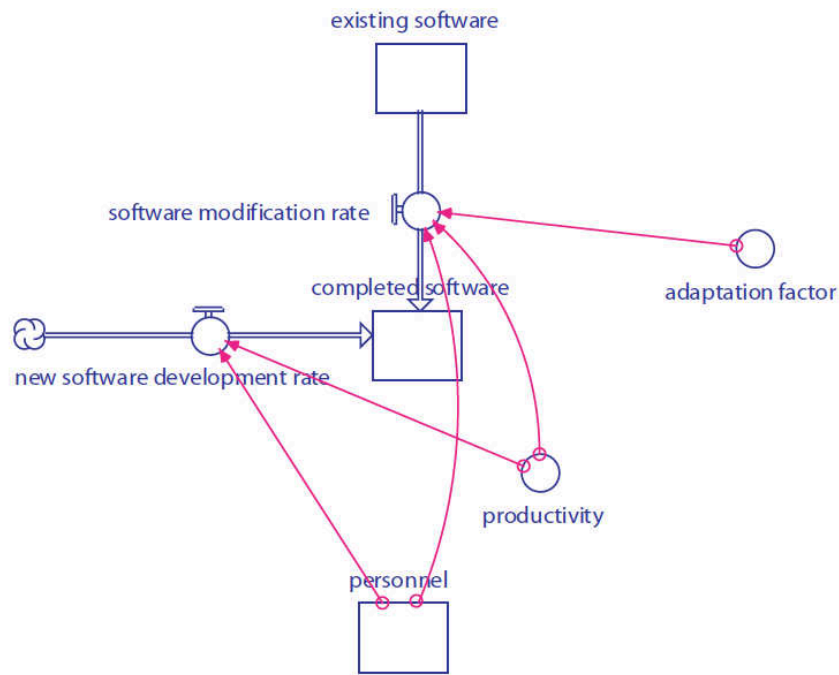
**Tabela 5. Tabela de análise e gerenciamento de risco.**

<b>Charette (1989)</b>	<b>Boehm (1991)</b>
A. Análise de Risco	A. Análise de Risco
1. Identificação dos Riscos	1. Identificação dos Riscos
2. Estimativa dos Riscos	2. Análise dos Riscos
3. Avaliação dos Riscos	3. Priorização dos Riscos
B. Gerenciamento de Risco	B. Gerenciamento de Risco
1. Planejamento dos Riscos	1. Planejamento do Gerenciamento dos Riscos
2. Controle dos Riscos	2. Resolução dos Riscos
3. Monitoramento dos Riscos	3. Monitoramento dos Riscos

**Fonte:** Elaborado pelo autor

Os dois processos são muito similares, exceto por algumas diferenças de nomenclatura e abrangência. Após o levantamento dos riscos e seus impactos, a pesquisa caminhou no contexto das equações para o modelo de DS.

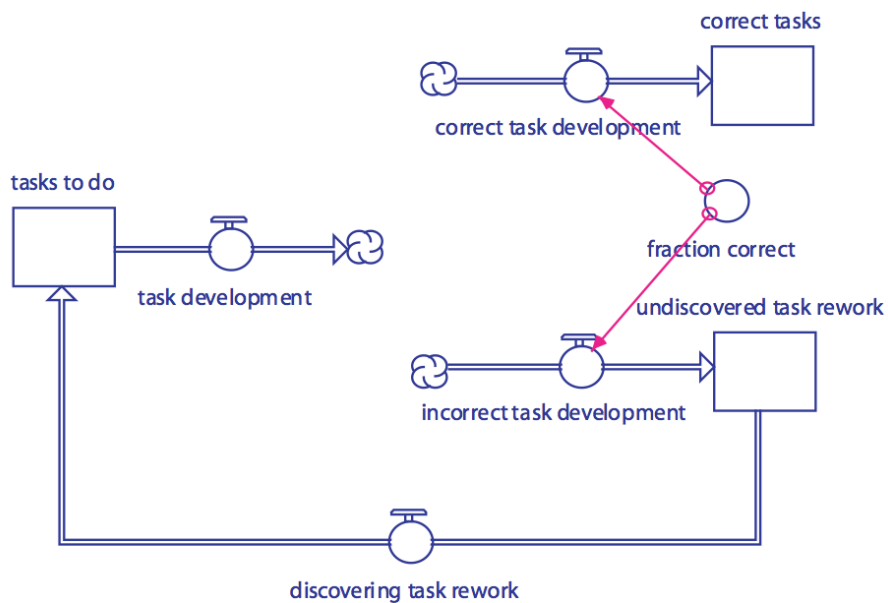
Sterman (2000) e Madachy (2007) forneceram os principais conhecimentos sobre a dinâmica de sistemas. Sterman (2000) fornece uma visão estratégica para negócios e políticas públicas e constrói a transformação de diagramas de influência em modelos de dinâmica de sistemas. Madachy (2007) foi mais utilizado para esta pesquisa. Por, além de fornecer conhecimentos similares ao do primeiro autor citado, apresenta uma série de exemplos de modelos iniciais para lidar com o processo de desenvolvimento de software.



Fonte: (MADACHY, 2007).

**Figura 10. Modelo alto nível de reusabilidade de software.**

A **Figura 10** apresenta uma visão geral de um modelo de DS para lidar com reusabilidade de software. Outro exemplo, e este importante para esta pesquisa, lida com retrabalho em software, **Figura 11**.



Fonte: (MADACHY, 2007).

**Figura 11. Produção e retrabalho em software.**

A **Figura 11** forneceu embasamento inicial para a construção de uma importante parte deste trabalho. Madachy (2007) alerta para o fato de que, retrabalho ocorre para corrigir tarefas incorretas, sejam essas identificadas por desenvolvedores, clientes, gerentes ou usuários.

A combinação de todos os conhecimentos e estudos apresentados anteriormente, utilizando um pouco da essência de cada um para a construção deste trabalho, fornecem um novo produto no que tange a avaliação de risco no desenvolvimento de software, um modelo de dinâmica de sistemas que visa o apoio à decisão a gerentes de projetos de softwares.

### 3 GERENCIAMENTO DE RISCOS COM DINÂMICA DE SISTEMAS PARA APOIO À DECISÃO

Para este trabalho foi realizada uma pesquisa classificada como aplicada ou tecnológica, utilizou-se de conhecimentos básicos para a produção de soluções para aplicação prática. Todas as variáveis que compõem o modelo desta pesquisa foram levantadas, analisadas, retiradas e justificadas com base na literatura, não constituindo assim, uma pesquisa experimental.

Conforme já pode ser verificado, a proposta deste trabalho parte do princípio de se utilizar um modelo de dinâmica de sistemas que visa fornecer aos gestores apoio às tomadas de decisão acerca dos riscos que impactam o desenvolvimento de software. O objetivo desta abordagem é permitir que o conhecimento do gestor possa ser integrado ao modelo; o gerente de projetos pode ajustar as variáveis do modelo à sua realidade, realizar simulações e utilizar estes resultados para tomar decisões.

Neste capítulo serão apresentados os detalhes da construção do modelo: variáveis de risco agrupadas por contexto, equações do sistema, ajustes do modelo, simulações e resultados.

#### 3.1 Seleção de variáveis do modelo

As variáveis e fatores de risco utilizados no estudo foram retiradas de trabalhos experimentais presentes na literatura. Na **Tabela 6** são listados trinta e três fatores de risco, organizados por ordem de relevância e extraídos dos trabalhos de (BARKI, 1993; BOEHM, 1991; LEOPOLDINO, 2004; SCHMIDT et al. 2001).

A lista inicial foi um trabalho exploratório com gerentes de projeto de software de diversas localidades, Leopoldino (2004) no Brasil e Schmidt et al. (2001) em Hong Kong, Finlândia e Estados Unidos da América.

As variáveis apresentadas por Barki (1993) e Boehm (1991) eram mais generalistas e foram destrinchadas na lista inicial. Quando alguma variável proposta não tinha interseção com nenhum outro item já existente, ela era inserida ao fim da lista. Na **Tabela 6** podem ser vistos todos os trinta e três fatores de risco, cada uma das quatro últimas colunas da tabela representam um dos autores pesquisados na literatura.

**Tabela 6. Levantamento dos fatores de risco do desenvolvimento de software.**

Fatores de Risco	Autores			
	a	b	c	d
1. Mudança de Escopo/objetivos	X	X	X	X
2. Falta de envolvimento adequado dos usuários	X	X	X	
3. Requisitos mal entendidos e/ou mal definidos	X	X		X
4. Escopo/objetivos pouco claros ou equivocados	X	X		X
5. Prazos e tempo para tarefas mal estimados	X	X		X
6. Gerenciamento impróprio de mudanças	X	X		X
7. Volatilidade nos requisitos (falta de requisitos estáticos)	X	X		X
8. Custos mal estimados	X	X		X
9. Falta de poderes para o gerenciamento de projetos	X			
10. Conflito entre departamentos de usuário	X	X		
11. Falha em gerenciar as expectativas finais dos usuários	X	X		
12. Planejamento inexistente ou inadequado	X	X		
13. Pessoal envolvido insuficiente/inapropriado	X	X	X	X
14. Falta de conhecimento/competência dos envolvidos no projeto	X	X	X	X
15. Falta de Cooperação dos usuários	X	X	X	
16. Falta de metodologia efetiva em gerenciamento de projetos	X	X		
17. Controle pobre ou inexistente	X	X		
18. Adoção de novo método/tecnologia	X	X	X	
19. Falha em obter comprometimento do cliente	X	X		
20. Definição imprópria de papéis e responsabilidades	X	X		
21. Falta de comprometimento da alta gerência	X	X	X	
22. Falta de motivação da equipe	X			
23. Falta de habilidade para o gerenciamento de projetos	X	X		
24. Assunto novo ou não familiar	X	X	X	
25. Mudança no proprietário do projeto ou na alta gerência		X		
26. Rotatividade da equipe		X		
27. Projeto com múltiplos fornecedores		X		
28. Usar nova metodologia de desenvolvimento em projetos importantes		X		
29. O sistema possui integração e interface com outros sistemas			X	X
30. Sistema complexo			X	
31. Tarefas complexas			X	X
32. Falta de tecnologias maduras/existentes			X	
33. Deficiência de execução em tempo real				X

**Fonte:** Elaborado pelo autor.

**Respectivamente os autores representados em cada coluna são, (a) Leopoldino (2004), (b) Schmidt et al. (2001), (c) Barki et al. (1993), (d) Boehm (1991). O “X” marca quais fatores de risco de software foram encontrados nas pesquisas de cada um desses autores.**

A partir dessa lista, foram realizadas algumas ações para classificar e ordenar os riscos listados. Os critérios utilizados foram:

- Representatividade;
- Relevância;
- Grupos de fatores de risco;

O critério de representatividade foi apresentado na **Tabela 6**, nele atribuiu-se um ponto para cada risco presente na literatura de cada autor (e.g., se o fator de risco estava presente na pesquisa de um autor, ele recebia um ponto, se estava na de dois autores, recebia dois pontos e assim sucessivamente), os riscos divididos por autor pode ser visto no Apêndice C.

No critério de relevância, foi atribuído um ponto extra para alguns riscos. Utilizando o estudo exploratório de Schmidt et al. (2001), onze fatores de risco de sua lista foram apresentados por gerentes de projeto de software dos três países da pesquisa, i.e., Hong Kong, Finlândia e Estados Unidos. Esses fatores também aparecem na pesquisa feita por Leopoldino (2004) no Brasil. Os onze fatores foram considerados relevantes, por serem de amplitude global e por terem sido listados por gerentes de projetos de culturas diferentes e inseridos em contextos socioeconômicos distintos. A seguir os onze fatores selecionados por representatividade:

- Falta de comprometimento da alta gerência;
- Falha em obter comprometimento usuário/cliente;
- Requisitos mal entendidos;
- Falta de envolvimento adequado dos usuários;
- Falta de conhecimento/competência dos envolvidos no projeto;
- Volatilidade dos requisitos (falta de requisitos estáticos);
- Mudança de escopo/objetivos;
- Adoção de novos métodos/tecnologia;
- Falha em gerenciar as expectativas finais dos usuários;
- Pessoal envolvido insuficiente/inapropriado;
- Conflito entre departamentos de usuário.

Na **Tabela 7** podem ser vistos os riscos organizados pelos critérios de representatividade e relevância.

**Tabela 7. Fatores de risco organizados por representatividade e relevância.**

<b>Fatores de Risco</b>	<b>Pontos Finais</b>
Mudança de Escopo/objetivos	5
Falta de conhecimento/competência dos envolvidos no projeto	5
Pessoal envolvido insuficiente/inapropriado	5
Adoção de novo método/tecnologia	4
Falta de comprometimento da alta gerência	4
Falta de envolvimento adequado dos usuários	4
Requisitos mal entendidos e/ou mal definidos	4
Volatilidade nos requisitos (falta de requisitos estáticos)	4
Assunto novo ou não familiar	3
Conflito entre departamentos de usuário	3
Custos mal estimados	3
Escopo/objetivos pouco claros ou equivocados	3
Falha em gerenciar as expectativas finais dos usuários	3
Falha em obter comprometimento do cliente	3
Falta de Cooperação dos usuários	3
Gerenciamento impróprio de mudanças	3
Prazos e tempo para tarefas mal estimados	3
Controle pobre ou inexistente	2
Definição imprópria de papéis e responsabilidades	2
Falta de habilidade para o gerenciamento de projetos	2
Falta de metodologia efetiva em gerenciamento de projetos	2
O sistema possui integração e interface com outros sistemas	2
Planejamento inexistente ou inadequado	2
Tarefas complexas	2
Deficiência de execução em tempo real	1
Falta de motivação da equipe	1
Falta de poderes para o gerenciamento de projetos	1
Falta de tecnologias maduras/existentes	1
Mudança no proprietário do projeto ou na alta gerência	1
Projeto com múltiplos fornecedores	1
Rotatividade da equipe	1
Sistema complexo	1
Usar nova metodologia de desenvolvimento em projetos importantes	1

**Fonte:** Elaborado pelo autor.

**Fatores de risco com a mesma pontuação foram colocados em ordem alfabética para facilitar a leitura.**

Um dos resultados do estudo foi desenvolver um conjunto de grupos de fatores de risco fundamentado na literatura. Esses grupos foram utilizados para um melhor entendimento acerca dos riscos e como alicerce para a construção do modelo.

A partir dos riscos listados na **Tabela 7**, é possível observar que alguns possuem impactos semelhantes nos projetos de software, como por exemplo: *mudança de escopo/objetivos; escopo/objetivos pouco claros ou equivocados; requisitos mal entendidos e/ou mal definidos; volatilidade nos requisitos* (falta de requisitos estáticos) e outros. Na **Tabela 8** é apresentada uma comparação entre grupos sugeridos por (BARKI et al. 1993; LEOPOLDINO, 2004; SCHMIDT et al. 2001). Segmentar os riscos em grupos facilita o entendimento do gerente de projetos acerca de cada parte do modelo, além de fornecer o conhecimento do possível campo de impacto daquele determinado risco. A seguir uma tabela comparativa de grupos de riscos de software de Barki et al. (1993), Schmidt et al. (2001) e Leopoldino (2004).

**Tabela 8. Tabela comparativa entre os grupos propostos na literatura**

Barki et al. (1993)	Schmidt et al. (2001)	Leopoldino (2004)
Novidade Tecnológica	Tecnologia	Conhecimento e incerteza tecnológica
Complexidade da Aplicação	Requisitos	Escopo e Requisitos
Tamanho da Aplicação	Escopo	
Expertise	Habilidades da Equipe	Equipe de desenvolvimento
	Pessoas	
	Gestão de Projetos	
-	Financiamento	Gestão de projetos
-	Processo de desenvolvimento	
-	Planejamento	
-	Programação/Agendamento	
-	Patrocínio/Propriedade	Relacionamento com cliente e usuário
-	Gestão de Relacionamentos	
Ambiente Organizacional	Ambiente Corporativo	Valor/importância atribuído ao projeto
-	Dependências externas	Relacionamento com o ambiente externo

**Fonte:** Elaborado pelo autor.

A **Tabela 8** foi construída seguindo o contexto do trabalho de cada pesquisador. Visitando a pesquisa de cada autor, foi possível comparar suas propostas de grupos de riscos. Cada autor atribuiu um nome e os riscos presentes em cada grupo, e a partir dessas informações foi gerada a comparação presente na tabela anterior. Tomando por base os dados apresentados na **Tabela 8** é proposta uma nova lista de grupos de fatores de risco. Essa nova proposta tem como foco a nomenclatura dos grupos de risco, fornecendo um padrão para facilitar o contexto de cada risco.

Segmentar os riscos em grupos foi importante para este trabalho por inserir os riscos em grupos de mesmo contexto e facilitar a construção do modelo. Os seis grupos propostos são<sup>9</sup>:

- **Gestão de Dependências:** relaciona-se com os riscos que envolvem qualquer tipo de dependência, seja direta ou indireta com o software ou o projeto. As dependências podem ser com relação a tecnologia, pessoas, fornecedor, outros sistemas etc;
- **Gestão de Escopos e Requisitos:** todos os riscos que envolvem os escopos e os requisitos do projeto serão tratados nesse grupo;
- **Gestão de Inovação e Tecnologias:** é o grupo destinado a lidar com as escolhas tomadas acerca das tecnologias e inovações;
- **Gestão de Pessoas:** são todos os fatores que envolverem pessoas no que tange conhecimentos, habilidades, capacidades, motivação etc;
- **Gestão de Projetos:** atributos comuns do gerenciamento como prazo, custo, esforço e outros, são inseridos nesse grupo;
- **Gestão de Relacionamentos:** aqui envolvem todos os riscos atribuídos ao relacionamento entre os stakeholders do projeto.

Anteriormente foram realizados o mapeamento e análise dos fatores de risco de maior impacto no desenvolvimento de software. Em seguida esses fatores foram agrupados por contexto. Os trinta e três fatores de risco foram distribuídos nos grupos anteriormente definidos. A seguir a tabela com os grupos e seus respectivos riscos associados.

---

<sup>9</sup> O nome *gestão* foi utilizado ao iniciar todos os grupos, para fazer referência ao ato de gerir os riscos envolvidos no desenvolvimento de software.

**Tabela 9. Agrupamento dos riscos por contexto**

---

**1. Gestão de Dependências**

---

- 1.1. Gerenciamento impróprio de mudanças.
- 1.2. Falha em gerenciar as expectativas finais dos usuários.
- 1.3. O sistema possui integração e interface com outros sistemas.
- 1.4. Mudança no proprietário do projeto ou na alta gerência.
- 1.5. Projeto com múltiplos fornecedores.
- 1.6. Sistemas complexos.

---

**2. Gestão de Escopo e Requisito**

---

- 2.1. Mudança de Escopo/objetivos.
- 2.2. Requisitos mal entendidos e/ou mal definidos.
- 2.3. Volatilidade nos requisitos (falta de requisitos estáticos).
- 2.4. Escopo/objetivos pouco claros ou equivocados.

---

**3. Gestão de Inovação e Tecnologias**

---

- 3.1. Adoção de novo método/tecnologia.
- 3.2. Assunto novo ou não familiar.
- 3.3. Usar nova metodologia de desenvolvimento em projetos importantes.
- 3.4. Falta de tecnologias maduras/existentes.
- 3.5. Deficiência de execução em tempo real.

---

**4. Gestão de Pessoas**

---

- 4.1. Pessoal envolvido insuficiente/inapropriado.
- 4.2. Falta de conhecimento/competência dos envolvidos no projeto.
- 4.3. Falta de habilidade para o gerenciamento de projetos.
- 4.4. Falta de motivação da equipe.
- 4.5. Rotatividade da equipe.

---

**5. Gestão de Projetos**

---

- 5.1. Prazos e tempo para tarefas mal estimados.
- 5.2. Custos mal estimados.
- 5.3. Planejamento inexistente ou inadequado.
- 5.4. Falta de metodologia efetiva em gerenciamento de projetos.
- 5.5. Controle pobre ou inexistente.
- 5.6. Definição imprópria de papéis e responsabilidades.
- 5.7. Tarefas complexas.

---

**6. Gestão de Relacionamentos**

---

- 6.1. Falta de envolvimento adequado dos usuários.
- 6.2. Falta de comprometimento da alta gerência.
- 6.3. Falta de Cooperação dos usuários.
- 6.4. Conflito entre departamentos de usuário.
- 6.5. Falha em obter comprometimento do cliente.
- 6.6. Falta de poderes para o gerenciamento de projetos.

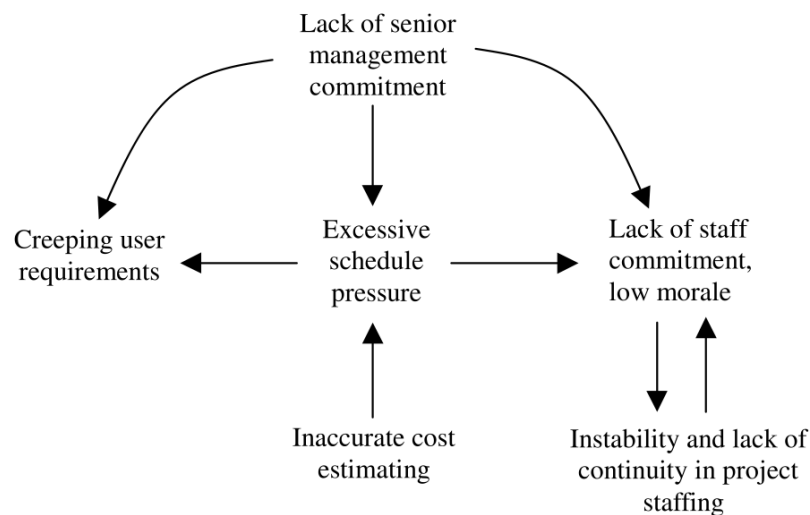
---

**Fonte:** Elaborado pelo autor.

Os riscos presentes em cada grupo estão organizados pela ordem de impacto. O primeiro item de cada grupo é o risco mais impactante daquele contexto e assim sucessivamente. Após a seleção das variáveis, o foco foi a construção do modelo.

### 3.2 Modelo de dinâmica de sistemas

Como apresentado, a identificação das variáveis fornece o ponto inicial para a modelagem, além disso é necessário conhecer os relacionamentos e efeitos das conexões (HUSTON et al. 2001). Para as conexões das variáveis, foram utilizados dados da literatura distribuídos em textos, tabelas e imagens. A seguir um diagrama de causalidade retirado do trabalho de Huston et al. (2001), nele são apresentadas as inferências.

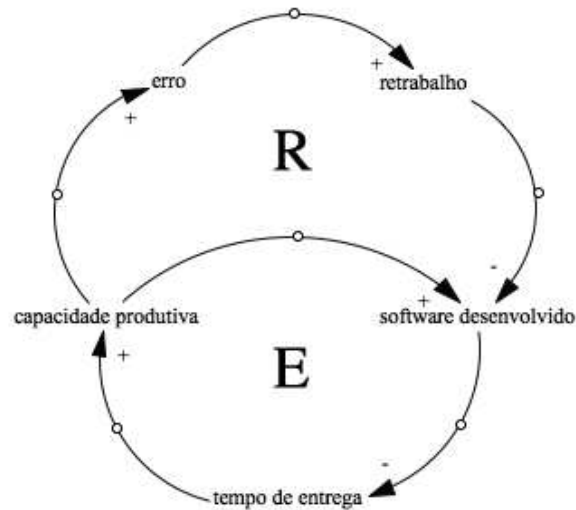


**Figura 12.** Possível diagrama de causalidade para as seis variáveis presentes.

**Fonte:** (HUSTON, 2001)

Como pode ser visto na **Figura 12**, não são apresentados os graus de influência, sinais de “+” e “-”, o autor foca no relacionamento entre os riscos. A partir das variáveis definidas no item anterior, foi possível construir diversos diagramas de causalidade. A construção desses diagramas, não seguiu uma organização cronológica, sendo assim, a construção do modelo de dinâmica de sistemas foi iterativa. A seguir um exemplo de um dos diagramas produzidos para

anexar ao modelo de DS em construção, esse diagrama foi construído a partir do estudo de Madachy (2007):



**Figura 13. Diagrama de causalidade simbolizando o retrabalho em produção de software.**

**Fonte:** Elaborado pelo autor.

Saindo dos detalhes da ferramenta e observando as relações entre os itens do diagrama, pode-se produzir várias inferências. Primeiramente, para facilitar o entendimento, cada variável considerada será explicada.

- **Capacidade produtiva:** é a capacidade de produzir o projeto, no caso, pode ser entendido como sendo a quantidade de pontos função que são produzidos;
- **Software desenvolvido:** é a quantidade de software que já foi produzido.
- **Tempo de entrega:** esse item se refere ao tempo gasto para desenvolver o projeto, ou seja, o tempo que é gasto para que o software esteja completamente desenvolvido.
- **Erro:** quantidade de erro produzido.
- **Retrabalho:** quantidade de retrabalho que é gerado a partir dos erros.

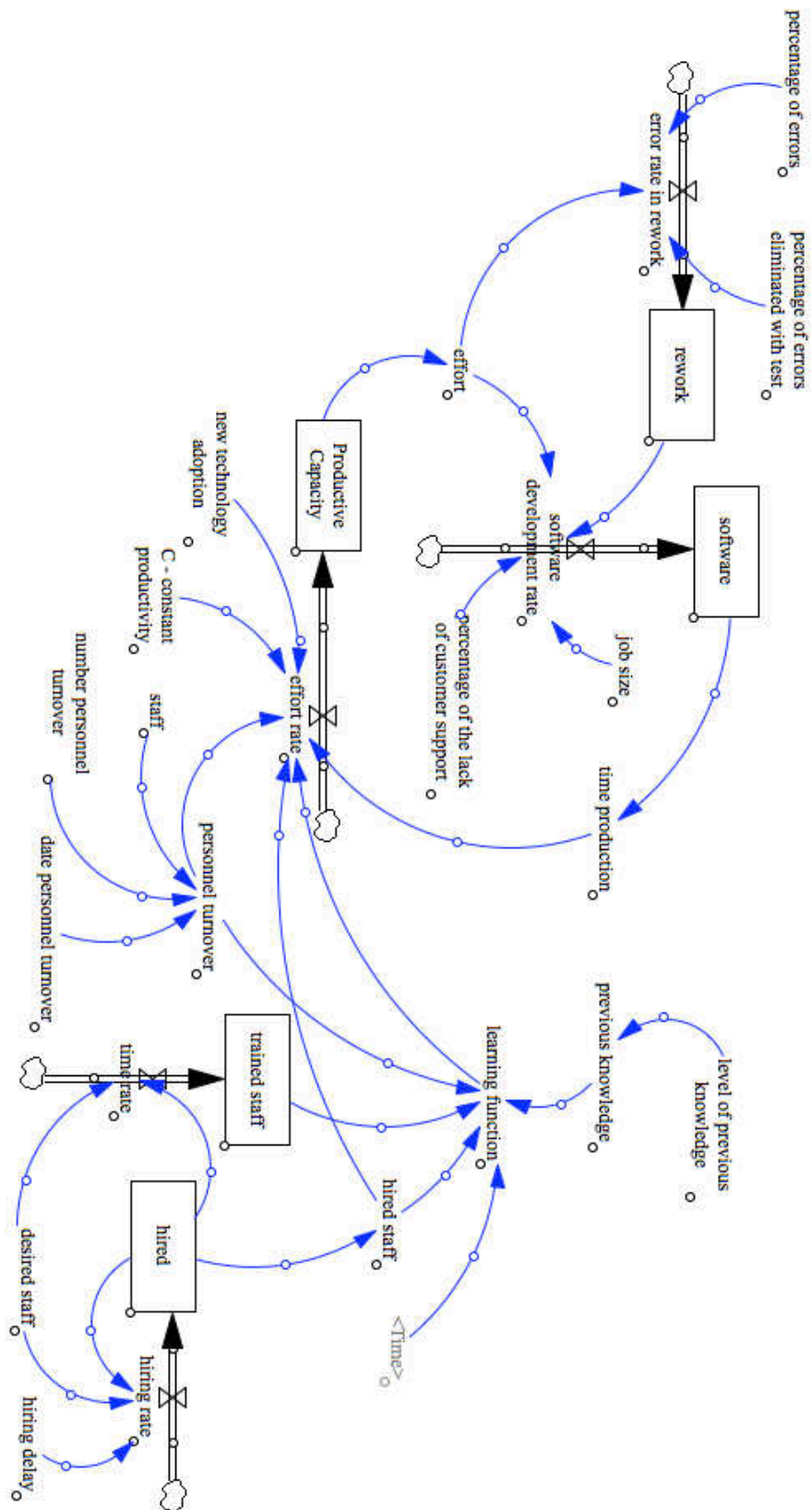
Pode ser iniciada uma leitura do diagrama a partir de qualquer item, por exemplo tomando como ponto inicial a *capacidade produtiva*, quanto mais código é produzido, tem-se mais *software desenvolvido*, e maior é a quantidade de *erro*

gerado, em consequência cresce também a quantidade de *retrabalho* que a equipe terá que realizar, e com isso, menos *software desenvolvido*, uma vez que retrabalho é refazer algo que a princípio já estava pronto. A quantidade de software produzido é afetada tanto pela *capacidade produtiva* quanto pelo *retrabalho*, e por sua vez, ele impacta o *tempo de entrega*, que significa o tempo necessário para entregar o projeto, ou seja, se temos muito software pronto, precisamos de menos tempo para entregá-lo, de outro modo, se o desenvolvimento não foi satisfatório o prazo de entrega cresce. O *tempo de entrega* afeta diretamente a *capacidade produtiva*, quanto mais tempo é gasto desenvolvendo algo, cresce também o conhecimento da equipe, tanto na tecnologia utilizada para desenvolver o projeto, quanto no projeto em si, com isso, a produção aumenta.

Uma nota importante, por mais que a produção aumente com o passar do tempo, esse fator não infere em um crescimento constante, a tendência é um aumento produtivo nos primeiros meses e posteriormente uma estabilidade, atingindo um estado de equilíbrio com o tempo. Esse equilíbrio pode ser quebrado aumentando a equipe, melhorando os treinamentos, técnicas motivacionais e outros.

Esses fatores podem ser testados a partir de simulações; diagramas de causalidade são convertidos em diagramas matemáticos que suportam simulações de cenários. Muitos sistemas são muito complexos para soluções analíticas fechadas, ou seja, planilhas, fluxogramas etc. A simulação é usada para exercitar modelos com entradas dadas para ver como o sistema se comporta (MADACHY, 2007).

Após a construção de inúmeros diagramas de sistemas e a conversão desses em um modelo de DS, a seguir o modelo principal deste trabalho.

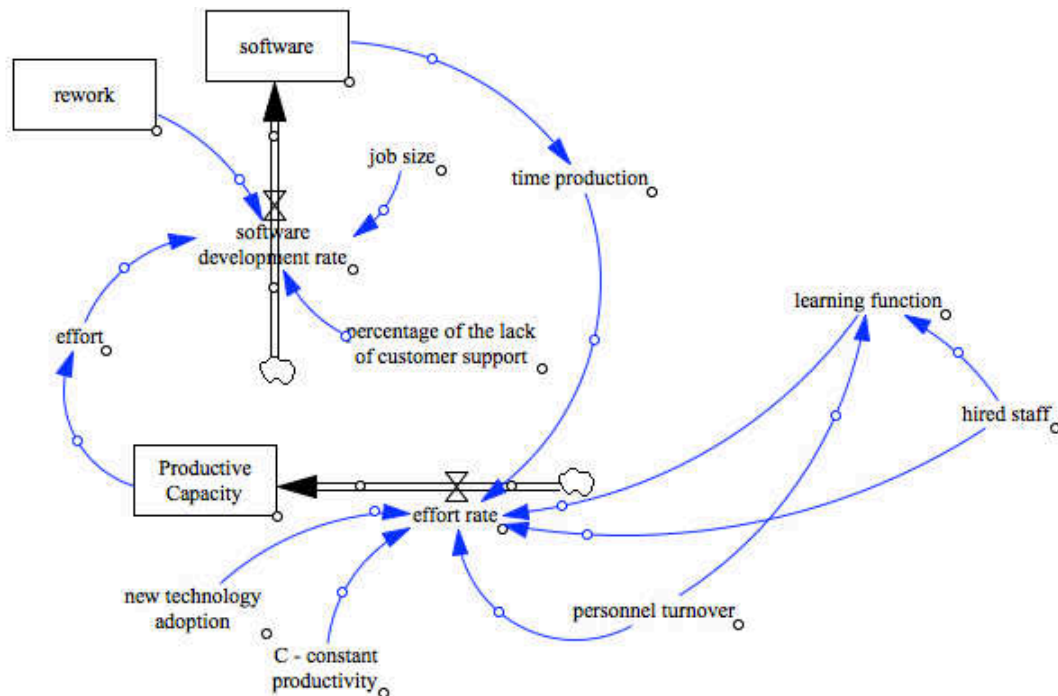


**Figura 14. Modelo de dinâmica de sistemas para avaliação de risco no desenvolvimento de software.**

**Fonte:** Elaborado pelo autor.

O modelo foi construído a partir dos contextos das variáveis de risco. Nesse sentido, um contexto pode ser representado por uma variável, pode ser um fluxo no tempo e até um conjunto de variáveis, como por exemplo o contexto de *gestão de pessoas* da **Tabela 9** é representado pela interação entre *personnel turnover* (rotatividade de pessoal), *learning function* (função de aprendizado) e as demais variáveis que interagem diretamente com essas.

Neste estudo o foco foi o impacto dos riscos na produtividade de software. Para facilitar o entendimento do modelo, a **Figura 14** foi segmentada em partes, as quais serão utilizadas para explicar detalhadamente o trabalho.



**Figura 15.** Segmento do modelo responsável por definir o esforço na produção de software.

**Fonte:** Elaborado pelo autor.

A **Figura 15** representa o “core” do trabalho. Nela podem ser vistos todos os contextos de risco se relacionando, a seguir a **Tabela 10** e uma lista dos contextos e as principais variáveis que os representam na **Figura 15**.

**Tabela 10. Correlação entre as variáveis dos grupos de risco e os da Figura 15**

<b>Variáveis Risco</b>	<b>Variáveis do Esforço de Produção</b>
<b>Gestão de Dependências</b>	
Gerenciamento impróprio de mudanças.	New technology adoption
Falha em gerenciar as expectativas finais dos usuários.	Percentage of the lack of the customer support
O sistema possui integração e interface com outros sistemas.	Personnel turnover
	Rework
<b>Gestão de Escopo e Requisito</b>	
Mudança de Escopo/objetivos.	
Requisitos mal entendidos e/ou mal definidos.	Effort
Volatilidade nos requisitos (falta de requisitos estáticos).	Job size
Escopo/objetivos pouco claros ou equivocados.	New technology adoption
	Rework
	Software
<b>Gestão de Inovação e Tecnologias</b>	
Adoção de novo método/tecnologia.	
Assunto novo ou não familiar.	
Usar nova metodologia de desenvolvimento em projetos importantes.	New technology adoption
Falta de tecnologias maduras/existentes.	Learning function
Deficiência de execução em tempo real.	
<b>Gestão de Pessoas</b>	
Pessoal envolvido insuficiente/inapropriado.	
Falta de conhecimento/competência dos envolvidos no projeto.	C - constant productivity
Falta de habilidade para o gerenciamento de projetos.	Effort rate
Falta de motivação da equipe.	Hired staff
Rotatividade da equipe.	Learning function
	Personnel turnover
<b>Gestão de Projetos</b>	
Prazos e tempo para tarefas mal estimados.	
Custos mal estimados.	Effort
Planejamento inexistente ou inadequado.	Effort rate
Falta de metodologia efetiva em gerenciamento de projetos.	Job size
Controle pobre ou inexistente.	Productive capacity
Definição imprópria de papéis e responsabilidades.	Software development rate
Tarefas complexas.	Time production
<b>Gestão de Relacionamentos</b>	
Falta de envolvimento adequado dos usuários.	Hired staff
Falta de comprometimento da alta gerência.	Percentage of the lack of the customer support
Falta de Cooperação dos usuários.	Personnel turnover
Conflito entre departamentos de usuário.	
Falha em obter comprometimento do cliente.	
Falta de poderes para o gerenciamento de projetos.	

**Fonte:** Elaborado pelo autor

- **Gestão de Dependências:** *rework (retrabalho), new technology adoption (adoção de nova tecnologia), percentage of the lack of the customer support (porcentagem da falta de suporte do cliente), personnel turnover (rotatividade da equipe)*;
- **Gestão de Escopos e Requisitos:** *job size (tamanho do projeto), rework (retrabalho), software (software), effort (esforço), new technology adoption (adoção de nova tecnologia)*;
- **Gestão de Inovação e Tecnologias:** *new technology adoption (adoção de nova tecnologia), learning function (função de aprendizado)*;
- **Gestão de Pessoas:** *personnel turnover (rotatividade de pessoal), C - constant productivity (constante de produtividade), hired staff (equipe contratada), learning function (função de aprendizado), effort rate (taxa de esforço)*;
- **Gestão de Projetos:** *effort rate (taxa de esforço), productive capacity (capacidade produtiva), effort (esforço), software development rate (taxa de desenvolvimento de software), job size (tamanho do projeto), time production (prazo de desenvolvimento)*;
- **Gestão de Relacionamentos:** *percentage of the lack of the customer support (porcentagem da falta de suporte do cliente), personnel turnover (rotatividade de pessoal), hired staff (equipe contratada)*.

Após a identificação dos contextos, o próximo passo é entender o segmento do modelo. O fluxo *effort rate* (taxa de esforço) armazena uma das principais equações, ele utiliza todas as variáveis de entrada para resultar na *productive capacity* (capacidade produtiva), esse estoque armazena a capacidade produtiva mensal, e a variável *effort* (esforço) armazena o esforço total até o momento. O fluxo *software development rate* (taxa de desenvolvimento de software) é responsável por calcular quanto do projeto já foi desenvolvido e descontar dessa produtividade o impacto do *rework* (retrabalho) e a *percentage of the lack of the customer support* (porcentagem da falta de apoio do cliente). O estoque *software* armazena a quantidade de software que já foi produzido, ou em outro ponto de vista, representa a quantidade de software que falta a ser produzido; enquanto houver software a ser construído, o *time*

*production* (tempo de produção) é incrementado e inserido na equação do fluxo *effort rate* (taxa de esforço) até a produção finalizar. A seguir serão apresentadas as principais equações presentes no segmento da **Figura 15**.

É comum autores estimarem produtividade a partir de uma equação linear, em que a produtividade é dada por uma relação de trabalho no tempo. É importante salientar que o mundo real não funciona com essa relação linear. Como a produtividade era um fator chave para o modelo, utilizou-se a função de produção, Cobb-Douglas, para estimar o esforço de produção de software, essa função compõe o fluxo *effort rate* (taxa de esforço) (PINDYCK et al. 2010). O formato da equação já foi apresentado, os dados da **Tabela 4** foram ajustados após inúmeras simulações, obtendo a seguinte equação:

$$y = 10,08 * (x)^{0,782} * (z)^{0,408} \quad (3)$$

onde y é o esforço de produção em horas ajustado no modelo para mês<sup>10</sup>, A (10,08), b (0,782) e c (0,408) são constantes especificadas que possuem valor positivo. Os fatores x e z são quaisquer medidas compostas por um conjunto de entrada ou uma única entrada cada uma (PENDHARKAR et al. 2008).

Para uma função Cobb-Douglas tradicional, considera-se x sendo trabalho e z capital, embora essa função aceite outras formas de variáveis (PINDYCK et al. 2010). No contexto deste estudo, x é trabalho e z é tamanho de projeto.

Tamanho de projeto não é um fator estático, mas sim uma equação do tipo SLIM-Putnam em sua forma padrão, definida por múltiplas variáveis (PUTNAM, 1978). Optou-se por essa abordagem, porque a equação Cobb-Douglas suporta funções em seus elementos, e assim, fortalecendo um comportamento não-linear e favorecendo uma maior aproximação com a realidade. Além disso, os elementos da equação a seguir estavam presentes na lista dos fatores de risco, confirmando a relevância desses.

$$S = C * K^{1/3} * t_d^{4/3} \quad (4)$$

---

<sup>10</sup> Os ajustes das constantes da Cobb-Douglas foram extraídos do trabalho de Pendharkar et al. (2008), nesse elas foram ajustadas para estimar a produtividade em hora. Neste trabalho optou-se por utilizar a distribuição do tempo em meses. Um homem possui aproximadamente cento e trinta horas mês (130h/mês) de produtividade (VAZQUEZ et al. 2009).

na equação (4),  $S$  é o tamanho de projeto,  $C$  é chamado de constante de produtividade,  $K$  é o esforço e  $t_d$  o tempo de desenvolvimento. A constante  $C$  de produtividade é uma constante estimada na SLIM-Putnam por fatores de linhas de código, a qual foi convertida para pontos função. Ela tem por objetivo ser um intensificador do trabalho dos desenvolvedores, possuindo uma variação entre 30 e 169, quanto maior a motivação e competência da equipe, maior o valor, caso contrário, menor o valor. A SLIM-Putnam é comumente utilizada isolando-se o esforço ( $K$ ), mas para este trabalho, necessitava-se do tamanho de projeto, por esse motivo utilizou-se a sua forma original (PUTNAM, 1978). Como a função Cobb-Douglas tem por objetivo calcular o esforço (produtividade), foi utilizada uma distribuição de recursos por tempo (VAZQUEZ et al. 2009), para substituir o esforço na SLIM-Putnam:

$$K = R * t_d \quad (5)$$

Na **Equação (5)** temos as seguintes variáveis:  $K$  significa esforço,  $R$  é recurso (i.e., tamanho de equipe) e  $t_d$  tempo de desenvolvimento (VAZQUEZ et al. 2009). Após essa definição foi substituído na equação de SLIM-Putnam, o esforço por: *tamanho de equipe vezes o tempo de desenvolvimento*, obtendo o seguinte resultado:

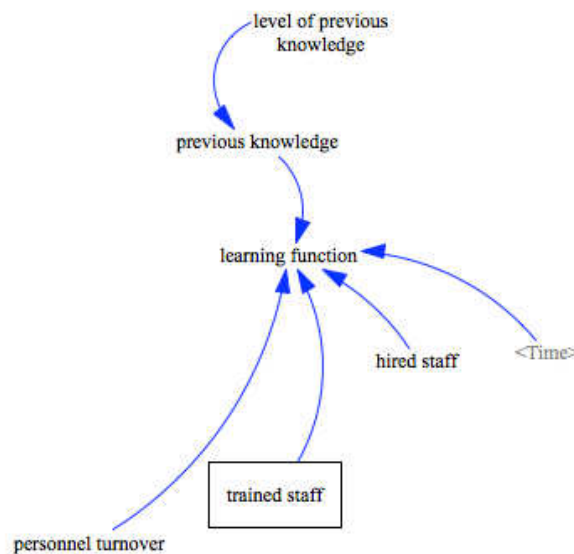
$$S = C * R^{1/3} * t_d^{5/3} \quad (6)$$

Definido o tamanho de projeto, ele foi inserido na equação Cobb-Douglas, substituindo  $z$  por  $S$ , tem-se a seguinte equação:

$$y = 10,08 * (x)^{0,782} * (C * R^{1/3} * t_d^{5/3})^{0,408} \quad (7)$$

Na pesquisa de Pendharkar et al. (2008) as variáveis utilizadas na sua função Cobb-Douglas, foram *tamanho de projeto* e *tamanho de equipe*. Durante o desenvolvimento deste trabalho, foram encontrados dados sobre o impacto da capacidade de aprendizado de uma equipe no desenvolvimento de software (COSTA, 2012; MADACHY, 2007). Assim, optou-se por adaptar a variável sugerida por Pendharkar et al. (2008), *tamanho de equipe*, com uma função de aprendizado.

Definiu-se *trabalho* como sendo a capacidade produtiva de uma equipe, levando em consideração a evolução desse time no decorrer do tempo (*função de aprendizado*).



**Figura 16. Segmento do modelo responsável por definir a função de aprendizado.**

**Fonte:** Elaborado pelo autor.

Na **Figura 16**, o contexto mais presente é o de *gestão de pessoas*, nela pode ser observado o relacionamento entre as variáveis de entrada da *learning function* (função de aprendizado).

Cada linguagem possui uma dificuldade inerente para ser utilizada no desenvolvimento de software. O modelo foi construído completamente levando em consideração uma estimativa em ponto função, o nível de dificuldade de uma linguagem pode definir o quanto de PF será produzido por um homem em um mês.

A correlação entre a dificuldade de uma linguagem e sua produtividade não é linear, i.e., diminuir o nível de dificuldade de uma linguagem pela metade não dobra a produtividade (JONES, 1996). Foi retirado do trabalho de Jones (1996), uma tabela comparativa entre o nível de dificuldade de uma linguagem e a produtividade média de um desenvolvedor usando essa linguagem em um mês. Distribuindo os dados dessa tabela em um gráfico, foi observado que o mesmo possuía o comportamento de uma curva de aprendizado.

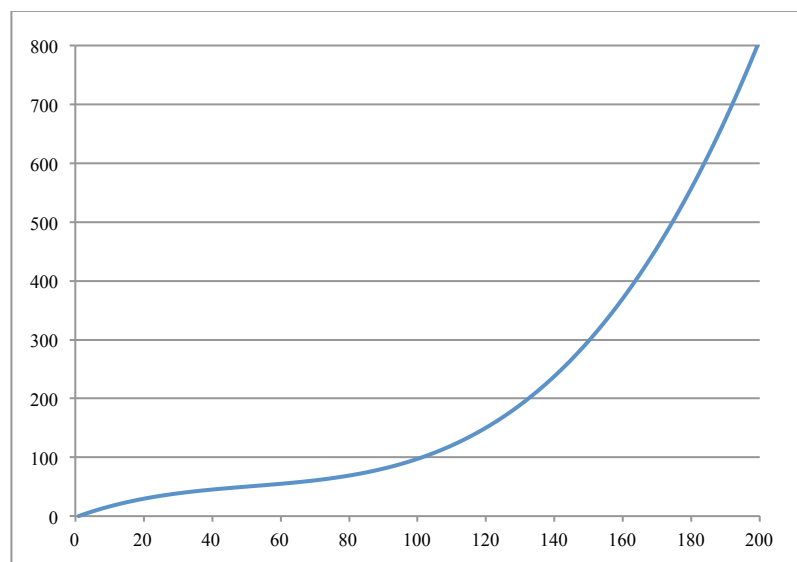
O uso de uma curva de aprendizado em dinâmica de sistemas é indicado, pelo fato de a mesma possuir um comportamento variante no tempo (MADACHY, 2007).

A partir da interpolação dos dados de Jones (1996), foi obtida uma função de aprendizado.

$$LF(w) = 0.1182 + 2.0828 * w - 0.032 * w^2 + 0.000211 * w^3 \quad (8)$$

Na equação (8), LF representa a curva de aprendizado, w uma combinação entre os fatores: <Time> (tempo), uma variável de sistema que marca o tempo (em meses) em execução e está alinhada diretamente com o *time production* (tempo de produção), e *previous knowledge* (conhecimento prévio) que é uma relação de *level of previous knowledge* (nível do conhecimento prévio) que varia de 0% à 100%.

O atributo *level of previous knowledge* (nível do conhecimento prévio) foi criado para facilitar a utilização do modelo. O gerente varia a porcentagem do nível de conhecimento entre 0% e 100%, assim, não precisa despendar tempo entendendo o cálculo realizado na variável auxiliar *previous knowledge* (conhecimento prévio), esses cálculos foram inseridos para ajustar o comportamento da função de aprendizado. A combinação entre a **Equação 7** e as variáveis de conhecimento prévio dos desenvolvedores, resultam na seguinte função de aprendizado:



**Fonte:** Elaborado pelo autor.

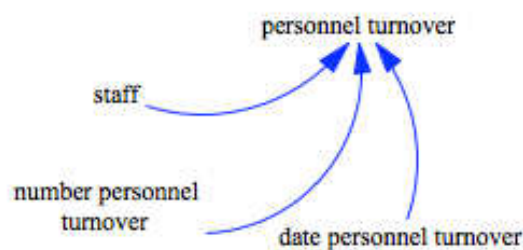
**Figura 17. Comportamento da função de aprendizado.**

O processo de aprendizado caracteriza-se por possuir um momento de crescimento nos primeiros instantes, depois um estado de estabilidade o qual produz

um sentimento de pouca ou nenhuma evolução, com a impressão de que o conhecimento posterior é menor do que o iniciado, mas após o período de estabilidade advém um alto desenvolvimento. A **Figura 17** representa o comportamento da *learning function* (função de aprendizado), ela simboliza como cada desenvolvedor evolui no sistema, ela interage diretamente com a quantidade de pessoas envolvidas, sendo as variáveis: *personnel turnover* (rotatividade de pessoal), *hired staff* (pessoal contratado) e *trained staff* (pessoal treinado).

Os dois próximos segmentos do modelo irão explicar o funcionamento da equipe envolvida, sendo que o primeiro irá definir *personnel turnover* (rotatividade de pessoal), e o segundo, *hired staff* (pessoal contratado) e *trained staff* (pessoal treinado).

A rotatividade de pessoal é um risco muito comum em projetos de software (SCHMIDT et al. 2009). A variável denominada *personnel turnover* (rotatividade de pessoal) é uma combinação de outras variáveis, ela determina se há ou não rotatividade na equipe, como pode ser visto a seguir.

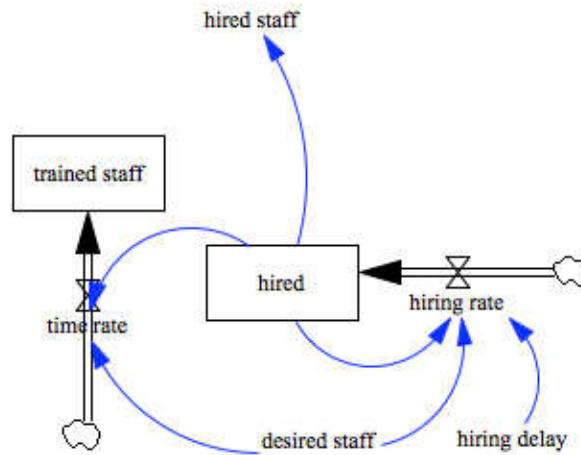


**Fonte:** Elaborado pelo autor.

### **Figura 18. Segmento do modelo responsável por definir a rotatividade de pessoal.**

Na **Figura 18** as variáveis *number personnel turnover* (número da rotatividade de pessoal) e *date personnel turnover* (data da rotatividade de pessoal), são responsáveis por modificar ou não o atributo *staff* (equipe). Caso não ocorra rotatividade a variável *personnel turnover* (rotatividade de pessoal) permanece com o mesmo valor da *staff* (equipe), caso contrário, o gerente define a quantidade de pessoas e o mês da saída; o mês da saída é definido informando quantos meses após o início do projeto ocorreu a rotatividade, ou seja, *date personnel turnover* (data da rotatividade de pessoal).

Uma vez que ocorre a rotatividade, é necessário contratar novos colaboradores. O processo de contratação descrito no modelo pode ser visto na **Figura 19**.



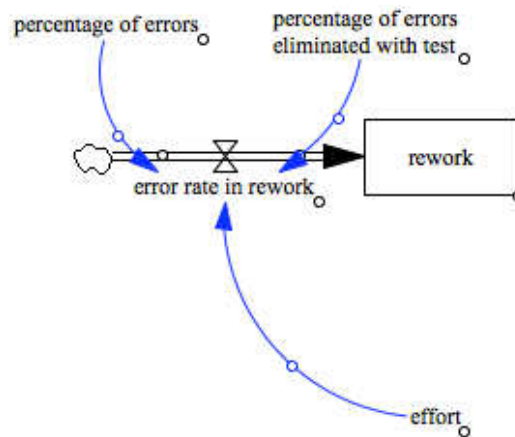
**Fonte:** Elaborado pelo autor.

**Figura 19. Segmento do modelo responsável por definir pessoal contratado.**

Para lidar com o processo de contratação, *hiring rate* (taxa de contratação), o gerente precisa definir a quantidade de pessoas que deseja contratar, simbolizado na variável *desired staff* (equipe desejada), além de ajustar o *hiring delay* (atraso na contratação) variável que define o tempo de atraso que ocorre na contratação. O tempo de demora varia de acordo com o tamanho da equipe que se deseja contratar, ele é um importante componente nos modelos de dinâmica de sistemas (MADACHY, 2007).

Após o estoque *hired* (contratados) atingir a quantidade desejada de contratação, ele define o atributo auxiliar *hired staff* (equipe contratada) com a quantidade do estoque, esse atributo fornece o tamanho da equipe contratada para a *learning function* (função de aprendizado), consecutivamente o fluxo *time rate* (taxa de tempo) é liberado para alimentar o estoque *trained staff* (equipe treinada), depois disso a nova equipe está pronta para atuar no projeto. Para completar o entendimento do modelo faltam alguns detalhes, um deles é o retrabalho.

Como já enunciado, o retrabalho ocorre para corrigir erros durante o desenvolvimento de software (MADACHY, 2007), sendo assim, quanto maior o projeto e a taxa de produtividade, maior será a tendência de retrabalho. A **Figura 20** representa esse segmento do modelo.



**Fonte:** Elaborado pelo autor.

**Figura 20. Segmento do modelo responsável por definir o retrabalho.**

As variáveis envolvidas na **Figura 20** definem a relação de produtividade, taxa de erro e teste. O atributo *effort* (esforço) armazena o total de esforço produtivo, ele alimenta o fluxo *error rate in rework* (taxa de erro em retrabalho) que utiliza as variáveis, *percentage of errors* (porcentagem de erros) e *percentage of errors eliminated with test* (porcentagem de erros eliminados com teste) para calcular o *rework* (retrabalho). O cálculo realizado no fluxo é simples, primeiro calcula-se a porcentagem de erro sobre a produtividade, com isso obtém-se a quantidade de erro gerada, retira-se desse valor os erros eliminados com testes, o restante é retrabalho.

Para finalizar a explicação do modelo, duas variáveis receberão uma atenção extra. As variáveis *percentage of the lack of customer support* (porcentagem da falta de suporte do cliente) e *new technology adoption* (adoção de nova tecnologia) foram inseridas no trabalho por terem grande impacto nos contextos apresentados e por fornecerem uma interação extra com os demais mecanismos do modelo. A primeira relacionada com os clientes, interage diretamente com o fluxo final do desenvolvimento do software e a segunda com a taxa de produção.

É possível alcançar um aumento de 5% a 35% na produtividade, com a utilização de novas tecnologias (GLASS, 2002). A partir dessa informação, foi inserida no modelo a variável *new technology adoption* (adoção de nova tecnologia). A aplicação desta variável tem por objetivo impactar a produtividade. Em um primeiro momento o conhecimento de uma nova tecnologia impactava apenas positivamente o modelo, mas após algumas adaptações, a variável foi ajustada para

impactar a produtividade entre 35% para menos e 35% para mais (o ajuste é feito entre 0,65 até 1,35), por padrão a variável fica inativa, ou seja, não afeta nem positivamente nem negativamente (valor padrão igual a um).

A falta de suporte dos clientes afeta diretamente o controle das metas (WAN et al. 2010). Essa variável não impacta diretamente a produtividade, mas sim, afeta a entrega do projeto; o cliente precisa participar ativamente do desenvolvimento do produto, tanto para fornecer as informações para o desenvolvimento quanto para aprovar o que está sendo feito. Como não foram encontrados na literatura valores estimados do impacto da falta de suporte do cliente, a variável *percentage of the lack of customer support* (porcentagem da falta de suporte do cliente) foi ajustada utilizando as simulações, ela varia de 0% a 100% e recebeu valores estimados para cada tamanho de projeto.

No próximo tópico serão apresentadas as simulações, seus ajustes e os resultados gerados a partir delas.

### 3.3 Ajuste do modelo e estimativas

Com a DS é possível simular o comportamento dinâmico dos riscos e realizar estimativas. Utilizando os contextos dos grupos de riscos listados neste trabalho, foram realizadas diversas análises utilizando os resultados das simulações. Para validar e produzir valores iniciais para as simulações, foram utilizados os dados da pesquisa de Pendharkar et al. (2008) introduzidos na **Tabela 4**, além dos dados fornecidos por uma micro empresa de software.

Com esses números em mãos foi possível validar os valores e ajustar o modelo. Na **Tabela 11** podem ser vistos os ajustes das variáveis do modelo de acordo com as informações já citadas, extraídas e adaptadas da literatura.

**Tabela 11. Ajuste geral do modelo baseado no tamanho do projeto**

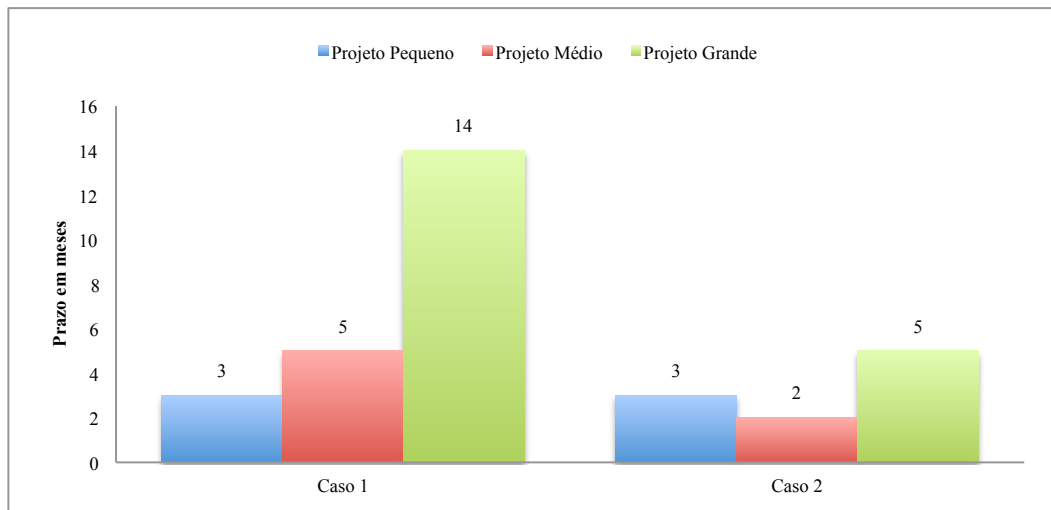
Variáveis	Tamanho Aproximado dos Projetos		
	Projeto 1	Projeto 2	Projeto 3
Tamanho do projeto	270 PF	520 PF	9800 PF
Tamanho da equipe	3 pessoas	8 pessoas	77 pessoas
Prazo de desenvolvimento	3 meses	5 meses	14 meses
Rotatividade	zero	zero	zero
C – Constante de produtividade	90	90	90
Equipe contratada	zero	zero	zero

Nível do conhecimento inicial da equipe	80	50	33
Adoção de nova tecnologia	1,0	1,0	1,0
Porcentagem de erro	35%	50%	60%
Porcentagem de erros eliminados com teste	20%	20%	20%
Retrabalho	28%	40%	48%
Porcentagem da falta de apoio do cliente	5%	10%	25%
Horas trabalhadas por homem-mês	130h	130h	130h

**Fonte:** Elaborado pelo autor

O usuário pode usar os dados da **Tabela 11** para direcionar as simulações com o modelo. A seguir serão apresentadas diversas simulações e resultados utilizando os dados apresentados.

A **Figura 21** apresenta dois casos. No primeiro caso o modelo foi alimentado com os dados da **Tabela 11**, respeitando a diferença de cada projeto. Como pode ser visto, cada projeto possui características diferentes, como por exemplo a *porcentagem de erros encontrados, porcentagem da falta de apoio do cliente* e outros. No segundo caso, foi simulado o comportamento do prazo, quando os projetos maiores (projeto 1 e projeto 2) são submetidos aos mesmos riscos do projeto 1. Foi mantido intacto somente o tamanho do projeto e tamanho da equipe. A intenção dessa abordagem é apresentar a individualidade de cada projeto, mostrando que os riscos de um projeto crescem com o seu tamanho, e isso influencia no prazo de desenvolvimento. Gerentes acostumados no desenvolvimento de projetos pequenos, podem acreditar que os riscos envolvidos nestes tipos de projetos são equivalentes à projetos médios e grandes, a **Figura 21** apresenta o comparativo entre os dois casos citados, mostrando que o tamanho dos projetos influenciam nos riscos envolvidos e que por sua vez estes riscos impactam o prazo de desenvolvimento.



**Fonte:** Elaborado pelo autor.

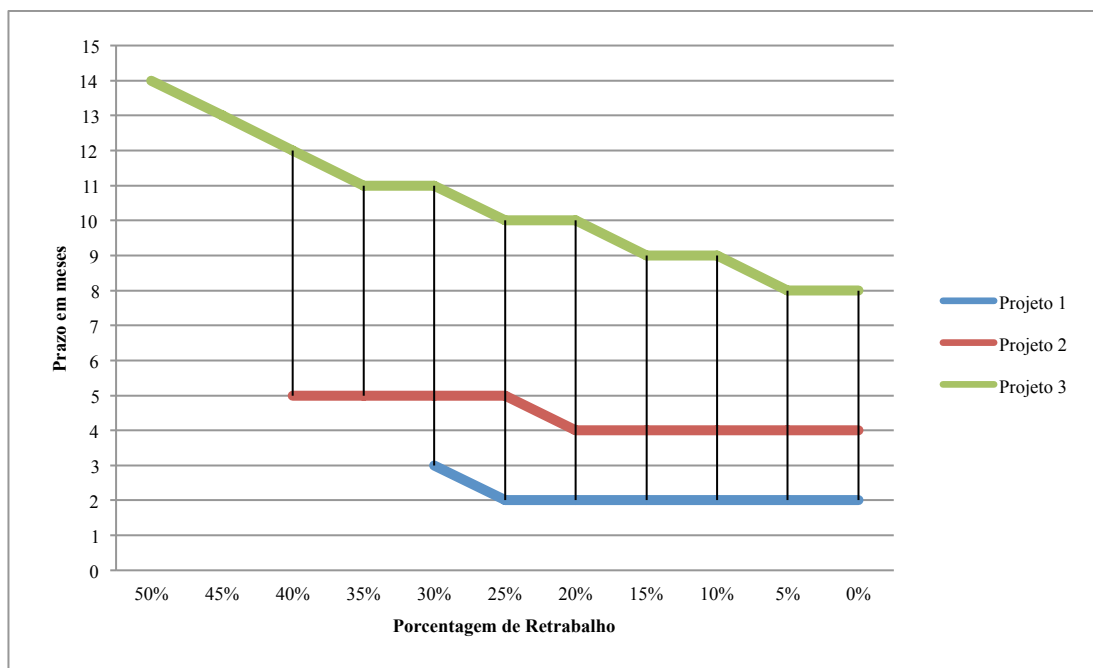
**Figura 21. Gráfico da variação do impacto dos riscos em projetos de tamanhos diferentes.**

O primeiro retângulo da esquerda para a direita representa um projeto pequeno, o segundo um projeto médio e o terceiro um projeto grande. Essa sequência é mantida para os dois exemplos. No Caso 1 as saídas respectivamente foram: três, cinco e quatorze meses; no Caso 2: três, dois e quatro. Com as informações apresentadas na **Figura 21** pode ser observado que se projetos de tamanhos diferentes tivessem a mesma quantificação de riscos, a dificuldade em executar um projeto grande ou pequeno seria praticamente a mesma.

Como já enunciado, algumas das variáveis apresentadas tiveram seus valores retirados da literatura, essas foram: *tamanho do projeto*, *tamanho da equipe*, *prazo de desenvolvimento*, *constante de produtividade*, *adoção de nova tecnologia* e *retrabalho*. As demais variáveis foram ajustadas com as simulações e utilizaram os seguintes parâmetros para os ajustes, a quantidade de pessoas necessárias (*tamanho da equipe*) para realizar um projeto de um determinado tamanho (*tamanho do projeto*) em um prazo determinado (*prazo de desenvolvimento*).

Cerca de 80% do sucesso do projeto está relacionado com 20% do esforço (seguindo a conhecida relação “regra de Pareto”), aproximadamente 80% dos erros estão em 20% dos módulos do projeto (Charette, 1989). Para lidar com os valores médios de retrabalho, sem diferenciar a qualidade da equipe que executa o projeto, definiu-se um valor padrão para a capacidade de correção de erros (i.e., 20%). Como o retrabalho ocorre para corrigir os erros que não foram capturados (MADACHY,

2007), foi observado que a quantidade de erros realmente cresce com o aumento do tamanho do projeto. O prazo de desenvolvimento está fortemente relacionado com o retrabalho, como pode ser visto na **Figura 22**.



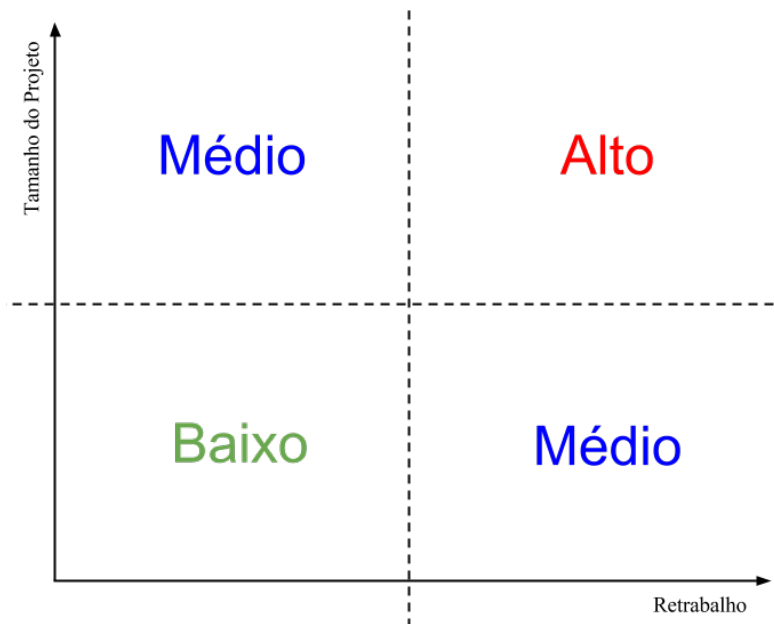
**Fonte:** Elaborado pelo autor.

**Figura 22. Gráfico da variação do impacto dos riscos em projetos de tamanhos diferentes.**

A curva do gráfico representa a queda no prazo de desenvolvimento com a redução da porcentagem de retrabalho. Para a construção da **Figura 22** foi utilizada a **Tabela 11**. Como pode ser visto o projeto de tamanho médio inicia-se em aproximadamente 40% e o projeto de tamanho pequeno em 30%.

A **Figura 22** apresenta o impacto do retrabalho em um projeto. De acordo com a literatura, em torno de 40% a 50% do esforço no desenvolvimento de software é gasto em retrabalho (SHULL, 2002). Esses dados mostram como o retrabalho está ligado ao desenvolvimento de software, por mais que a redução do retrabalho a zero não seja uma realidade. A **Figura 22** expõe como a redução desse fator pode diminuir o prazo de desenvolvimento. Outra observação sobre esta figura é que o impacto do retrabalho está ligado ao tamanho do projeto, em projetos maiores é necessário ter mais atenção com esse fator de risco, uma vez que reduzindo o retrabalho, os efeitos da diminuição do prazo serão mais perceptíveis, na **Figura 22**

pode ser visto que a redução do prazo no projeto 3 é seis vezes maior do que nos demais. A **Figura 23** apresenta o impacto desse risco no projeto.

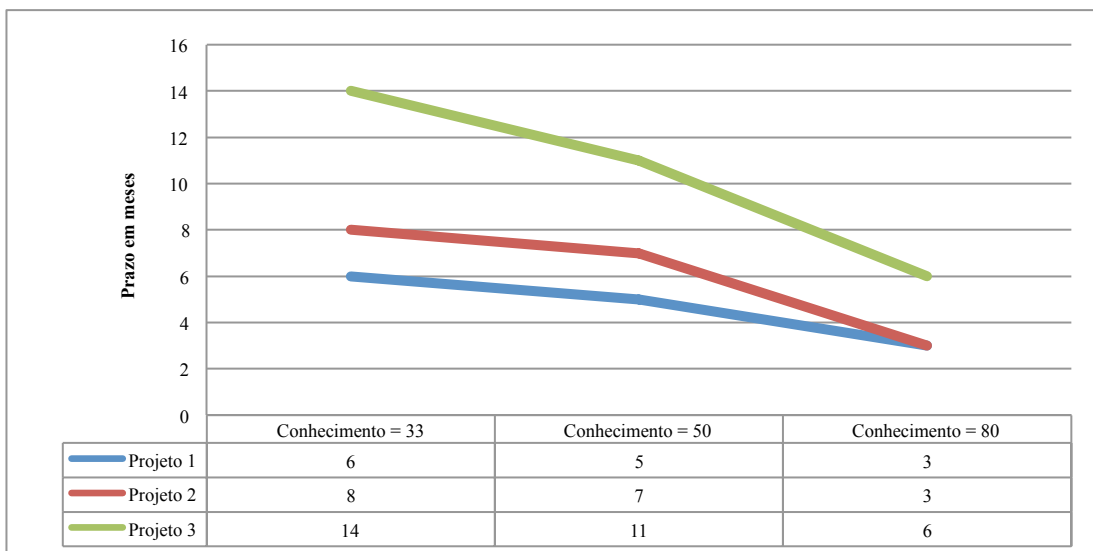


**Fonte:** Elaborado pelo autor.

**Figura 23. Risco de atraso do projeto baseado na relação tamanho do projeto e retrabalho.**

A relação retrabalho e tamanho do projeto é estritamente conectada. O risco de atrasos na entrega do projeto é intensificado com o tamanho, o retrabalho pode ser visualizado como um aumento no tamanho do projeto, uma vez que algo terá que ser refeito, com isso, o prazo de desenvolvimento é afetado. Outras duas variáveis muito importantes estimadas com as simulações foram: *conhecimento inicial da equipe*; *porcentagem da falta de apoio do cliente*.

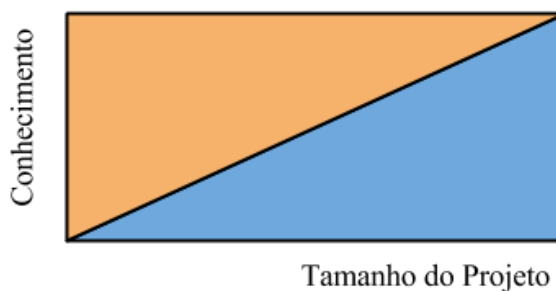
Quanto maior a equipe de trabalho, maior o impacto de um conhecimento inicial sobre o projeto; com as simulações foi observado que com um conhecimento médio acerca de um projeto amplo, pode-se reduzir em até 20% o prazo de desenvolvimento.



**Fonte:** Elaborado pelo autor.

**Figura 24. Redução do prazo de desenvolvimento de um projeto com o aumento do conhecimento.**

A **Figura 24** apresenta o impacto do aumento do conhecimento no prazo de desenvolvimento. Em projetos maiores o impacto da redução do prazo é maior, o desafio é obter um alto conhecimento prévio em grandes projetos. A dualidade entre conhecimento e o tamanho do projeto, pode ser vista como um dos principais motivos que dificulta essa otimização no prazo de desenvolvimento.



**Fonte:** Elaborado pelo autor.

**Figura 25. Conhecimento contra tamanho do projeto.**

A **Figura 25** deve ser lida variando o tamanho do projeto, quanto maior o tamanho, maior a dificuldade em ter um conhecimento prévio, de modo oposto, se o projeto for pequeno, mais fácil ter um conhecimento inicial. Embora essa dificuldade exista, cabe ao gerente de projeto desenvolver mecanismos para lidar com esse risco,

como por exemplo, usar históricos de projetos, desenvolvimento modular, gestão de conhecimento, relacionamento próximo com os clientes e outras soluções. O apoio do cliente é crucial no desenvolvimento de software, por essa razão, uma variável para representá-lo foi inserida no modelo.

Como já citado, o suporte do cliente (*porcentagem de apoio do cliente*) não aumenta a produtividade, mas reduz possíveis atrasos (WAN, 2010). Se em alguma etapa o cliente atrasar para confirmar algum requisito, o projeto é afetado. Durante a pesquisa foi capturado o risco da *falta de apoio do cliente*, o qual impactava apenas quando não havia o suporte dos clientes. Nos projetos maiores foi necessário diminuir a *porcentagem do apoio do cliente* nas simulações. Uma conclusão importante, não é que os clientes ficam menos solícitos em grandes projetos, mas que o número de processos que deverão ser aprovados aumenta, e com isso, aumenta o risco de o cliente falhar com o suporte necessário.

A partir desses ajustes e direcionamentos à tomada de decisão, foram realizadas algumas inferências com as simulações, como por exemplo, estimar o tamanho de uma equipe baseando-se no tamanho do projeto, levando em conta os fatores de risco. Após o modelo ter sido ajustado, foi possível estimar um tamanho médio de equipe por tamanho de projeto de software em ponto função. Com essa informação é fornecido ao gerente saber quantas pessoas ele necessita para realizar um projeto. Os valores estimados foram colocados em comparação com dados da literatura na **Tabela 12**.

**Tabela 12. Estimativa de tamanho de equipe baseado no tamanho do projeto.**

Projeto (PF)	Caso 1		Caso 2	
	Tamanho Equipe	Prazo	Tamanho Equipe	Prazo
Projeto 1 (270)	3 pessoas	3 meses	3 pessoas	3 meses
Projeto 2 (520)	8 pessoas	5 meses	6 pessoas	6 meses
Projeto 3 (9800)	77 pessoas	14 meses	108 pessoas	12 meses

**Fonte:** Elaborado pelo autor

A **Tabela 12** apresenta a comparação dos dados dos trabalhos empíricos da literatura com os dados estimados. Sendo (a) os dados reais e (b) os dados estimados. O tamanho da equipe no projeto 1 se manteve e com isso o prazo também ficou

igual, no projeto 2 houve uma redução de duas pessoas e um aumento de um mês no prazo de desenvolvimento e no projeto 3, houve um aumento de 31 pessoas na equipe e uma redução de 2 meses no desenvolvimento. A estimativa é baseada na produtividade média de uma pessoa sobre o tamanho do projeto.

O valor estimado são de 90 PF por pessoa. Esse valor foi estimado a partir da média entre a menor produtividade por pessoa, 520 PF divididos por 8 (65) e a maior produtividade, 9800 PF divididos por 77 (127) (PENDHARKAR et al. 20008). O valor obtido foi de 96, o qual foi ajustado para 90 PF utilizando as simulações. Foi utilizada a estimativa quantitativa em pontos função para facilitar a utilização, necessitando apenas que o projeto tenha o seu tamanho definido em pontos função.

A fórmula a seguir foi gerada a partir das simulações realizadas para este trabalho e apresenta um cálculo simples que o gerente pode fazer para estimar a equipe:

$$\text{tamanho equipe} = \text{tamanho projeto (PF)} / 90 \text{ (PF)} \quad (9)$$

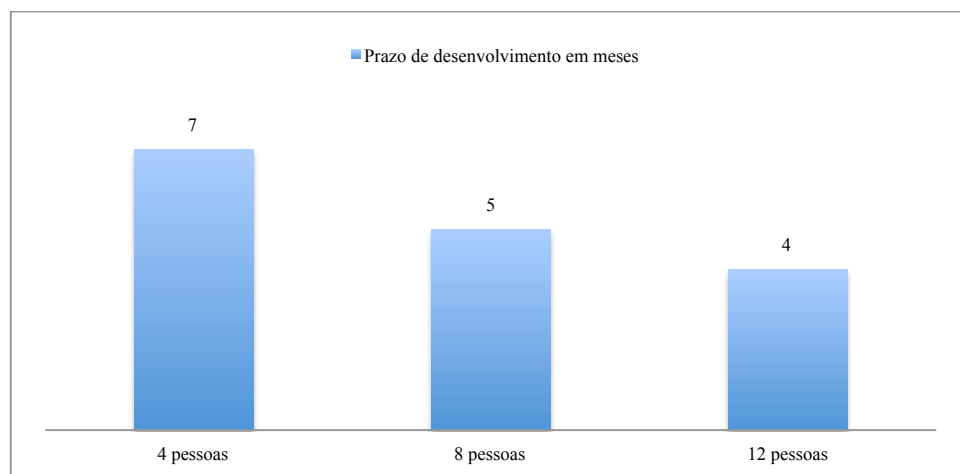
Essa fórmula pode ser usada para estimar o tamanho da equipe quando o gerente não possui um histórico de desenvolvimento, no caso de haver histórico, o ideal é seguir a capacidade produtiva da equipe. O erro da fórmula cresce com o aumento de tamanho do projeto, como pode ser observado na **Tabela 12**. Cada equipe e projeto possuem características intrínsecas, mas isso não impede a utilização de estimativas que auxiliam às tomadas de decisão. Cabe ao gerente de projetos avaliar quais estimativas fazem sentido para a sua realidade.

## 4 SIMULAÇÕES DO MODELO

Como já apresentado a tomada de decisão é envolvida por problemas classificados em: estruturados, semi-estruturados e não-estruturados (desestruturados). Cabe ao gerente de projetos utilizar ferramentas de apoio à decisão, conhecimentos e históricos para lidar com os desafios existentes no gerenciamento de projetos.

Nesta parte do trabalho, serão apresentados cenários de desafios que o gerente de projetos de software pode enfrentar. Foram realizadas diversas simulações e comparações, a fim de que esses cenários auxiliem a visão do gerente acerca dos projetos. Foi utilizado como cenário base, o projeto de médio porte (i.e., 520 PF) descrito na **Tabela 10**, e levou-se em consideração todos os valores dos atributos listados na tabela para o projeto desse porte.

Um desafio comum de um gerente de projetos é definir o tamanho da equipe à sua disposição. Foram geradas simulações com uma variação no tamanho da equipe em 50%, a **Figura 26** apresenta os resultados obtidos no prazo de desenvolvimento.



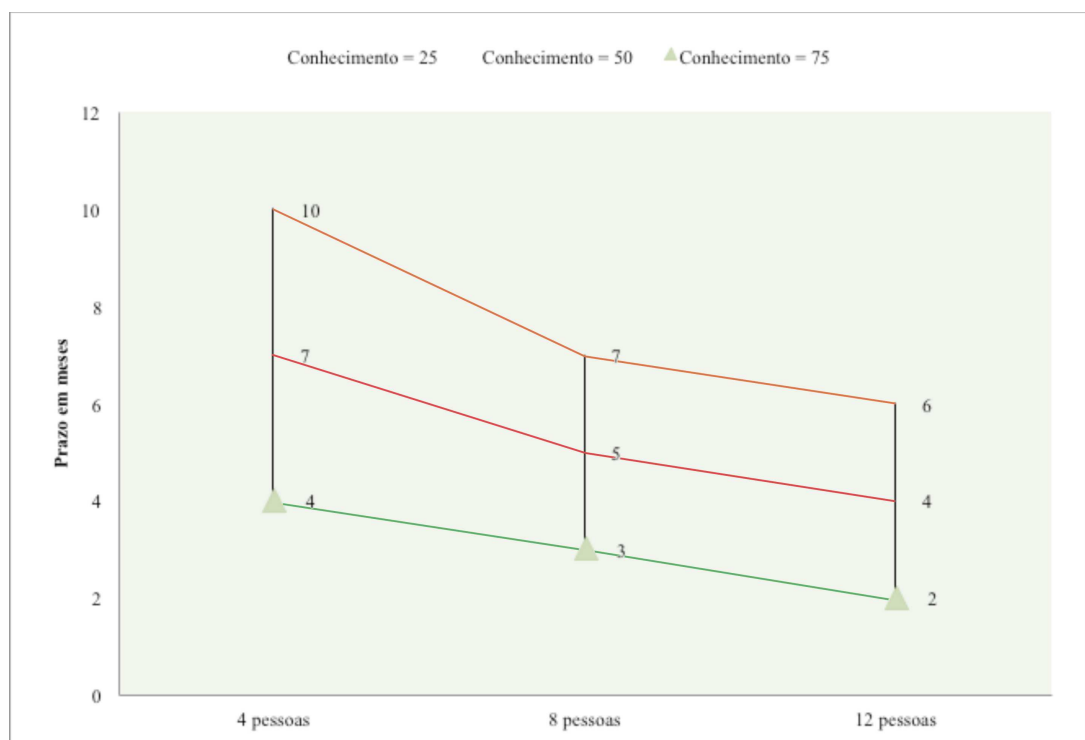
**Fonte:** Elaborado pelo autor.

**Figura 26. Impacto do tamanho da equipe no prazo de desenvolvimento de um projeto de 520 PF.**

Na **Figura 26** pode ser visto a variação no prazo provocado a partir da alteração do tamanho de uma equipe em um projeto médio. Os dados foram gerados a partir de simulações. A **Figura 26** apresenta uma visão do prazo de

desenvolvimento considerando equipes com capacidades produtivas idênticas. Para os três cenários apresentados anteriormente, o único fator que foi alterado nas simulações de um para outro, foi o tamanho da equipe, todos os demais dados foram mantidos. Dessa forma, é possível comparar o efeito de alterar o tamanho da equipe. Com as simulações, foi observado que reduzir a equipe em 50% gerou mais impacto do que aumentá-la na mesma proporção, essa diferença será discutida juntamente com os dados da **Figura 27**.

No cenário anterior as equipes possuíam um conhecimento médio sobre a linguagem e o projeto. Na **Figura 27** é exposta uma comparação entre os tamanhos das equipes variando também o nível de conhecimento. Considerando o conhecimento médio em cinquenta, ele teve uma variação de 50% para mais e para menos.



**Fonte:** Elaborado pelo autor.

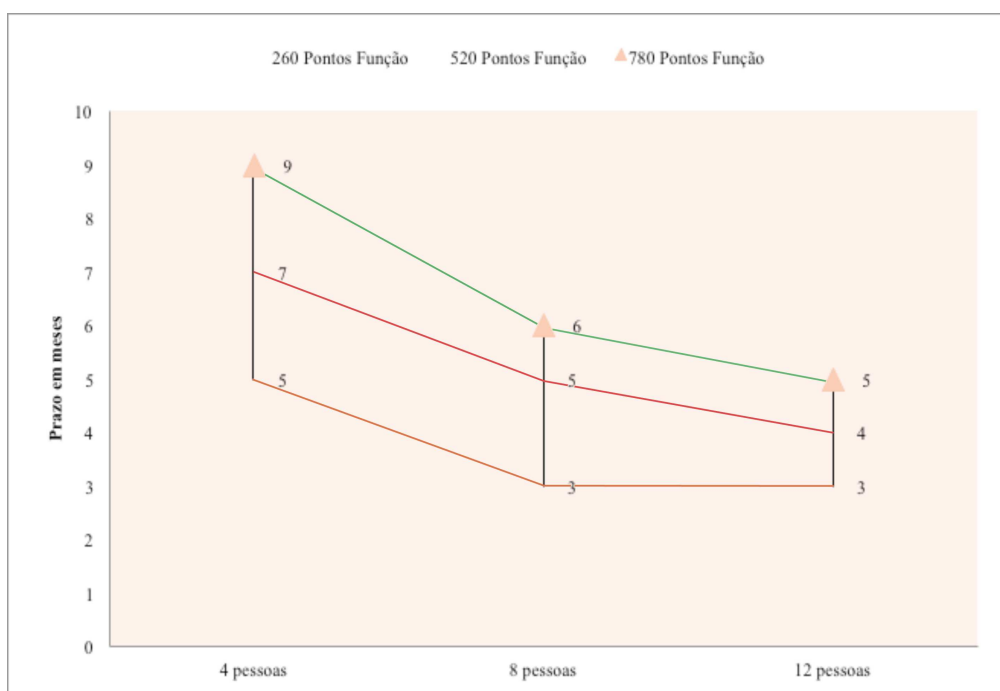
**Figura 27. Impacto da variação de conhecimento no prazo de desenvolvimento em um projeto de 520 PF.**

Na **Figura 27** pode ser observado que o conhecimento gera maior resultado na menor equipe, a diferença entre o pior caso e o melhor nessa equipe é de seis meses.

Uma equipe menor bem treinada consegue ter um rendimento melhor ou igual a uma equipe de tamanho três vezes maior com conhecimento baixo ou médio.

O impacto do conhecimento não é linear, ou seja, ele não decresce ou aumenta na mesma proporção nos três casos. Costa (2012) apresenta uma discussão sobre a relação do tamanho da equipe e produtividade, quanto maior o número de pessoas, a comunicação interna acaba sendo reduzida e os conflitos aumentados. Esses são alguns dos fatores que não permitem que a expressão do conhecimento dos desenvolvedores sejam iguais, independente da quantidade de pessoas envolvidas.

Na próxima análise o tamanho do projeto será o foco, foi mantida a variação do tamanho da equipe em 50% para mais e menos, nas simulações o conhecimento inicial foi deixado médio para todas as equipes. Na **Figura 28** a seguir é apresentado como o prazo de desenvolvimento de cada equipe variou reduzindo e aumentando o tamanho do projeto em 50%.



**Fonte:** Elaborado pelo autor.

**Figura 28. Impacto da variação do tamanho do projeto no prazo de desenvolvimento em um projeto de 520 PF.**

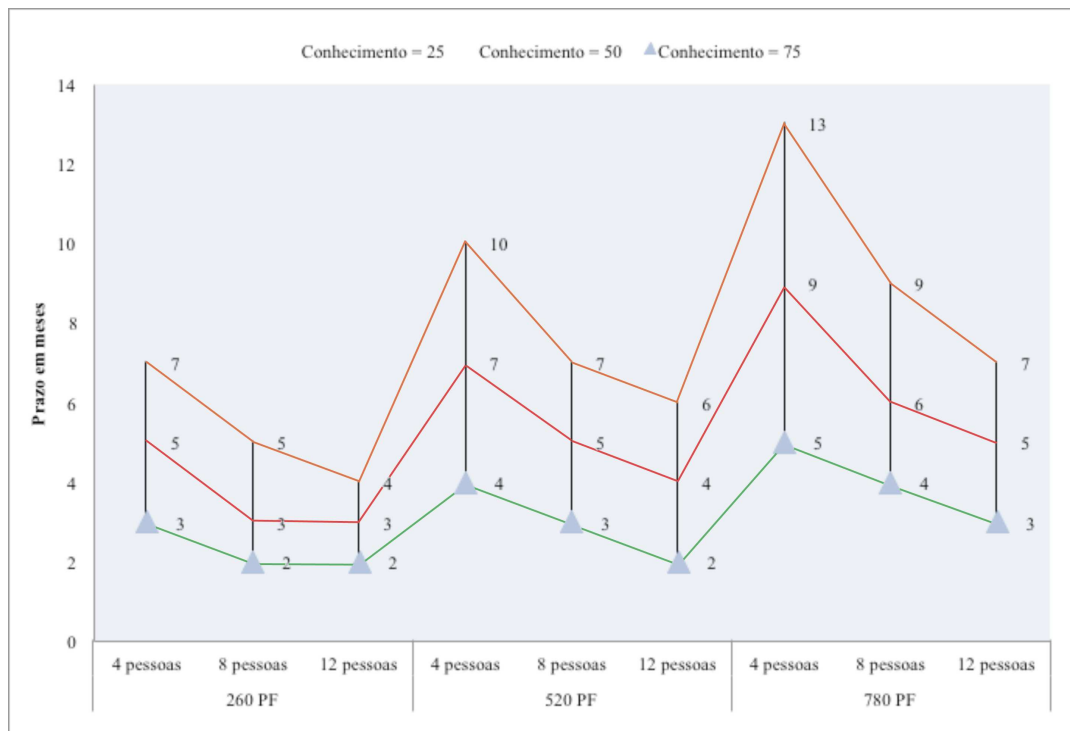
A **Figura 28** gera alguns questionamentos. Avaliando o comportamento do gráfico, fica nítido o impacto do tamanho do projeto em cada equipe, o questionamento gira em torno do motivo de o comportamento ser diferente. Nos

projetos de quatro e oito pessoas, a diferença entre o projeto menor com o médio é de dois meses e para a equipe de doze pessoas, a diferença é de apenas um mês. Nas equações utilizadas para realizar as simulações, quanto mais tempo uma equipe passa desenvolvendo um projeto, melhor ela fica, uma vez que o conhecimento aumenta com o passar do tempo, projetos maiores dão mais tempo para a equipe melhorar e aumentar o alinhamento entre seus integrantes.

Permanecendo na **Figura 28**, foi levantada no parágrafo anterior a questão da equipe de doze pessoas reduzir apenas um mês do projeto médio para o menor, e quando compara-se o prazo de desenvolvimento dela com a equipe de oito desenvolvedores, o prazo do projeto de 260 PF é igual.

As simulações deixam claro que aumentar o tamanho da equipe nem sempre é uma vantagem de produção. Sendo assim, o gerente precisa tomar cuidado para não colocar menos pessoas que o necessário para realizar um projeto e nem ultrapassar demais o valor ideal. Em um dos casos da **Figura 28** isso ocorreu, doze desenvolvedores para realizar um projeto de duzentos e sessenta pontos função, mostrou-se uma decisão ruim, o número de processos concorrentes em projetos pequenos é pequeno, isso faz com que em certos momentos desenvolvedores fiquem ociosos.

Na **Figura 29** pode ser visto o impacto da variação do conhecimento prévio (de vinte e cinco à setenta e cinco) dos desenvolvedores em cada tamanho de projeto de software, os dados foram gerados com simulações, cada indicador do gráfico representa o impacto da variação do conhecimento no prazo.



Fonte: Elaborado pelo autor.

**Figura 29. Impacto gerado no prazo, a partir da variação do conhecimento da equipe em projetos de tamanhos diferentes.**

A partir dessa figura é possível abstrair uma série de informações, analisando cada um dos três tamanhos de projetos. Nos próximos parágrafos, serão apresentadas possíveis análises que o gerente de projetos pode realizar.

Primeiro e já citado anteriormente no parágrafo de discussão da **Figura 28**. Nele questionou-se o fato de um grupo de 8 e 12 desenvolvedores obterem o mesmo desempenho para um projeto de 260 PF. Observando a **Figura 29** no segmento de 260 PF, pode ser visto que essas equipes continuam obtendo resultados semelhantes mesmo quando aumentamos o conhecimento de ambas, isso ocorre porque em projetos pequenos o número de processos concorrentes é menor, com isso, aumentar o tamanho da equipe não é uma decisão interessante, ter uma equipe média bem treinada neste caso pode ser uma decisão melhor.

No segundo segmento (projeto de tamanho 520PF), pode ser observado o prazo de desenvolvimento de uma equipe pequena (4 pessoas) e uma equipe grande (12 pessoas), um time de desenvolvimento pequeno e muito bem treinado, consegue ter um desempenho semelhante a uma equipe grande com conhecimento médio. Quanto menor a equipe mais fácil de ser treinada e gerenciada, mas a saída de um

desenvolvedor (turnover) possui maior impacto negativo, treinar e gerenciar uma equipe maior não é tão fácil, mas a vantagem neste caso é o menor impacto negativo da saída de algum profissional. Nesta situação, o gerente precisa optar entre administrar a motivação de uma equipe menor, ou o desafio de gerenciar uma grande equipe.

O terceiro e último tamanho de projeto (720 PF), pode ser analisado no contexto da variação do tamanho da equipe no menor conhecimento (conhecimento 25). Projetos grandes, possuem um número grande de processos paralelos, uma equipe pequena (4 pessoas) e pouco treinada, pode chegar a ter quase metade do rendimento de uma equipe grande (12 pessoas) nas mesmas condições. Neste caso utilizar um número reduzido de desenvolvedores não é uma decisão ótima, uma vez que além do baixo rendimento, a saída de um profissional possui grandes efeitos negativos.

Observando os dados apresentados, pode-se concluir que uma equipe pequena competente e bem treinada, em projetos pequenos e médios, consegue obter resultados satisfatórios quando comparados com uma equipe três vezes maior e de mesma competência, ou seja, ter uma equipe menor e qualificada mostrou-se nos casos pequeno e médio uma boa decisão, já em um projeto maior não se constitui uma decisão ótima. Foram apresentadas apenas uma análise em cada caso, o gerente de projetos pode abstrair outras análises ou utilizar o modelo para simular diversos outros cenários.

## 5 CONCLUSÃO

Esta pesquisa apresenta uma proposta de modelagem de processos com foco na gestão de risco e procura justificar, com base em dados e informações obtidas na literatura, a relação entre as variáveis presentes no modelo e os resultados das simulações. Nesse quesito, o trabalho está fortemente amparado pela pesquisa bibliográfica de trabalhos de autores de diversas áreas correlatas.

A pesquisa foi elaborada com o a aplicação do método de modelagem conhecido como, dinâmica de sistemas, o qual é útil para problemas que envolvem fatores intangíveis. Além disso, este método possibilita a construção de “loops” e “feedbacks”, estruturas recorrentes em modelagens de sistemas complexos. O impacto do uso de simulações no apoio à tomada de decisão é indiscutível e a dinâmica de sistemas é uma poderosa ferramenta que pode ser usada para essa finalidade.

Uma das dificuldades do estudo foi obter fontes para quantificar todas as variáveis, algumas foram ajustadas com as simulações. Como ponto positivo, foram encontradas fontes para validar as principais variáveis e resultados da pesquisa.

O trabalho oferece uma compilação de riscos que impactam o desenvolvimento de software. Os trinta e três fatores de risco agrupados em seis conjuntos, formam uma base de conhecimento para o gerente de projetos, que pode usar dessa ferramenta para auxiliar em suas decisões. Além dessa base é apresentada uma tabela comparativa entre projetos de pequeno, médio e grande porte, e, resultados gerados a partir de simulações para fornecer aos gerentes de projetos conhecimentos para lhes auxiliar nas tomadas de decisão.

No gerenciamento de risco é desejado diminuir as incertezas. Esse processo pode ser elaborado com um grande planejamento de ações para lidar com cada situação adversa, ou pode ser algo simples, como por exemplo, ter conhecimento dos problemas que podem atrapalhar o fluxo desejado do projeto. Gerenciar risco não é uma tarefa que deve ser executada apenas em grandes projetos, mas sim uma atividade que deve ser incorporada ao cotidiano do gerenciamento de projetos.

A área de atuação do modelo transcende a aplicação empresarial. A construção do aplicativo produzido com a pesquisa produzida, melhor descrito no Apêndice B, fornece mais que um suporte à gerentes de projetos, mas também uma fonte valiosa de ensino para disciplinas específicas.

O objetivo principal deste trabalho era produzir um modelo de dinâmica de sistemas com os riscos que impactam o desenvolvimento de software. Além desse objetivo, o trabalho atendeu todos os outros propostos. Trinta e três variáveis de risco foram levantadas a partir da literatura, essas foram organizadas em grupos de risco por contextos e posteriormente transformadas em um modelo de estoque fluxo. O modelo apresentado na pesquisa pode ser usado como suporte a decisão, fornecendo ao gerente de projetos a possibilidade de inserir entradas e analisar o impacto dessas entradas no prazo de desenvolvimento do projeto. Como resultado extra, o modelo foi adaptado para um aplicativo para celular, que fornece uma maior facilidade para a utilização tanto por profissionais quanto por estudantes.

Não é esperado que este estudo contemple todo o arcabouço de gerenciamento de risco em projetos de software, nem tão pouco que seja uma fonte singular e única nessa área; risco é incerteza, como diversas vezes foi citado aqui, seria inocente acreditar que os problemas do gerenciamento de risco se encerrariam com esta pesquisa.

## **5.1 Trabalhos futuros**

O modelo construído não abrange todos os fatores envolvidos no processo de risco. Uma possível extensão para esse trabalho é a realização de novos refinamentos no modelo com o objetivo de incluir novas variáveis e relacionamentos, oferecendo suporte a novos cenários. Também pode ser construído um modelo integrado com os trabalhos já realizados em dinâmica de sistemas no grupo do LES da UFV, para obter um modelo mais completo e robusto, abordando várias atividades do processo de desenvolvimento de software, como fonte pode ser usado os trabalhos de Coelho (2013), Costa (2012), Ambrósio (2011) e Hermsdorf (2011). Esta pesquisa fornece informações valiosas sobre produtividade e pode ser integrada aos demais trabalhos.

## APÊNDICE A

### 5.2 Equações do modelo

(1) effort rate =

$$\frac{((10.08 * (\text{learning function})^{0.782}) * ("C - \text{constant productivity}" * ((\text{personnel turnover} + \text{hired staff})^{1/3}) * (((\text{time production} + 1))^{5/3}))^{0.408}) * \text{new technology adoption}}{130}$$

**Fonte:** (PENDHARKAR et al. 2008); (PINDYCK, 2010); (PUTNAM, 1978); (GLASS, 2002).

(2) productive capacity =  
effort rate

(3) effort =  
Productive Capacity

(4) software development rate =  
IF THEN ELSE( ((job size+rework)\*(1+percentage of the lack of customer support))-effort >= 0 , ((job size+rework)\*(1+(percentage of the lack of customer support/100)))-effort , 0 )

**Fonte:** (MADACHY, 2007); (WAN et al. 2010); (SHULL et al. 2002)

(5) software =  
software development rate

(6) time production =  
IF THEN ELSE( software > 0 , 1 , 0 )

(7) error rate in rework =  
(percentage of errors\*effort)\*(1-percentage of errors eliminated with test)

**Fonte:** (MADACHY, 2007); (SHULL et al. 2002)

(8) rework =

error rate in rework

**Fonte:** (MADACHY, 2007); (SHULL et al. 2002)

(9) personnel turnover =

DELAY FIXED(staff-number personnel turnover , date personnel turnover ,  
staff)

(10) time rate =

IF THEN ELSE(hired < desired staff , 0 , 1)

**Fonte:** (MADACHY, 2007)

(11) hiring rate =

INTEGER( ((desired staff-hired)/hiring delay) )

**Fonte:** (MADACHY, 2007)

(12) trained staff =

time rate

**Fonte:** (MADACHY, 2007)

(13) hired =

hiring rate

**Fonte:** (MADACHY, 2007)

(14) hired staff =

INTEGER(hired)

**Fonte:** (MADACHY, 2007)

(15) learning function =

(((0.1182 + 2.0828\*(Time+previous knowledge) - 0.032\*((Time+previous  
knowledge)^2) + 0.000211\*((Time+previous knowledge)^3))\*personnel  
turnover) + IF THEN ELSE( trained staff > 0 , (((0.1182+2.0828\*(trained  
staff+previous knowledge)-0.032\*((trained staff+previous  
knowledge)^2)+0.000211\*((trained staff + previous knowledge)^3))\*hired  
staff) , 0 )

**Fonte:** (MADACHY, 2007); (JONES, 1996)

(16) previous knowledge =  
level of previous knowledge<sup>1.16</sup>

**Fonte:** (MADACHY, 2007); (JONES, 1996)

## APÊNDICE B

Este trabalho é construído por diversas peças com uma finalidade em comum, apoio à decisão. A verdadeira contribuição dessa pesquisa não está em dinâmica de sistemas, risco, economia etc. Este estudo foi elaborado para fornecer um suporte extra ao trabalho de gerenciamento de projetos e se de algum modo ele fornecer uma percepção extra ao gerente de projetos, e essa percepção o auxiliar em sua tomada de decisão, esta pesquisa terá atendido à sua finalidade.

Simular diversos cenários e analisar os seus efeitos, não trazem experiência, mas fornecem conhecimento que auxilia na construção do conhecimento empírico quando aplicado no mundo real. Por sua vez, aplicar aprendizado gera vivência que produz experiência. Isso mostra que essas variáveis não estão desconectadas, mas são pertencentes a um mesmo sistema, que é compostos por diversas outras variáveis que estruturam a complexidade do aprendizado.

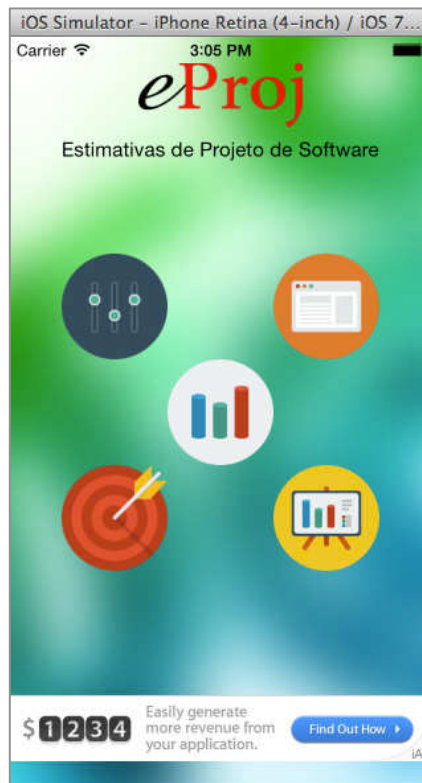
Quanto maior a quantidade de informação que uma pessoa possui acerca de alguma coisa, menor é o risco de algo não ocorrer como o esperado. Não é tratado aqui de uma ciência exata, em que a soma de dois fatores sempre terá um resultado conhecido e almejado, mas sim uma ferramenta que auxilie no processo de construção do conhecimento e fornecimento de informação. Pensando nessas ideias, ao final do desenvolvimento resolveu-se transformar parte dos resultados da pesquisa em um aplicativo para celular, o que não era um objetivo inicial<sup>11</sup>.

### **5.3 Aplicativo de auxílio à estimativa de software**

Neste tópico será apresentado a criação e funcionalidades do aplicativo de estimativa de projetos. O aplicativo foi nomeado eProj, estimativa de Projetos de software. Na tela principal o usuário verá cinco botões, quatro desses fornecem acesso às funcionalidades disponibilizadas, e um último com informações sobre a pesquisa e o aplicativo.

---

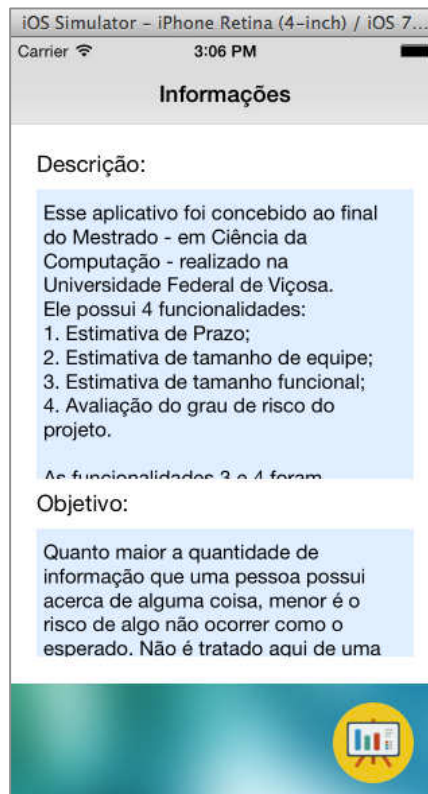
<sup>11</sup> O aplicativo será disponibilizado gratuitamente para download na loja oficial da Apple.



**Fonte:** Elaborado pelo autor.

**Figura 30.** Tela principal do aplicativo eProj.

Cada círculo na **Figura 30**, fornece acesso às telas do aplicativo. Todas serão detalhadas a seguir. Como início, será apresentada a tela de Informações.



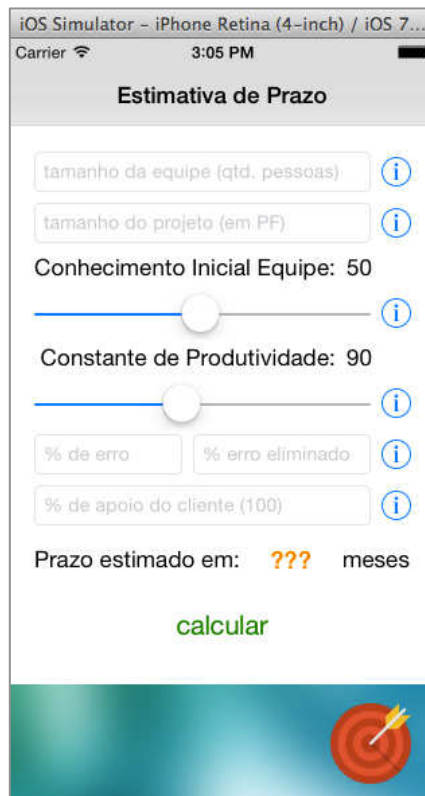
**Fonte:** Elaborado pelo autor.

**Figura 31. Tela de informações.**

A **Figura 31** mostra a tela de informações, nela foram colocados a descrição e objetivo do aplicativo, o conteúdo foi baseado na dissertação, ela é citada e existirá um link para acessá-la. Quanto às funcionalidades, elas são:

- Estimativa de Prazo;
- Estimativa de tamanho de equipe;
- Estimativa de tamanho funcional;
- Avaliação do grau de risco do projeto.

A funcionalidade de estimativa de prazo foi desenvolvida levando em consideração esta pesquisa. A utilização de equações em todo o modelo de dinâmica de sistemas, facilitou a construção do software.

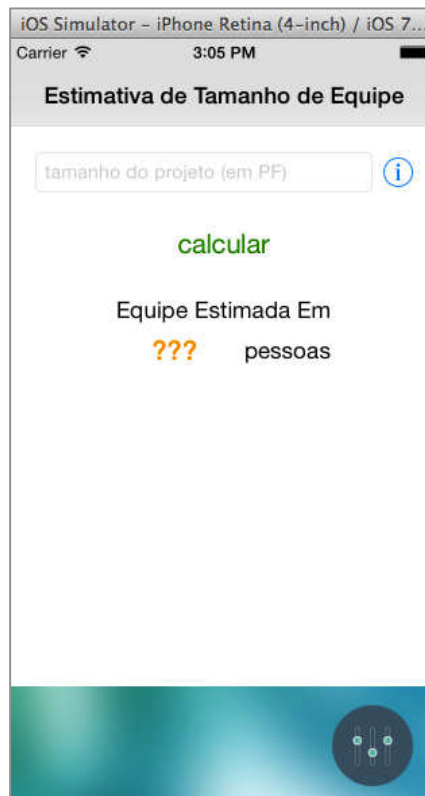


**Fonte:** Elaborado pelo autor.

**Figura 32. Tela de estimativa de prazo.**

Nessa primeira versão do aplicativo, não foram utilizados todos os fatores de risco colocados no modelo. A retirada dos fatores, turnover (saída de profissionais) e contratação, foi baseada na necessidade desses para o funcionamento das equações principais, esses fatores não são menos importantes, apenas possuem equações individuais que são calculadas separadamente e depois relacionadas com o modelo.

Assim como o cálculo do prazo, a estimativa de tamanho de equipe, foi retirada dos resultados alcançados com este trabalho. Durante as pesquisas foi observado que uma outra necessidade comum aos gerentes, é saber o tamanho de equipe necessária para realizar um projeto. Esta estimativa foi apresentada no estudo, e, a partir disso, foi inserida como funcionalidade no aplicativo.



**Fonte:** Elaborado pelo autor.

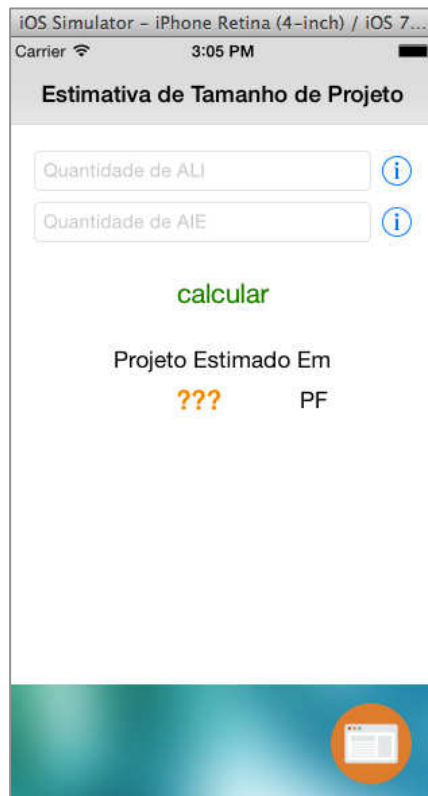
**Figura 33. Tela de estimativa de tamanho de equipe.**

O usuário pode estimar o tamanho de equipe necessária, inserindo o tamanho do projeto em pontos função; o resultado pode ser usado isoladamente, ou, para estimar o prazo do projeto. Outra estimativa importante é o tamanho do projeto, que é utilizado tanto para estimar a equipe, quanto o prazo de desenvolvimento.

A contagem Indicativa fornece um valor indicativo do número de pontos de função do sistema sem necessitar saber mais detalhes. Para realizar a contagem indicativa da NESMA<sup>12</sup>, é preciso somente realizar a identificação dos ALI (Arquivos Lógicos Internos) e AIE (Arquivos de Interface Externa), considerando-se 35 PF para cada ALI e 15 PF para cada AIE identificado (VAZQUEZ, 2009).

---

<sup>12</sup> NESMA (Netherlands Software Metrics Association) é a segunda maior organização de mensuração de tamanho funcional de software do mundo.



**Fonte:** Elaborado pelo autor.

**Figura 34. Tela de estimativa de tamanho de projeto.**

A **Figura 34** apresenta a funcionalidade de estimativa de tamanho de projeto, ela fornece suporte tanto para encontrar o tamanho de equipe, quanto para estimar o prazo do projeto. Para finalizar, foi utilizado o trabalho de McFarlan (1981), para desenvolver o cálculo do grau de risco de um projeto.

Para a avaliação de risco, foi utilizada a metodologia de portfólio de aplicações. Segundo Fernandes (1990), essa metodologia avalia o risco de um determinado projeto, e, apresenta sugestões de soluções gerenciais. Ela leva em consideração os seguintes riscos inerentes de um projeto:

- Falha na obtenção de todos ou alguns dos benefícios esperados;
- Custos de implementação que excedem os níveis planejados;
- Tempo de implementação maior que o planejado;
- Desempenho técnico do sistema final inferior ao estimado inicialmente;
- Incompatibilidade do sistema com o hardware e o software selecionados.

Fernandes (1990) discursa que a determinação do risco de um projeto compreende a combinação de três fatores.

- **Tamanho do projeto:** representado pelo prazo e número de departamentos envolvidos no desenvolvimento;
- **Experiência com a tecnologia:** referencia o grau de conhecimento da equipe e da organização acerca das tecnologias;
- **Estrutura do projeto:** representa o nível de estrutura com que o projeto será implantado; alta estrutura significa que o projeto foi formalizado e apoiado pelos usuários, o oposto é definido por uma baixa estrutura.

A **Figura 35** apresenta a combinação desses três fatores na definição do grau de risco.

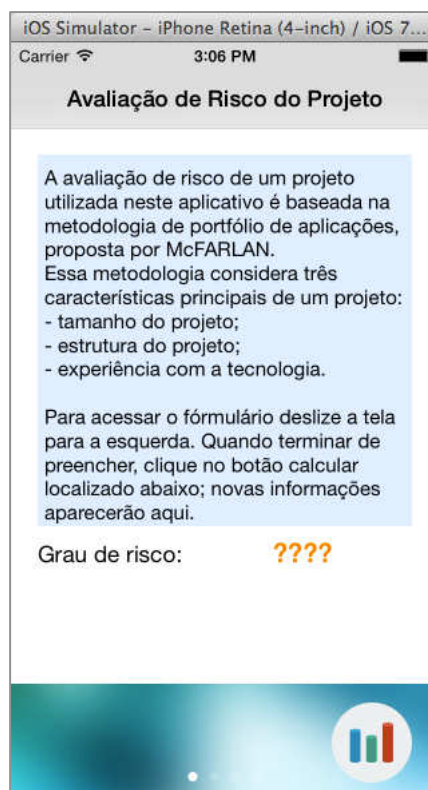
		GRAU DE ESTRUTURA DO PROJETO	
		ALTA	BAIXA
NOVIDADE TECNOLÓGICA	BAIXA	Projeto Grante Risco Baixo	Projeto Grante Risco Baixo
		Projeto Pequeno Risco Muito Baixo	Projeto Pequeno Risco Muito Baixo
		Projeto Grante Risco Médio	Projeto Grante Risco Muito Alto
	ALTA	Projeto Pequeno Risco Médio/Baixo	Projeto Pequeno Risco Alto

**Fonte:** Elaborado pelo autor e adaptado de (FERNANDES, 1990).

**Figura 35. Matriz de risco.**

Para estimar o grau de risco, foi utilizado o “Questionário de Avaliação de Risco” proposto por McFarlan (1981). Toda pergunta do questionário possui um peso e toda resposta uma pontuação, cada segmento do questionário representa um dos fatores de definição do risco do projeto, sendo avaliação do risco em: tamanho; estrutura; e tecnologia. A funcionalidade no aplicativo é representada por quatro

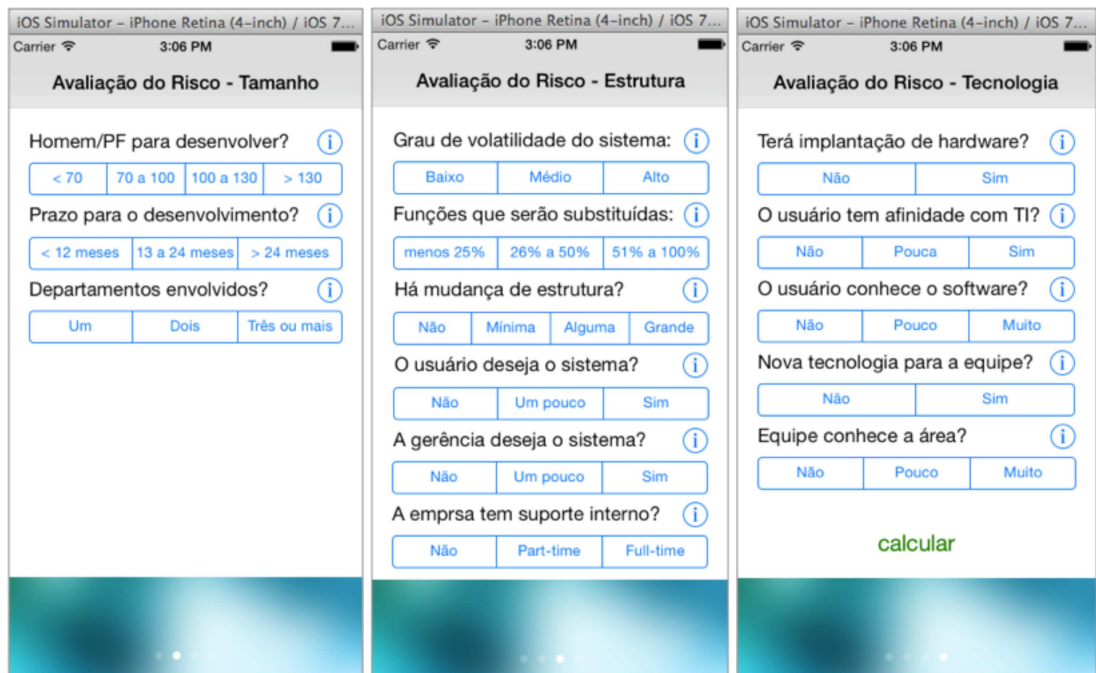
telas, sendo a primeira, uma tela de introdução da metodologia e de resultados, as demais são os questionários.



**Fonte:** Elaborado pelo autor.

**Figura 36. Tela de introdução da Avaliação do Grau de Risco.**

A tela da **Figura 36** representa a entrada da funcionalidade de estimativa de grau de risco. Para acessar os formulários o usuário deve deslizar a tela para a esquerda, assim, visualizando cada questionário. A seguir as definições de cada item dos formulários.



Fonte: Elaborado pelo autor.

**Figura 37. Tela de introdução da Avaliação do Grau de Risco.**

A **Figura 37** representa as três telas do questionário proposto por McFarlan (1981). Em cada tela foi feita a média ponderada e estabelecido o ponto médio; o usuário responde os questionários, e em cada um, o sistema define o grau de cada fator, se o valor é menor que o ponto médio, baixo, se o valor é maior, alto. Utilizando a **Figura 35** e a avaliação de cada fator, calcula-se o grau de risco do projeto.

McFarlan (1981) apresenta na metodologia, juntamente com o “Questionário de Avaliação de Risco”, quais abordagens gerenciais possuem maior contribuição de acordo com o grau de risco. Essas abordagens gerenciais fundamentam-se em (FERNANDES, 1990):

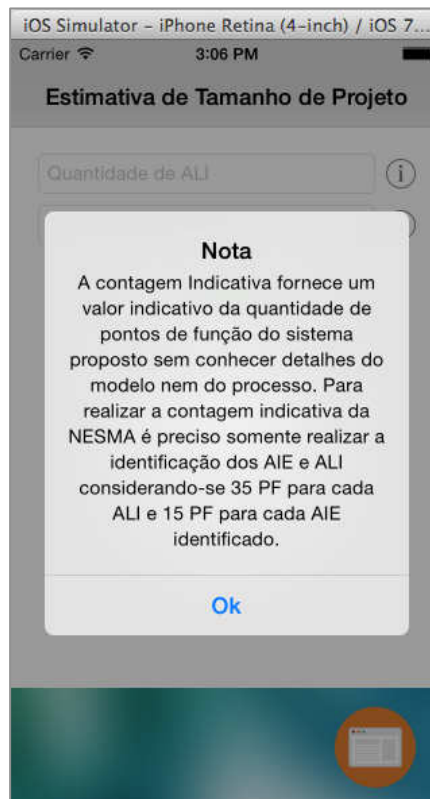
- Ferramentas de Integração Externa;
- Ferramentas de Integração Interna;
- Ferramentas de Planejamento;
- Ferramentas de Controle.



**Fonte:** Elaborado pelo autor.

**Figura 38.** Tela de resultado da avaliação do grau de risco.

Como pode ser visto na **Figura 38**, o grau de risco é apresentado e logo acima no campo de texto, são definidas as ferramentas gerenciais de maior contribuição. Se o leitor ateu-se aos detalhes, terá observado um botão ao lado de todos os campos de texto do aplicativo.



**Fonte:** Elaborado pelo autor.

**Figura 39.** Tela de exibição do botão de ajuda.

A **Figura 39** anterior apresenta o resultado de um desses botões. Eles foram colocados para auxiliar os usuários no preenchimento de todos os campos e entendimento do que está sendo feito.

Foram apresentadas todas as funcionalidades do aplicativo. A proposta é que essa seja a primeira versão e que ele sofra melhorias com sugestões dos usuários. Como dito na introdução deste tópico, o objetivo é atingir tanto os gerentes de projeto quanto, estudantes e professores. A motivação do autor em construir este resultado extra foi extrapolar o papel, mesmo o modelo sendo disponibilizado para acesso, ele é de maior utilidade para pesquisas posteriores e para a melhoria desta.

#### **5.4 Decisões em projeto utilizando o aplicativo de estimativa**

Neste ponto já foi apresentado o aplicativo desenvolvido, o qual utilizou esta pesquisa e outros estudos relacionados à estimativas. Agora será apresentado um cenário de utilização das funcionalidades fornecidas com a ferramenta. Lembrando que essas são:

- Estimativa de Prazo;
- Estimativa de tamanho de equipe;
- Estimativa de tamanho funcional;
- Avaliação do grau de risco do projeto.

Será apresentada uma sequência de utilização das estimativas, mas essa não restringe o uso individual de cada funcionalidade ou optar por seguir outra sequência. O projeto que será estimado é um sistema mobile que gerencia toda a estadia de um cliente em uma hospedagem, e.g., hotel, albergue, camping, cruzeiro etc.

Considere que o local de hospedagem possua diversas atividades, locais para compras (sejam comidas, equipamentos, roupas etc), diversos tipos de quartos, agendas com atividades, e aceitará “check in”, “check out” e pagamento via celular. Algumas das informação devem estar presentes tanto no celular do cliente quanto no servidor da empresa que controla o sistema, dessa maneira, alguns Arquivos Lógicos Interno – ALI (arquivos que ficam no celular do cliente), também são Arquivos de Interface Externa – AIE (arquivos que o celular acessa no servidor).

O cenário levará em consideração apenas o aplicativo, o software que gerencia o servidor e que os funcionários do local utilizam não será tratado nas estimativas. Na **Tabela 13** são apresentadas as definições do projeto que será estimado.

**Tabela 13. Definições do projeto.**

<b>DESCRIÇÃO</b>	<b>TIPO</b>
Hóspedes	ALI
Quartos	ALI
Gastos	ALI
Atividades	ALI
Registros	ALI
Agenda	ALI
Serviços	AIE
Quartos (servidor)	AIE
Gastos (servidor)	AIE
Atividades e Agenda (servidor)	AIE
Pagamento	AIE

**Fonte:** Elaborado pelo autor

Como apresentado anteriormente, para realizar a contagem indicativa usando o padrão NESMA, não são necessários mais detalhes do projeto. Somente realizar a identificação dos ALI e AIE (VAZQUEZ, 2009). Primeiramente foi estimado o tamanho funcional do projeto.

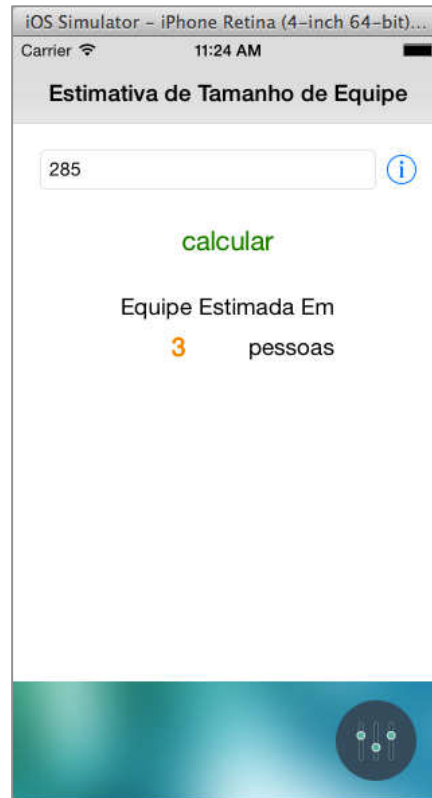


**Fonte:** Elaborado pelo autor.

**Figura 40. Estimando o tamanho do projeto.**

Os campos preenchidos na funcionalidade são nessa ordem: quantidade de ALI; quantidade de AIE. A partir deste ponto, o usuário poderia optar por estimar o tamanho da equipe ótima para realizar este projeto, ou optar por avaliar o prazo de desenvolvimento. No cenário em questão, o gerente possui apenas dois desenvolvedores, à primeira vista não seria necessário estimar o tamanho da equipe, uma vez que o tamanho da equipe já está definido por questões de recursos da empresa desenvolvedora. Embora os recursos já estejam definidos, nesta situação estimar o tamanho da equipe serve como parâmetro para avaliar se o recurso é abaixo, igual ou acima do sugerido, fornecendo assim não uma informação de

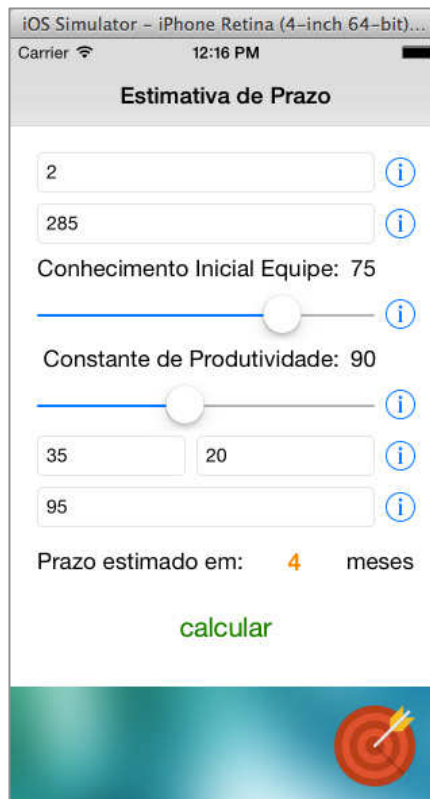
contingente necessário, mas se há um maior risco de atraso por utilizar menos desenvolvedores, e com isso, informar ao gerente que precisa se preocupar mais com a competência da equipe.



**Fonte:** Elaborado pelo autor.

**Figura 41. Estimando o tamanho da equipe.**

O campo preenchido na funcionalidade foi: tamanho do projeto em pontos função. O tamanho da equipe sugerida é de três pessoas e o gerente possui apenas dois desenvolvedores. Mas para este cenário, foi considerado que eles possuem muito conhecimento na tecnologia e área de desenvolvimento. Com as informações obtidas até aqui, pode ser estimado o prazo.



**Fonte:** Elaborado pelo autor.

**Figura 42. Estimando o prazo do projeto.**

Os campos preenchidos na funcionalidade são nessa ordem: tamanho da equipe; tamanho do projeto em pontos função; conhecimento inicial da equipe; constante de produtividade; porcentagem de erros mensais do projeto; porcentagem de correção dos erros; porcentagem de apoio do cliente. O primeiro campo era o recurso que o gerente detinha, o segundo campo foi estimado com a funcionalidade de estimativa de tamanho de projeto, e para o preenchimento dos demais campos, foi utilizado a **Tabela 10** com os valores base estimados para um projeto pequeno.

O prazo aferido foi de quatro meses, munido de todas as estimativas o gerente pode avaliar o risco do projeto. Para avaliar o grau de risco de um projeto, devem ser respondidos três questionários, sendo o primeiro a avaliação referente ao tamanho do projeto.

iOS Simulator - iPhone Retina (4-inch 64-bit)...

Carrier 3:14 PM

### Avaliação do Risco - Tamanho

Homem/PF para desenvolver? ⓘ

< 70 70 a 100 100 a 130 > 130

Prazo para o desenvolvimento? ⓘ

< 12 meses 13 a 24 meses > 24 meses

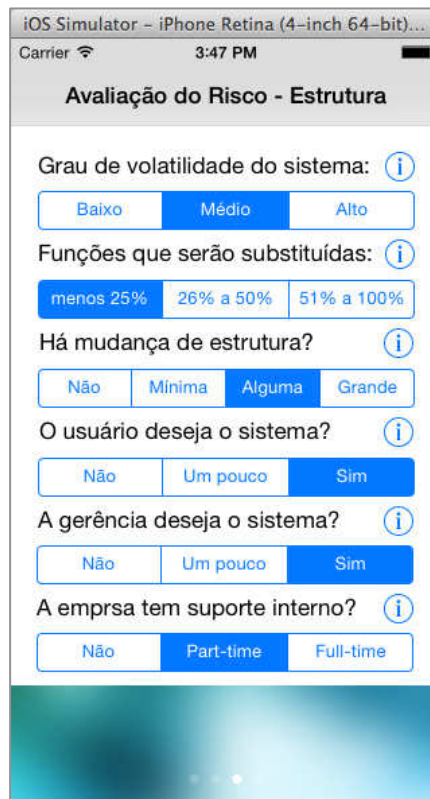
Departamentos envolvidos? ⓘ

Um Dois Três ou mais

**Fonte:** Elaborado pelo autor.

**Figura 43. Questionário avaliação de risco em tamanho.**

A primeira pergunta é referente à quantidade de pontos função que serão destinados a cada desenvolvedor. Como o projeto é de 285 PF, ele possui mais do que 130 PF para cada um. O prazo de desenvolvimento é menor do que doze meses e o número de departamentos envolvidos é três ou mais, i.e., departamento de projetos, comercial e financeiro. O próximo questionário é referente à estrutura do projeto.



**Fonte:** Elaborado pelo autor.

**Figura 44. Questionário avaliação de risco em estrutura.**

O grau de volatilidade do sistema é médio, ele possui os requisitos bem definidos, mas alguns devem sofrer mudanças com o avanço do ciclo de vida do projeto. O aplicativo irá substituir algumas funções, mas menos do que 25% das tarefas dos funcionários das empresas de hospedagens. Mesmo assim, irá levar a algumas mudanças de estrutura na empresa que o implantar.

Foi definido que o usuário deseja o sistema por facilitar o acesso à serviços dentro das hospedagens, e, uma vez que favorece os clientes, implica em interesse por parte da gerência por ser um atrativo a mais para a empresa. Por fim, com relação ao suporte interno, o mais comum é a empresa treinar os funcionários - que possuem diversas outras tarefas - para lidar com imprevistos acerca do aplicativo, e em casos extremos, chamar um suporte mais qualificado. Sendo assim, foi marcada a opção de suporte em parte do tempo.

Para finalizar, o último questionário avalia o risco com relação à tecnologia.

iOS Simulator - iPhone Retina (4-inch 64-bit)...

Carrier 4:15 PM

**Avaliação do Risco - Tecnologia**

Terá implantação de hardware? ⓘ

Não  Sim

O usuário tem afinidade com TI? ⓘ

Não  Pouca  Sim

O usuário conhece o software? ⓘ

Não  Pouco  Muito

Nova tecnologia para a equipe? ⓘ

Não  Sim

Equipe conhece a área? ⓘ

Não  Pouco  Muito

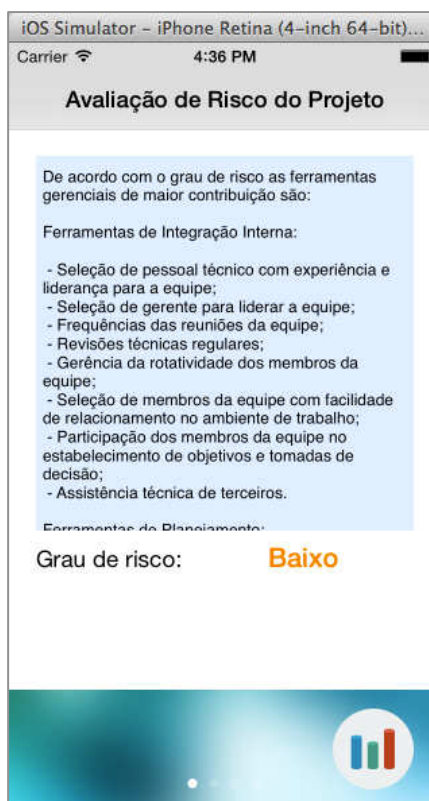
calcular

**Fonte:** Elaborado pelo autor.

**Figura 45. Questionário avaliação de risco em tecnologia.**

A seguir a análise para o último grupo de perguntas e respostas. Como dito no início deste item, este cenário apresentado avalia apenas o projeto do aplicativo, não levando em consideração o servidor e o sistema que o controla, desse modo, o leitor deve considerar que toda a implantação de hardware já está pronta para o funcionamento do aplicativo, justificando assim, a opção marcada. Nas duas próximas perguntas, o usuário deve ser entendido como os clientes da hospedagem e os funcionários do local, tanto para a questão de afinidade com TI e conhecimento do software. Foi considerado que os usuários conhecem um pouco das duas questões, por levar em consideração que atualmente o uso de aplicativos em celulares tem crescido muito, mesmo sendo aplicativos diferentes, a lógica de funcionamento normalmente é muito semelhante. Neste cenário como já apresentado, optou-se por uma equipe competente, a qual tem domínio da tecnologia de desenvolvimento e conhece um pouco da área, não são especialistas em hotelaria, mas suas experiências em hospedagens e as explicações dos clientes, são suficientes para justificar a opção marcada.

Após preencher completamente todos os formulários, pode ser calculado a avaliação do grau de risco do projeto.



**Fonte:** Elaborado pelo autor.

**Figura 46. Resultado da avaliação do grau de risco do projeto.**

O projeto possui baixo grau de risco, lembrando que a variação é de muito baixo a muito alto. Este resultado é coerente com o projeto em questão, mesmo o projeto possuindo algumas incertezas com relação ao tamanho da equipe, o tipo de projeto é comum e seu tamanho funcional pequeno. Por mais que o aplicativo possa parecer interessante do ponto de vista de mercado ou para o usuário, ele não possui grandes desafios de desenvolvimento e possui semelhanças com diversos outros, como sistemas de gerenciamento de: clientes de restaurantes, equipes de projeto, congressistas de evento etc. Além do grau de risco, como resultado também são apresentadas as melhores ferramentas gerenciais para o grau do projeto:

- **Ferramentas de Integração Interna:**
  - Seleção de pessoal técnico com experiência e liderança para a equipe;

- Seleção de gerente para liderar a equipe;
- Frequências das reuniões da equipe;
- Revisões técnicas regulares;
- Gerência da rotatividade dos membros da equipe;
- Seleção de membros da equipe com facilidade de relacionamento no ambiente de trabalho;
- Participação dos membros da equipe no estabelecimento de objetivos e tomadas de decisão;
- Assistência técnica de terceiros.
- **Ferramentas de Planejamento:**
  - PERT, CPM, Redes de Precedência;
  - Seleção dos pontos de controle ou milestones;
  - Padrões de especificações de sistemas;
  - Especificação de estudos de viabilidade;
  - Processos de aprovação de fases do projeto;
  - Procedimento de revisão de sistemas.
- **Ferramentas de Controle:**
  - Relatórios formais de progresso;
  - Controle de mudanças;
  - Apresentação de produtos nos pontos de controle;
  - Relatórios de progresso Real x Previsto.

O gerente pode avaliar quais dessas ferramentas indicadas fazem mais sentido à sua realidade, e aplicá-las no dia a dia do gerenciamento de projeto. Com esta última análise são finalizadas as estimativas.

Os cenários apresentados neste tópico, tinha como objetivo descrever o uso do aplicativo, tanto para o gerenciamento de projetos quanto para o estudo sobre essa área. Outro objetivo era a apresentação da abrangência da ferramenta, e a importância desta pesquisa; a qual, forneceu as mais diversas equações para a criação do aplicativo. Por fim, deseja-se que o cenário apresentado motive o uso da ferramenta, e que esta possa exercer a função para a qual foi criada.

## APÊNDICE C

**Tabela 14. Variáveis de riscos propostas por Schmidt et al. (2001)**

<b>Fatores de Risco</b>
1. Mudança de Escopo/objetivos
2. Falta de envolvimento adequado dos usuários
3. Requisitos mal entendidos e/ou mal definidos
4. Escopo/objetivos pouco claros ou equivocados
5. Prazos e tempo para tarefas mal estimados
6. Gerenciamento impróprio de mudanças
7. Volatilidade nos requisitos (falta de requisitos estáticos)
8. Custos mal estimados
9. Conflito entre departamentos de usuário
10. Falha em gerenciar as expectativas finais dos usuários
11. Planejamento inexistente ou inadequado
12. Pessoal envolvido insuficiente/inapropriado
13. Falta de conhecimento/competência dos envolvidos no projeto
14. Falta de Cooperação dos usuários
15. Falta de metodologia efetiva em gerenciamento de projetos
16. Controle pobre ou inexistente
17. Adoção de novo método/tecnologia
18. Falha em obter comprometimento do cliente
19. Definição imprópria de papéis e responsabilidades
20. Falta de comprometimento da alta gerência
21. Falta de habilidade para o gerenciamento de projetos
22. Assunto novo ou não familiar
23. Mudança no proprietário do projeto ou na alta gerência
24. Rotatividade da equipe
25. Projeto com múltiplos fornecedores
26. Usar nova metodologia de desenvolvimento em projetos importantes

**Fonte:** Elaborado pelo autor.

**Tabela 15. Variáveis de riscos propostas por Leopoldino (2004)**

---

<b>Fatores de Risco</b>
1. Mudança de Escopo/objetivos
2. Falta de envolvimento adequado dos usuários
3. Requisitos mal entendidos e/ou mal definidos
4. Escopo/objetivos pouco claros ou equivocados
5. Prazos e tempo para tarefas mal estimados
6. Gerenciamento impróprio de mudanças
7. Volatilidade nos requisitos (falta de requisitos estáticos)
8. Custos mal estimados
9. Falta de poderes para o gerenciamento de projetos
10. Conflito entre departamentos de usuário
11. Falha em gerenciar as expectativas finais dos usuários
12. Planejamento inexistente ou inadequado
13. Pessoal envolvido insuficiente/inapropriado
14. Falta de conhecimento/competência dos envolvidos no projeto
15. Falta de Cooperação dos usuários
16. Falta de metodologia efetiva em gerenciamento de projetos
17. Controle pobre ou inexistente
18. Adoção de novo método/tecnologia
19. Falha em obter comprometimento do cliente
20. Definição imprópria de papéis e responsabilidades
21. Falta de comprometimento da alta gerência
22. Falta de motivação da equipe
23. Falta de habilidade para o gerenciamento de projetos
24. Assunto novo ou não familiar

---

**Fonte:** Elaborado pelo autor.

**Tabela 16. Variáveis de riscos propostas por Barki et al. (1993)**

---

<b>Fatores de Risco</b>
1. Mudança de Escopo/objetivos
2. Falta de envolvimento adequado dos usuários
3. Pessoal envolvido insuficiente/inapropriado
4. Falta de conhecimento/competência dos envolvidos no projeto
5. Falta de Cooperação dos usuários
6. Adoção de novo método/tecnologia
7. Falta de comprometimento da alta gerência
8. Assunto novo ou não familiar
9. O sistema possui integração e interface com outros sistemas
10. Sistema complexo
11. Tarefas complexas
12. Falta de tecnologias maduras/existentes

---

**Fonte:** Elaborado pelo autor.

**Tabela 17. Variáveis de riscos propostas por Boehm (1991)**

---

<b>Fatores de Risco</b>
1. Mudança de Escopo/objetivos
2. Requisitos mal entendidos e/ou mal definidos
3. Escopo/objetivos pouco claros ou equivocados
4. Prazos e tempo para tarefas mal estimados
5. Gerenciamento impróprio de mudanças
6. Volatilidade nos requisitos (falta de requisitos estáticos)
7. Custos mal estimados
8. Pessoal envolvido insuficiente/inapropriado
9. Falta de conhecimento/competência dos envolvidos no projeto
10. O sistema possui integração e interface com outros sistemas
11. Tarefas complexas
12. Deficiência de execução em tempo real

---

**Fonte:** Elaborado pelo autor.

## REFERÊNCIAS BIBLIOGRÁFICAS

AMBRÓSIO, B. G., BRAGA, J. L., RESENDE FILHO, M. A.: Modeling and scenario simulation for decision support in management of requirements activities in software projects. *Journal of Software Maintenance and Evolution (Print)*, v.23, p.35-50, 2011. <http://dx.doi.org/10.1002/smr.469> (2011)

BARKI, H., RIVARD, S., TALBOT, J.: An Integrative Contingency Model of Software Project Risk Management. *Journal of Management Information Systems*, V.17, N 4, p. 37-69, Spring (2001)

BARKI, H., RIVARD, S., TALBOT, J.: Toward an assessment of software development risk. *Journal of Management Information Systems*, 10. 2 (Fall 1993), 203-225. (1993)

BARROS, M. de O., WERNER, C. M. L., TRAVASSOS, G. H.: Supporting risks in software project management. *The Journal of Systems and Software*, 23 Nov. (2002)

BOEHM, B. W.: Software Engineering Economics, *IEEE Transactions on Software Engineering*, Vol. SE-10, No. 1, January, pp. 4-21 (1984)

BOEHM, B. W.: Software Risk Management: principles and practices, *IEEE Software*, Vol. 8, No. 1, pp. 32-40 (1991)

CHARETTE, R.N.: *Software Engineering Risk Analysis and Management*. McGraw-Hill, New York (1989)

COELHO, J., J., de S.: Modelo de dinâmica de sistemas para apoio a decisões no processo de inspeção de Software. Dissertação de Mestrado – CCE/DPI, Universidade Federal de Viçosa, Viçosa (2013)

COSTA, S. D.: Apoio à decisão na gestão de pessoas em projetos de software: uma

abordagem utilizando simulação com dinâmica de sistemas. Dissertação de Mestrado – CCE/DPI, Universidade Federal de Viçosa, Viçosa (2012)

CMMI / SEI – Software Engineering Institute.: Capability Maturity Model Integration – CMMI Overview. Carnegie Mellon University, Version 1.3 (2011)

DAIBERT, M. S. Monitoramento De Riscos Em Projetos De Software: Uma Abordagem Baseada Em Dinâmica De Sistemas E Técnicas De Inteligência Computacional. (2010)

FERNANDES, A. A.; KUGLER, J. L. C.: Gerências de projetos de sistemas – uma abordagem prática. Livros Técnicos e Científicos Editora, 2ed., Rio de Janeiro (1990)

FORRESTER J.W. Industrial Dynamics. MIT Press: Cambridge, MA, 1961.

FORRESTER J.W. Principles of Systems. MIT Press: Cambridge, MA, 1968.

FORRESTER, J. W.; SENGE, P. Tests for building confidence in system dynamics models. A. Legasto *et al.* (eds.), TIMS Studies in the Management Sciences (System Dynamics), North-Holland, The Netherlands, p. 209-228 (1980)

GIBBS, W.W.: Software's chronic crisis. Scientific American. Vol. 271, No. 3, September, pp. 86-95 (1994)

GLASS R., L.: Facts and Fallacies of Software Engineering. Addison Wesley, October 25, ISBN: 0-321-11742-5 (2002)

HERMSDORF, V. O. *et al.* Modelagem da atividade de elicitação de requisitos utilizando a técnica de entrevista: uma abordagem utilizando dinâmica de sistemas. XIV WER - Workshop em Engenharia de Requisitos Proceedings XIV CibSE. Rio de Janeiro, RJ, v. 14. p. 309-320 (2011)

HUSTON, D. X., MACKULAK, G. T., COLLOFELLO J. S.: Stochastic Simulation of Risk Factor Potential Effects for Software Development Risk Management, The Journal of Systems and Software 59, p. 247-257 (2001)

JOHNSON. J.: Chaos: the dollar drain of IT project failures. Application Development Trends, 2, 1 January. 41-7 (1995)

JONES, C.: Programming Languages Table. Software Productivity Research, Inc. Release 8.2, March (1996)

KELLNER, M. I., MADACHY, R. J., RAFFO, D.M.: Software process simulation modeling: Why? What? How? Journal of System and Software 46, pp. 91-105 (1999)

LEOPOLDINO, C. B. Avaliação de Risco em Desenvolvimento de Software. Dissertação de Mestrado – Escola de Administração do Rio Grande do Sul, Porto Alegre (2004)

LOPES, J. S., BRAGA, J. L.: Estimativas de tamanho em projetos de software utilizando pontos função. Engenharia de Software Magazine. Ano 4, Ed. 44 (2012)

MADACHY, R.: Software Process Dynamics, Wiley-IEEE Press, Washington D.C. (2007)

McFARLAN, F. W.: Information Technology Changes The Way You Compete. Harvard Business Review. May-June (1984)

McFARLAN, F. W.: Portfolio Approach to Information Systems. Harvard Business Review. September-October (1981)

PENDHARKAR, P. C., RODGER, J. A., SUBRAMANIAN, G. H.: An empirical study of the Cobb-Douglas production function properties of software development effort. Information & Software Technology. Vol. 50, pp. 1181-1188 (2008)

PETERSON, I.: Fatal defect: chasing killer computer bugs. New York, NY: Vintage Books, pp. 268 (1996)

PINDYCK, R. S., RUBINFELD, D. L.: Microeconomia. Ed. 7, Editora, Prentice Hall (2010)

PMBOK / PMI - Project Management Institute.: A Guide to the Project Management Body of Knowledge. Newtown Square, Pennsylvania – USA. 5 ed (2013)

PUTNAM, L. H.: A General Empirical Solution to the Macro Software Sizing and Estimating Problem, IEEE Transactions on Software Engineering,, vol. SE-4, pp. 345-361 (1978)

SCHMIDT, R., LYYTINEN, K., KEIL, M., CULE, P.: Identifying Software Project Risks: An International Delphi Study. Journal of Management Information System. Vol. 17, No. 4, pp. 5-36 (2001)

SHULL, W. ET AL. What We Have Learned about Fighting Defects. International Software Metrics Symposium, Ottawa, Canada (2002)

SENGE, P.M. A quinta disciplina: arte, teoria e prática da organização de aprendizagem. ed. 28, São Paulo, SP. Editora Best Seller (2012)

STERMAN, J. D. (2000) Business dynamics: systems thinking and modeling for a complex world. Boston, MA: Irwin McGraw-Hill, 982 p. (2000)

VAZQUEZ, C. E., SIMÕES, G. S., ALBERT, R. M.: Análise de ponto de função medição, estimativa e gerenciamento de projetos de software. São Paulo, Editora Érica, ed. 8 (2009)

VENSIM.: Vensim from Ventana Systems, Inc, 1996-2013. Disponível em: <http://www.vensim.com/freedownload.html>. (2013)

VILLELA, P. R. C.: Introdução à dinâmica de sistemas. Universidade Federal de Juiz de Fora, Juiz de Fora (2005)

WAN, J., WAN, D., ZHANG, H.: Case Study on Business Risk Management for Software Outsourcing Service Provider with ISM\*. Technology and Investment, 1, 257-266 (2010)