# UNIVERSIDADE DE SÃO PAULO FACULDADE DE ECONOMIA, ADMINISTRAÇÃO E CONTABILIDADE DEPARTAMENTO DE ADMINISTRAÇÃO PROGRAMA DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO

# UM NOVO ENFOQUE PARA O GERENCIAMENTO DE PROJETOS DE DESENVOLVIMENTO DE SOFTWARE

Marisa Villas Bôas Dias

Orientador: Prof. Dr. Antonio Cesar Amaru Maximiano

SÃO PAULO

2005

Prof. Dr. Adolpho José Melfi Reitor da Universidade de São Paulo

Profa. Dra. Maria Tereza Leme Fleury Diretora da Faculdade de Economia, Administração e Contabilidade

> Prof. Dr. Eduardo Pinheiro Gondim de Vasconcellos Chefe do Departamento de Administração

Prof. Dr. Isak Kruglianskas Coordenador do Programa de Pós-Graduação em Administração

#### MARISA VILLAS BÔAS DIAS

# UM NOVO ENFOQUE PARA O GERENCIAMENTO DE PROJETOS DE DESENVOLVIMENTO DE SOFTWARE

Dissertação apresentada ao Departamento de Administração da Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo, como requisito para a obtenção do título de Mestre em Administração.

Orientador: Prof. Dr. Antonio Cesar Amaru Maximiano

SÃO PAULO

2005

Dissertação defendida e aprovada no Departamento de Administração da Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo – Programa de Pós-Graduação em Administração, pela seguinte banca examinadora:

Prof. Dr. Antonio Cesar Amaru Maximiano

Prof. Dr. Hamilton Luiz Correa Prof. Dr. Roque Rabechini Junior

## FICHA CATALOGRÁFICA

Elaborada pela Seção de Processamento Técnico do SBD/FEA/USP

Dias, Marisa Villas Bôas

Um novo enfoque para o gerenciamento de projetos de desenvolvimento de software / Marisa Villas Bôas Dias. -- São Paulo, 2005.

202 p.

Dissertação (Mestrado) — Universidade de São Paulo, 2005 Bibliografia

1. Administração de projetos 2. Softwares 3. Métodos ágeis I. Universidade de São Paulo. Faculdade de Economia, Administração e Contabilidade. II. Título.

Aos meus dois grandes amores Pedro e Marcelo. À minha mãe Ilia. Agradeço à minha família, em especial, ao meu filho Pedro o amor incondicional e ao meu amigo e companheiro Marcelo os dois maiores presentes de minha vida e o apoio às minhas escolhas. Agradeço à minha mãe o carinho dedicado e o esforço em propiciar a minha sólida formação. Agradeço à Sra. Selma e ao Sr. Antonio o acolhimento e o incentivo ao meu desenvolvimento. Agradeço às minhas irmãs e ao meu irmão a sua amizade e, em especial à Mônica, o empenho na revisão deste trabalho. Agradeço ao meu orientador e amigo Prof. Dr. Antonio Cesar Amaru Maximiano a confiança, os sábios conselhos e as oportunidades oferecidas. Agradeço ao Prof. Dr. Alonso Mazini Soler, meu co-orientador e também amigo, a apresentação deste tema tão fascinante, o amplo apoio acadêmico e os constantes desafios propostos. Agradeço ao Prof. Dr. Roberto Sbragia e ao Prof. Dr. Isak Kruglianskas as sugestões muito úteis a esta dissertação. Agradeço ao Fernando Henrique Ferraz Pereira da Rosa e à Profa. Dra. Julia Maria Pavan Soler o inestimável auxílio no tratamento estatístico dos dados desta pesquisa e o apoio metodológico. Agradeço ao Luis Cesar Verdi o exemplo de profissionalismo e o auxílio prestado em minha transição de carreira. Agradeço à Vera Vasconcelos o ombro amigo e aos demais amigos da "HP Consulting", em especial, ao Farhad Abdollahyan e ao José Finocchio, com os quais muito aprendi sobre a "difícil arte" de gerenciar projetos. Agradeço aos colegas da FEA – USP, em especial à Cristina Fedato, a prazerosa convivência e a valiosa troca de experiências. Por fim, agradeço à Universidade de São Paulo, em especial à POLI - USP e à FEA - USP, todo o conhecimento transmitido desde a minha graduação.

"Não podemos resolver os nossos problemas com o mesmo pensamento que usamos para criá-los".

\*\*Albert Einstein\*\*

#### **RESUMO**

Esta dissertação tem por objetivo principal identificar o enfoque de gerenciamento de projetos - ágil ou clássico - mais apropriado para o desenvolvimento de software com o uso de Métodos Ágeis. De forma mais específica, a dissertação investiga se existe uma associação entre o desempenho dos projetos de desenvolvimento de software realizados com o uso de Métodos Ágeis e o enfoque de gerenciamento de projetos adotado. Este trabalho é decorrente de um estudo exploratório inicial, por meio do qual se buscou a ampliação do conhecimento sobre o tema, a elaboração de um estudo comparativo entre o Gerenciamento Ágil de Projetos e o Gerenciamento Clássico de Projetos, a investigação das principais características de um projeto de desenvolvimento de software realizado com o uso de Métodos Ágeis, a identificação de uma comunidade de pessoas que tivesse experiência em projetos desta natureza e, por fim, a estruturação de uma segunda etapa de pesquisa. Esta segunda etapa, de caráter quantitativo-descritivo, visou à descrição do conjunto de dados e a determinação das relações relevantes entre as variáveis de interesse. Para tanto, foram definidas as variáveis independentes, dependentes e intervenientes da pesquisa, como sendo, respectivamente, os enfoques de gerenciamento de projetos ágil e clássico, o desempenho dos projetos de desenvolvimento de software e os desenvolvimentos de software conduzidos com o uso dos Métodos Ágeis. Procedeu-se a uma amostragem intencional por julgamento, sendo selecionada uma amostra de pesquisa composta por pessoas com interesse e/ou experiência em projetos de desenvolvimento de software com o uso de Métodos Ágeis, associadas a grupos de internet especializados na discussão sobre o tema. Para a coleta de dados utilizou-se um questionário auto-administrado, enviado por meio de correio eletrônico à amostra selecionada. O tratamento dos dados foi feito com o uso de métodos estatísticos: análise descritiva, análise discriminante e regressão logística. A partir dos resultados da pesquisa pôde-se concluir que os Métodos Ágeis e o Gerenciamento Ágil de Projetos, apesar de recentes, já fazem parte da realidade brasileira. Quanto à resposta à pergunta problema, não houve evidência amostral para encontrar uma associação estatisticamente significativa entre o desempenho de um projeto de desenvolvimento de software e o enfoque de gerenciamento de projetos adotado, não sendo possível comprovar, de forma conclusiva, a existência de um enfoque de gerenciamento de projetos mais apropriado para o desenvolvimento de software com o uso de Métodos Ágeis. Porém, os resultados da análise descritiva sugeriram que a maioria dos respondentes indicou o Gerenciamento Ágil de Projetos como o enfoque mais apropriado para o desenvolvimento de software com o uso de Métodos Ágeis. Pôde-se constatar também a possibilidade de se adotar qualquer um dos enfoques de gerenciamento de projetos – ágil ou clássico – ou mesmo uma combinação deles, nos projetos desta natureza. Na pesquisa ainda foram identificados o critério primordial utilizado para mensuração do desempenho dos projetos de desenvolvimento de software realizados com o uso de Métodos Ágeis, as características principais destes projetos, seus fatores críticos de sucesso, além de se comprovar a importância do apoio da alta administração na adoção do Gerenciamento Ágil de Projetos. Cabe ressaltar que todas estas conclusões devem ficar restritas ao âmbito desta dissertação. Para pesquisas futuras recomenda-se a adequação do instrumento de pesquisa e o cuidado especial na seleção da amostra. Sugere-se a exploração de assuntos correlatos ao tema, como o estudo da prontidão das organizações para a adoção do Gerenciamento Ágil de Projetos, ou mesmo, a avaliação dos resultados obtidos com sua aplicação. Por fim, o cenário brasileiro atual favorece a realização de estudos nesta temática, atendendo às necessidades de pesquisadores que se interessam pelo assunto.

#### **ABSTRACT**

The main purpose of this here research is to identify the most appropriated project management approach for software development using the Agile Methods. Essentially, this study aims to determine if there is a statistical relationship between the software development project performance and its project management approach. This research is based on a preliminary exploratory study that ensured the content and context understanding, the comparison between the Agile Project Management and the Classical Project Management, the identification of the main characteristics of an agile software development project, the identification of the people involved in this kind of software development initiative and also provided the basis to structure a second phase of this research. This second quantitativedescriptive phase intends to describe the research data and to find the relevant relationships between the variables of interest. The independent variables of this study were defined as being the agile and classical project management approaches; the dependent variable was defined as being the project performance; and, the intervenient variables were defined as being the agile software development projects. A non-probabilistic intentional sampling was done. The sampling framework was drawn from people who have experience managing or participating in software projects conducted using one of the Agile Methods, and were associated to internet discussion groups on this theme. For data gathering it was used a selfapplied survey, sent by e-mail to the selected groups. The respondents' data were primarily analyzed using descriptive analysis and, after that, using discriminant analysis and logistics regression. It was possible to conclude that, although the Agile Methods and the Agile Project Management are quite recent, they are being used here in Brazil. Considering the main research question, there was not a statistical evidence to prove the relationship between the software development project performance and its project management approach. Thus, it was not possible to establish a final conclusion about the most appropriated project management approach for software development using the Agile Methods. Nevertheless, the results of the descriptive analysis indicated that most of the respondents tended to choose the Agile Project Management as the approach that better fits the agile software development initiatives. It was also possible to conclude that the referred projects may be managed using either the agile or the classical project management approach, or even using a combination of these two approaches. The main agile software development project success criterias were identified, as well as the project characteristics and the main critical success factors. The important role of the upper management in supporting the Agile Project Management adoption was also discussed. The research conclusions should be restricted to this here context. For future studies, special attention should be paid to the research instrument improvement and to the sampling process. The study of the organization readiness for the Agile Project Management adoption or the analysis of this project management approach implementation results could be the aims of new investigations. Finally, it is important to notice that the current Brazilian scenario stimulates and favors the development of future studies, meeting the expectations of the researchers that are interested in this subject.

# **SUMÁRIO**

LISTA DE TABELAS	3
LISTA DE GRÁFICOS	6
LISTA DE ILUSTRAÇÕES	7
1 INTRODUÇÃO	
1.1 Contexto e Justificativa	8
1.2 Pergunta de Pesquisa	17
1.3 Objetivos e Contribuições	
1.4 Delimitação do Estudo	18
1.5 Modelo Conceitual	19
1.6 Estrutura do Trabalho	20
2 REVISÃO BIBLIOGRÁFICA	21
2.1 Desenvolvimento de Software	21
2.1.1 Processo de Desenvolvimento de um Novo Produto ou Software	21
2.1.2 Desenvolvimento de Software no Brasil	25
2.2 Gerenciamento Clássico de Projetos	29
2.2.1 Projetos e o Gerenciamento Clássico de Projetos	29
2.2.2 Processos do Gerenciamento Clássico de Projetos	32
2.2.3 Projetos de Desenvolvimento de Software	
2.3 Sucesso e Fracasso no Gerenciamento de Projetos	44
2.3.1 Visão Tradicional do Sucesso de um Projeto	45
2.3.2 Desempenho dos Projetos de Desenvolvimento de Software	49
2.3.3 Visão Moderna de Sucesso de um Projeto	53
2.4 Métodos Ágeis de Desenvolvimento de Software	
2.4.1 Definição e Origem dos Métodos Ágeis de Desenvolvimento de Software	58
2.4.2 Manifesto para o Desenvolvimento Ágil de Software	62
2.4.3 Principais Métodos Ágeis de Desenvolvimento de Software	
2.4.3.1 Extreme Programming (XP)	
2.4.3.2 Scrum	
2.4.3.3 Crystal Methods	
2.4.3.4 Dynamic Systems Development Method (DSDM)	
2.4.3.5 Feature-Driven Development (FDD)	
2.4.3.6 Lean Development (LD)	
2.4.3.7 Adaptative Software Development (ASD)	
2.4.3.8 Resumo das Características Principais dos Métodos Ágeis	
2.4.4 Aplicação dos Métodos Ágeis nas Organizações	
2.4.5 Resultados da Aplicação dos Métodos Ágeis	
2.4.6 Limitações dos Métodos Ágeis	86
2.4.7 Fatores Críticos de Sucesso de Projetos de Desenvolvimento de Software	
Realizados com o Uso dos Métodos Ágeis	
2.5 Gerenciamento Ágil de Projetos	
2.5.1 Evolução do Conceito dos Métodos Ágeis para o Gerenciamento de Projetos	
2.5.2 Definição, Valores e Princípios do Gerenciamento Ágil de Projetos	
2.5.3 Ciclo de Vida e Fases do Gerenciamento Ágil de Projetos	
2.5.4 Práticas do Gerenciamento Ágil de Projetos	
2.5.4.1 Práticas da Fase Visão	
2.5.4.2 Práticas da Fase Especulação	
2.5.4.3 Práticas da Fase Exploração	104

2.5.4.4 Prática da Fase Adaptação	104
2.5.4.5 Prática da Fase Encerramento	106
2.5.5 Aplicações, Resultados e Limitações do Gerenciamento Ágil de Projetos	106
2.5.6 Comparação: Gerenciamento Clássico e Gerenciamento Ágil de Projetos	
2.6 Considerações Finais	110
3 METODOLOGIA DE PESQUISA	114
3.1 Tipologia da Pesquisa	114
3.2 Variáveis da Pesquisa	116
3.3 Amostra da Pesquisa	118
3.4 Coleta de Dados da Pesquisa	119
3.5 Tratamento de Dados da Pesquisa	122
3.5.1 Análise Descritiva	122
3.5.2 Análise Discriminante	123
3.5.3 Regressão Logística	125
3.6 Limitações do Método	125
3.7 Considerações Finais	
4 RESULTADOS E ANÁLISE	128
4.1 Análise Descritiva dos Dados da Pesquisa	128
4.1.1 Análise Descritiva dos Respondentes e Informações Complementares	128
4.1.2 Análise Descritiva dos Projetos de Desenvolvimento de Software	130
4.1.3 Análise Descritiva das Práticas Relativas aos Métodos Ágeis	133
4.1.4 Análise de Associação por Tabelas de Contingência	136
4.2 Análise das Variáveis Relacionadas ao Desempenho dos Projetos de Desenvolvi	mento
de Software	147
4.2.1 Resultados da Análise Discriminante	147
4.2.2 Resultados da Regressão Logística	150
4.3 Análise das Variáveis Relacionadas ao Enfoque de Gerenciamento de Projetos	151
4.3.1 Resultados da Análise Discriminante	151
4.3.2 Resultados da Regressão Logística	153
4.4 Considerações Finais e Limitações Observadas	154
5 CONCLUSÕES	
REFERÊNCIAS	163
ANEXOS	174

# LISTA DE TABELAS

Tabela I - Mapeamento dos processos de gerenciamento de projeto	38
Tabela 2 - Visão geral do SW-CMM	
Tabela 3 - Fatores críticos de sucesso em ordem de importância	48
Tabela 4 - Fatores críticos de sucesso em projetos de desenvolvimento de software	51
Tabela 5 - Novo paradigma da gestão de projetos	56
Tabela 6 - Das velhas restrições aos novos processos	56
Tabela 7 - Princípios dos métodos ágeis de desenvolvimento de software	63
Tabela 8 - Características principais para utilização do XP	68
Tabela 9 - Características principais para utilização do Scrum	69
Tabela 10 - Características principais para utilização dos Crystal Methods	70
Tabela 11 - Características principais para utilização do FDD	72
Tabela 12 - Características principais dos métodos ágeis selecionados	74
Tabela 13 - Comparação entre os enfoques clássico e ágil de desenvolvimento de software	75
Tabela 14 - Características das empresas pesquisadas	82
Tabela 15 - Ganhos de produtividade com XP em projetos Web	84
Tabela 16 - Sumário dos pressupostos relativos aos princípios propostos pela Agile Allianc	e87
Tabela 17 - Princípios do gerenciamento ágil de projetos	99
Tabela 18 - Práticas da fase "Visão" do gerenciamento ágil de projetos	102
Tabela 19 - Práticas da fase "Especulação" do gerenciamento ágil de projetos	103
Tabela 20 - Práticas da fase "Exploração" do gerenciamento ágil de projetos	104
Tabela 21 - Práticas da fase "Adaptação" do gerenciamento ágil de projetos	105
Tabela 22 - Aplicabilidade do gerenciamento ágil de projetos	107
Tabela 23 - Alinhamento entre os enfoques ágil e clássico de gerenciamento de projetos	108
Tabela 24 - Comparação dos enfoques clássico e ágil de gerenciamento de projetos por a	área
de conhecimento	110
Tabela 25 - Resumo da metodologia de pesquisa empregada	127
Tabela 26 - Experiência em desenvolvimento de software com o uso de Métodos Ágeis	178
Tabela 27 - Qualificação do respondente quanto ao cargo (Q1)	178
Tabela 28 - Tempo de experiência com Métodos Ágeis (Q2)	178
Tabela 29 - Método ágil utilizado (Q3)	
Tabela 30 - Percentual de projetos realizados com o uso de Métodos Ágeis (Q4)	178
Tabela 31 - Principal critério para mensurar o sucesso de um projeto (Q6)	
Tabela 32 - Desempenho do projeto (Q7)	
Tabela 33 - Grau de Combinação entre os enfoques ágil e clássico de gerenciamento de	e
projetos (Q8)	
Tabela 34 - Envolvimento do <i>champion</i> do sistema (Q9)	
Tabela 35 - Prazo para a entrega do projeto (Q10)	
Tabela 36 – Número de integrantes da equipe de desenvolvimento (Q11)	
Tabela 37 - Percentual de funcionalidades liberadas na primeira versão (Q12)	
Tabela 38 – Respostas das questões 13 a 34	
Tabela 39 - Questão 1 versus Questão 7	
Tabela 40 - Questão 2 versus Questão 7	
Tabela 41 - Questão 3 versus Questão 7	
Tabela 42 - Questão 4 versus Questão 7	
Tabela 43 - Questão 6 versus Questão 7	
Tabela 44 - Questão 8 versus Questão 7	
Tabela 45 - Questão 9 versus Questão 7	182

Tabela 46 - Questão 10 versus Questão 7	182
Tabela 47 - Questão 11 versus Questão 7	
Tabela 48 - Questão 12 versus Questão 7	
Tabela 49 - Questão 13 <i>versus</i> Questão 7	
Tabela 50 - Questão 14 <i>versus</i> Questão 7	
Tabela 51 - Questão 15 <i>versus</i> Questão 7	
Tabela 52 - Questão 16 versus Questão 7	
Tabela 53 - Questão 17 <i>versus</i> Questão 7	
Tabela 54 - Questão 18 <i>versus</i> Questão 7.	
Tabela 55 - Questão 19 <i>versus</i> Questão 7	
Tabela 56 - Questão 20 versus Questão 7.	
Tabela 57 - Questão 21 <i>versus</i> Questão 7	
Tabela 58 - Questão 22 <i>versus</i> Questão 7	
Tabela 59 - Questão 23 <i>versus</i> Questão 7	
Tabela 60 - Questão 24 <i>versus</i> Questão 7	
Tabela 61 - Questão 25 versus Questão 7	
Tabela 62 - Questão 26 versus Questão 7	
Tabela 63 - Questão 27 <i>versus</i> Questão 7	
Tabela 64 - Questão 28 versus Questão 7	
Tabela 65 - Questão 29 versus Questão 7	
Tabela 66 - Questão 30 versus Questão 7	
Tabela 67 - Questão 31 <i>versus</i> Questão 7	
Tabela 68 - Questão 32 <i>versus</i> Questão 7	
Tabela 69 - Questão 33 versus Questão 7	
Tabela 70 - Questão 34 <i>versus</i> Questão 7	
Tabela 71 - Questão 1 versus Questão 8	
Tabela 72 - Questão 2 versus Questão 8	
Tabela 73 - Questão 3 versus Questão 8	
Tabela 74 - Questão 4 <i>versus</i> Questão 8	
Tabela 75 - Questão 6 <i>versus</i> Questão 8	
Tabela 76 - Questão 7 versus Questão 8	
Tabela 77 - Questão 9 versus Questão 8	
Tabela 78 - Questão 10 versus Questão 8.	
Tabela 79 - Questão 11 <i>versus</i> Questão 8	
Tabela 80 - Questão 12 <i>versus</i> Questão 8	
Tabela 81 - Questão 13 <i>versus</i> Questão 8	
Tabela 82 - Questão 14 <i>versus</i> Questão 8	
Tabela 83 - Questão 15 <i>versus</i> Questão 8	
Tabela 84 - Questão 16 <i>versus</i> Questão 8	
Tabela 85 - Questão 17 <i>versus</i> Questão 8	
Tabela 86 - Questão 18 <i>versus</i> Questão 8	
Tabela 87 - Questão 19 <i>versus</i> Questão 8	
Tabela 88 - Questão 20 versus Questão 8.	
Tabela 89 - Questão 21 <i>versus</i> Questão 8	
Tabela 90 - Questão 22 <i>versus</i> Questão 8	
Tabela 91 - Questão 23 <i>versus</i> Questão 8	
Tabela 91 - Questão 23 versus Questão 8	
Tabela 93 - Questão 25 <i>versus</i> Questão 8	
Tabela 94 - Questão 26 <i>versus</i> Questão 8	
Tabela 95 - Questão 27 <i>versus</i> Questão 8	
1 aucia 73 - Questau 21 versus Questau o	170

Tabela 96 - Questão 28 versus Questão 8	196
Tabela 97 - Questão 29 <i>versus</i> Questão 8	197
Tabela 98 - Questão 30 versus Questão 8	197
Tabela 99 - Questão 31 versus Questão 8	197
Tabela 100 - Questão 32 versus Questão 8	197
Tabela 101 - Questão 33 versus Questão 8	198
Tabela 102 - Questão 34 versus Questão 8	198
Tabela 103 - Dez melhores modelos para a predição da Q7	199
Tabela 104 - Resultados da classificação para Q7	200
Tabela 105 - Dez melhores modelos para a predição da Q8	200
Tabela 106 - Resultados da classificação para Q8	201
Tabela 107 - Estimativas dos parâmetros no modelo logístico para Q7	202

# LISTA DE GRÁFICOS

Gráfico 1 – Evolução histórica do desempenho dos projetos de TI	51
Gráfico 2 - Gráficos de <i>boxplot</i> da proporção de acerto para os 9.108 modelos (Q7)	
Gráfico 3 - Gráficos de <i>boxplot</i> da proporção de acerto para os 9.108 modelos (Q8)	

# LISTA DE ILUSTRAÇÕES

Ilustração 1 – Modelo conceitual da pesquisa	19
Ilustração 2 – Fatores críticos de sucesso para o desenvolvimento de novos produtos	23
Ilustração 3 – Triângulo do grupo de processos do gerenciamento de projetos	35
Ilustração 4 – Interação de grupos de processo em um projeto	35
Ilustração 5 – Resumo de alto nível das interações entre os grupos de processo	36
Ilustração 6 – Os três objetivos do projeto: custo-prazo-qualidade	47
Ilustração 7 – Perspectiva do ciclo de vida dos resultados do projeto	53
Ilustração 8 – Indicadores de sucesso de um projeto	55
Ilustração 9 – Custo da mudança do software	66
Ilustração 10 — Esquema dos Crystal Methods	70
Ilustração 11 – Relacionamento entre princípios, práticas, pressupostos e limitações	87
Ilustração 12 - Relacionamento entre as plataformas de gerenciamento clássico e ágil	de
projetos	95
Ilustração 13 – Fluxo do gerenciamento ágil de projetos	100
Ilustração 14 – Fases do gerenciamento ágil de projetos	101

### 1 INTRODUÇÃO

#### 1.1 Contexto e Justificativa

A nova ordem econômica que começou a se formar em fins do século XX e que tem se fortalecido nestes primeiros anos da década de 2000, impulsionada pela tecnologia da informação, caracteriza-se pela globalização dos mercados e do conhecimento. O conhecimento passa a assumir o papel de ativo fundamental e indispensável no novo padrão competitivo estabelecido.

Hamel e Prahalad (1995) mencionam que a busca pela competitividade, antes associada à melhoria contínua da qualidade, à redução de custos e preços, ao aumento da produtividade, à rápida e eficaz introdução de bens tangíveis e intangíveis de alto valor agregado no mercado, passa a depender mais fortemente da criação e renovação de vantagens competitivas associadas ao aprendizado, à qualidade dos recursos humanos e à capacitação produtiva das empresas. Este novo tipo de competitividade baseia-se principalmente na construção de competências essenciais para a aquisição de conhecimentos e de inovação. Prahalad e Ramaswamy (2004, p. 15-16) consideram esse movimento "[...] uma transformação profunda, mas silenciosa, da nossa sociedade."

A tecnologia de informação (TI) é apontada como uma ferramenta extremamente poderosa que impulsionou esse processo de transformação (PORTER; MILLAR, 1985). De acordo com Porter (1989, p. 156) "a tecnologia de sistemas de informação é particularmente penetrante na cadeia de valores, visto que cada atividade de valor cria e utiliza informação". A tecnologia de sistemas de informação também "[...] coordena e otimiza os elos de ligação entre os vários indivíduos e departamentos" (PORTER, *Ibid.*). Sendo assim, a eficiência interna, a capacidade de operação, a habilidade de oferecer produtos e serviços, a agilidade e a flexibilidade, assim como outras vantagens competitivas tão necessárias à sobrevivência e ao crescimento das organizações, atualmente, são fortemente influenciadas pelos sistemas de TI (PORTER, 1989, p. 153-156).

Neste contexto, o desenvolvimento de novos sistemas de TI (doravante denominados software) passou a integrar de forma cada vez mais marcante a agenda das organizações nos últimos anos. Estas iniciativas, em essência similares ao desenvolvimento de um novo produto, começaram a ser conduzidas e gerenciadas sob a forma de projetos e seus resultados passaram a exercer influência direta sobre o sucesso das organizações (SCHILLING; HILL, 1998; KERZNER, 2002).

De acordo com Thomsett (2002), os projetos de desenvolvimento de software têm basicamente duas vertentes – uma técnica e outra gerencial. O autor afirma que, por determinado período, muita atenção foi dada ao aprimoramento dos modelos de desenvolvimento de software (ênfase técnica), ficando o componente gerencial relegado a segundo plano.

De fato, apesar de um início tímido, a importância crescente da tecnologia de informação nas últimas décadas, fez com que vários pesquisadores e praticantes passassem a estudar o tema com maior profundidade, criando modelos específicos para desenvolvimento de software, como por exemplo, os *Modelos em Cascata*, os *Modelos em Espiral* e os processos *iterativos*, entre outros (BECK, 1999; COHEN *et al*, 2003). Estes modelos levam em consideração, além dos aspectos técnicos, as necessidades dos usuários. Apesar de apresentarem algumas diferenças de abordagem, os modelos citados acima, integrantes dos chamados "métodos clássicos de desenvolvimento de software", têm por características principais o foco no planejamento e a ênfase na elaboração de uma documentação bastante completa do software a ser desenvolvido, em uma etapa inicial do projeto (COHEN *et al*, *Ibid.*).

A grande mudança de âmbito gerencial nos projetos de desenvolvimento de software ocorreu, entretanto, em 1986, com a publicação do SW-CMM – *Software Capability Maturity Model* (COHEN *et al*, 2003). Criado pelo SEI – *Software Engineering Institute*, da Carnegie Mellon University, com o intuito de promover o amadurecimento das organizações no processo de desenvolvimento de software, o SW-CMM reúne as melhores práticas de engenharia (desenvolvimento de software) e de gerenciamento de projetos, além de apontar o caminho para o aprimoramento dos processos de construção de software nas organizações (SEI, 1995; PAULK, 2001).

Do ponto de vista do gerenciamento de projetos, o SW-CMM está bastante alinhado aos conceitos do *Gerenciamento Clássico de Projetos*. Cabe salientar que, neste estudo, denomina-se Gerenciamento Clássico de Projetos, o enfoque de gerenciamento de projetos estruturado por meio de processos, que deposita importância fundamental no planejamento e no controle, assim como no rígido gerenciamento de mudanças. Este enfoque tem por base os conceitos defendidos por autores como Verzuh (1999), Kerzner (2002), Maximiano (2002), Dinsmore e Neto (2004). Seu conteúdo principal encontra-se reunido no Guia PMBoK¹ publicado pelo PMI, *Project Management Institute*² (2004) e será apresentado no tópico 2.2 deste trabalho.

De acordo com o STANDISH GROUP INTERNATIONAL (2003), entre 1994 e 2002, muitas organizações investiram na estruturação de práticas relacionadas ao gerenciamento de projetos e na formação de gerentes de projeto visando a uma melhora no desempenho de seus projetos de desenvolvimento de software. Este movimento, que ocorreu em vários outros segmentos de negócio no mesmo período, pode ser verificado pelo número crescente de associados ao PMI e pelo surgimento de grupos de discussão sobre o tema (PMI, 2005).

O mercado de software, considerado atualmente um dos mais importantes do mundo, movimentou cerca de US\$ 190 bilhões em 2004, entre o desenvolvimento e a venda de produtos prontos, de acordo com informações do IDC, *International Data Corporation* (2004). O crescimento anual da parcela relativa ao desenvolvimento de software está estimado em 6,1% para os anos de 2003 a 2008 (IDC, *Ibid.*). Segundo o IDC (2004) este ritmo de crescimento se mostrou real em 2003, entretanto, apesar deste crescimento acelerado, muitas companhias especializadas no desenvolvimento de software têm enfrentado dificuldades, sem atingir níveis adequados de lucratividade ou mesmo sem alcançar um patamar sustentável de vendas. As mesmas empresas reportam problemas constantes quanto à qualidade dos

<sup>&</sup>lt;sup>1</sup> O PMBoK® – *Project Management Body of Knowledge* – é um guia que reúne o conjunto de conhecimentos intrínsecos à profissão de gerenciamento de projetos (PMI, 2004, p. 3).

<sup>&</sup>lt;sup>2</sup> O *Project Management Institute* (PMI®), com mais de 170.000 associados, é hoje a maior entidade mundial sem fins lucrativos voltada ao Gerenciamento de Projetos. Tem atuação global e visa à busca do aprendizado e do desenvolvimento do profissionalismo do Gerenciamento de Projetos como ciência e arte (PMI, 2005).

produtos, ao cumprimento de prazos e dos custos dos projetos de desenvolvimento de software.

Problemas similares são vivenciados por outras organizações, que apesar de não terem no desenvolvimento de software sua atividade principal, realizam esta atividade internamente para atendimento às necessidades de seu negócio. Este cenário está refletido em várias pesquisas, realizadas na última década, para a avaliação do desempenho dos projetos de desenvolvimento de software (JIANG *et al*, 1996; STANDISH GROUP INTERNATIONAL, 1999; 2001; 2003; GIGA INFORMATION GROUP, 2002). De forma geral, os resultados destas pesquisas não foram dos mais animadores: as empresas têm falhado sistematicamente na entrega de seus projetos de desenvolvimento de software. Ressalta-se que nas referidas pesquisas, o sucesso de um projeto é definido em termos do cumprimento da restrição tripla do projeto (MEREDITH; MANTEL, 2000, p.4), ou seja, do alcance dos objetivos de prazo, custo e qualidade.

Citando como exemplo o relatório divulgado pelo STANDISH GROUP INTERNATIONAL<sup>3</sup> (2003), tem-se que apenas 34% dos projetos de desenvolvimento de software podem ser considerados de sucesso ao final de sua execução e que os demais 66% são encerrados antes do planejado ou terminam com resultados considerados insatisfatórios. O mesmo relatório sugere que as principais causas do insucesso dos projetos de desenvolvimento de software não recaem sobre a falta de domínio técnico, mas sim, são atribuídos à ausência ou à inadequação de métodos, técnicas ou práticas de gerenciamento de projetos. Dados mais recentes da mesma pesquisa (relativos ao resultado acumulado até o terceiro trimestre de 2004) apontam um percentual de 29 % de projetos bem-sucedidos, frente a 71% de projetos com resultados insatisfatórios (STANDISH GROUP INTERNATIONAL, 2004).

A análise da série de relatórios *The Chaos Report* publicados pelo STANDISH GROUP INTERNATIONAL (1999; 2001; 2003; 2004) e da literatura correlata (HIGHSMITH, 2004;

<sup>&</sup>lt;sup>3</sup> O STANDISH GROUP INTERNATIONAL é uma entidade de pesquisa norte-americana que avalia o desempenho dos projetos de desenvolvimento de software no mundo, referenciada em algumas publicações sobre o tema (SCOTT, 2001; HIGHSMITH, 2002; JOHNSON, 2002; LAZAREVIC, 2003; AMBLER, 2005).

CHIN, 2004; THOMSETT, 2002; McCABE; POLEN, 2001) sugere a existência de um hiato entre as práticas do gerenciamento clássico de projetos adotadas pelas organizações e suas necessidades propriamente ditas. Chin (*op. cit.*) e Highsmith (*op. cit.*) mencionam que, apesar dos ganhos obtidos, o gerenciamento clássico de projetos não se mostrou plenamente efetivo para os projetos de desenvolvimento de software e que há uma necessidade premente de se repensar a prática de gerenciamento de projetos nas organizações.

Highsmith (2004) e Chin (2004) explicam esta diferença pelo fato de que, usualmente, os projetos de desenvolvimento de software estão inseridos em ambientes de negócio bastante dinâmicos, sujeitos a mudanças constantes, o que foge aos padrões do gerenciamento clássico de projetos. Além disso, o desenvolvimento de software, por ser uma atividade criativa e intelectual, caracterizada por alto grau de inovação, representa um dos grandes desafios dos gerentes e técnicos da área (LAZAREVIC, 2003).

Como resposta às crescentes pressões do mercado e ao desempenho insatisfatório dos projetos de desenvolvimento de software conduzidos com o uso de métodos clássicos de desenvolvimento e gerenciados de acordo com os princípios do gerenciamento clássico de projetos, uma nova abordagem para o desenvolvimento de software foi criada nos últimos anos. Esta reação ocorreu, em princípio, no âmbito técnico, ou seja, na estruturação de novos modelos / métodos de desenvolvimento de software, que não só se diferenciavam dos modelos tradicionais de desenvolvimento de novos produtos, como também contestavam e/ou propunham alternativas aos métodos convencionais de desenvolvimento de software utilizados até meados da década de 90. São os chamados *Métodos Ágeis de Desenvolvimento de Software*, que têm por foco o atendimento das expectativas e das necessidades do cliente, a entrega rápida de valor, o reconhecimento da capacidade dos indivíduos e, principalmente, a adaptação a ambientes de negócio bastante dinâmicos (COHEN *et al*, 2003; BECK *et al*, 2001).

Como exemplos dos principais Métodos Ágeis encontrados na literatura, podem ser citados o Extreme Programming (XP), o Scrum, o Lean Development (LD), o Feature Driven Development (FDD), o Adaptative Software Development (ASD), o Dinamic Software Development Model (DSDM) e os Crystal Methods (AUER; MILLER, 2002; BECK, 2000; BECK; FOWLER, 2001; JEFFRIES et al, 2001; COCKBURN, 2001; SCHAWABER, 2002;

SCHWABER; BEEDLE 2001; HIGHSMITH, 2002, POPPENDIECK; POPPENDIECK, 2003). Todos estes métodos são explicados no tópico 2.4 deste trabalho.

Como evolução natural, uma vez que os projetos de desenvolvimento de software têm sempre uma vertente técnica e outra gerencial, o conceito dos Métodos Ágeis de Desenvolvimento de Software (ênfase técnica) foi expandido, dando origem a um novo enfoque de gerenciamento de projetos – o *Gerenciamento Ágil de Projetos*. Highsmith (2004) e Chin (2004), precursores desta corrente, criticam a postura prescritiva e de conformidade ao plano normalmente adotada por muitos autores e gerentes de projeto e defendem a adoção de um modelo mais ágil e flexível, que tenha por base os seguintes valores: priorizar as respostas às mudanças ao seguimento de um plano, a entrega de produto à entrega de documentação, a colaboração à negociação de contratos e o indivíduo e suas interações aos processos e às ferramentas. Entretanto, apesar de defenderem a agilidade no gerenciamento de projetos, Highsmith (2004) e Chin (2004) afirmam que não se deve perder a essência do gerenciamento de projetos, ou seja a entrega de produtos dentro das restrições de prazo, custo e qualidade, sempre agregando valor à organização. Esta nova visão do gerenciamento de projetos – o Gerenciamento Ágil de Projetos – será abordada de forma detalhada no tópico 2.5 deste documento.

Uma vez que os Métodos Ágeis de Desenvolvimento de Software são relativamente recentes, as publicações disponíveis sobre o assunto abordam basicamente os conceitos e as técnicas, porém poucos são os estudos que constatam e asseguram os resultados reais de sua utilização. Poole e Huisman (2001), Cockburn e Highsmith (2001b), Maurer e Martel (2002), Reifer (2002) e Gawlas (2004) apresentam suas contribuições ao estudo empírico, ao retratarem os benefícios obtidos por algumas organizações ao empregarem os Métodos Ágeis em seus projetos de desenvolvimento de software. Por outro lado, autores como Bohem (2002), Turk et al (2003; 2005), Nerur et al (2005), têm uma visão mais crítica e apontam algumas limitações à aplicação desses métodos. Lazaveric (2003), por sua vez, divulga uma pesquisa voltada à identificação dos fatores críticos para o sucesso de projetos de desenvolvimento de software conduzidos com uso dos Métodos Ágeis de Desenvolvimento de Software. Todos estes pontos estão reunidos no tópico 2.4 desta pesquisa.

Com relação ao Gerenciamento Ágil de Projetos, a situação se repete, porém com um agravante. Por ser ainda mais novo (as primeiras idéias surgiram entre 2002 e 2004), a literatura disponível versa apenas sobre os conceitos e as técnicas, explorando as diferenças

entre os enfoques clássico e ágil de gerenciamento de projetos, com raras recomendações quanto à sua aplicação (THOMSETT, 2002; HIGHSMITH, 2004; CHIN, 2004). Até o momento do encerramento desta dissertação, não foi constato nenhum estudo que relatasse os resultados da utilização do Gerenciamento Ágil de Projetos, ou uma avaliação dos fatores críticos de sucesso de projetos desta natureza. Este fato é um dos motivadores da realização desta pesquisa. Ressalta-se que as informações conceituais relativas ao Gerenciamento Ágil de Projetos pesquisadas na literatura encontram-se consolidadas no tópico 2.5 deste trabalho.

Enfocando a realidade brasileira, de acordo com o ministro do Desenvolvimento, Indústria e Comércio, para crescer e se desenvolver, o Brasil precisa evoluir em áreas de maior valor agregado e deve investir na exportação do conhecimento (BRASIL, 2005a). Neste contexto, a indústria de software nacional passou a ser foco de atenção, sendo incluída na política industrial do atual governo. Segundo informações da Agência Brasil (BRASIL, 2005a), o governo tem a intenção de que em 2007, as exportações de software e serviços de informática cheguem à cifra de US\$ 2 bilhões e, para tanto, foram tomadas várias medidas de estímulo ao crescimento. Em 2004, um levantamento que teve apoio da Assespro, Associação das Empresas Brasileiras de Tecnologia da Informação, Software e Internet de São Paulo e do ITS, Instituto de Tecnologia de Software de São Paulo, indicou um volume exportado no período de cerca da US\$ 235 milhões (BRASIL, *Ibid.*). Além da contribuição relativa ao volume das exportações, o segmento de software ainda é visto com especial atenção pelo governo, por seu grande potencial de geração de empregos e por sua capacidade de absorção de mão-de-obra jovem, recém-saída das universidades (BRASIL, 2005a).

Entretanto, aumentar o volume de exportações num mercado competitivo como o de software, não é uma tarefa simples. A indústria brasileira de software precisa adquirir um padrão de competitividade internacional e a aquisição desta competitividade passa por um esforço sustentado rumo à certificação da qualidade dos produtos e serviços e à formação da imagem ou da identidade do software do país. O êxito das empresas indianas, que associaram sua imagem à maturidade de seus processos de desenvolvimento de software (SW-CMM), só faz aumentar a barreira de entrada do Brasil neste mercado (MIT; SOFTEX CAMPINAS, 2003, p. 69).

Dado que a estratégia de vincular a imagem a um modelo de maturidade de desenvolvimento de software já foi utilizada de forma bem-sucedida por outro país emergente – a Índia – a

valorização da criatividade dos profissionais brasileiros e a adoção da agilidade como um diferencial, são apontadas como alguns caminhos a trilhar pelo Brasil (MIT; SOFTEX CAMPINAS, 2003, p. 69). Neste contexto, a utilização dos Métodos Ágeis de Desenvolvimento de Software pode fornecer as bases para a consolidação desta estratégia (BECK; MEE, 2003).

Apesar da última Pesquisa sobre Qualidade e Produtividade do Setor de Software Brasileiro, ano base 2004, divulgada pelo MCT, Ministério da Ciência e Tecnologia, ainda não incorporar formalmente os Métodos Ágeis de Desenvolvimento de Software, é possível verificar que eles já fazem parte da realidade brasileira há algum tempo (BRASIL, 2005b). Desde 2001, os Métodos Ágeis são objeto de estudo em universidades (como por exemplo, a Pontifícia Universidade Católica do Rio de Janeiro, a Universidade de São Paulo, a Universidade Federal de São Carlos, a Universidade Federal do Ceará, a Universidade de Uberlândia, entre outras), constituem tema de grupos de discussão na internet e integram a agenda de congressos e seminários da área, como o Simpósio Brasileiro de Engenharia de Software – SBES ou os patrocinados pela SUCESU<sup>4</sup> ou pela iniciativa privada (EXTREME PROGRAMMING, 2002; SBES, 2003; 2005; UFBA, 2003; USP, 2004; SUGAR LOAF PLOP, 2005; PUC-RJ, 2004; SUCESU, 2005a).

Finalmente, explorando a questão relativa à melhor abordagem de gerenciamento de projetos (ágil ou clássica) para o desenvolvimento de software com o emprego de Métodos Ágeis, não foi encontrada na literatura uma resposta única, completa e definitiva. Isto, apesar da unanimidade dos autores pesquisados em ressaltar a importância do papel do gerenciamento de projetos para o sucesso das iniciativas de desenvolvimento de software.

Thomsett (2002), Highsmith (2004) e Chin (2004) sugerem, de forma genérica, que o Gerenciamento Ágil de Projetos, é indicado para o desenvolvimento de software, como também para quaisquer outros projetos de desenvolvimento de novos produtos. Thomsett (2002) chega a mencionar que as práticas do Gerenciamento Ágil de Projetos poderiam ser

<sup>&</sup>lt;sup>4</sup> A SUCESU é uma sociedade civil sem fins lucrativos, representativa dos diversos setores de Tecnologia da Informação e Comunicação, que atua em vários estados do Brasil.

utilizadas até mesmo no desenvolvimento de software conduzido segundo os métodos clássicos. Em contrapartida, Paulk (2001) menciona que os Métodos Ágeis e o SW-CMM (que tem por base o gerenciamento clássico de projetos) não são incompatíveis, ou seja, o autor sugere que o gerenciamento clássico de projetos poder ser aplicado no desenvolvimento de software realizado com o uso de Métodos Ágeis. Enfim, há posições bastante distintas sobre o assunto, sendo que não foi encontrada nenhuma pesquisa de campo que abordasse esta questão de forma direta, até o momento do encerramento desta dissertação. O estudo mais próximo a esta discussão foi desenvolvido por Lazarevic (2003), ao investigar os fatores críticos de sucesso para o desenvolvimento de software realizado com o uso de Métodos Ágeis.

Outrossim, a realização da presente pesquisa, de caráter exploratório-descritivo, é justificada ao se considerar:

- a) A inegável importância da tecnologia de informação no cenário de transformação sócioeconômico mundial e nacional;
- b) Que uma parcela considerável dos projetos de desenvolvimento de software não atinge os objetivos inicialmente propostos;
- c) A percepção da existência de um hiato entre a prática atual de gerenciamento de projetos e a necessidade das organizações;
- d) O surgimento dos Métodos Ágeis de Desenvolvimento de Software e do Gerenciamento Ágil de Projetos, como soluções promissoras, visando a melhorar o desempenho dos projetos de desenvolvimento de software;
- e) O caráter inovador do Gerenciamento Ágil de Projetos;
- f) A carência de pesquisas empíricas sobre o tema.

Finalizando, este trabalho foi delineado no intuito de contribuir para as pesquisas nessa temática, provendo evidências empíricas da realidade brasileira.

#### 1.2 Pergunta de Pesquisa

Dados o contexto e a justificativa discutidos no tópico anterior, apresenta-se aqui a perguntabase desta pesquisa:

Qual o enfoque de gerenciamento de projetos mais apropriado para o desenvolvimento de software conduzido com o uso de um Método Ágil?

Para garantir o perfeito entendimento da pergunta, deve-se considerar as seguintes definições operacionais:

- São chamados *Métodos Ágeis de Desenvolvimento de Software* aqueles que têm por foco: a adaptação dos requisitos do software às necessidades do negócio e não a entrega de especificações pré-definidas; a ênfase nas pessoas e não nos processos; e, a ênfase na inovação, na qualidade e na entrega de valor e não na documentação (COCKBURN, 2001; BECK *et al*, 2001; UDO; KOPPENSTEINER, 2003; LAZAREVIC, 2003 HIGHSMITH, 2004). Entre os Métodos Ágeis pesquisados estão o *Extreme Programming* (XP), *Scrum*, *Feature Driven Development* (FDD), *Adaptative Software Development* (ASD), *Crystal Methods*, todos eles detalhados no tópico 2.4.3 deste trabalho.
- Com relação aos enfoques de gerenciamento de projetos, são considerados os conceitos e técnicas de duas abordagens principais:
- a) O *Gerenciamento Clássico de Projetos*, estruturado segundo uma visão de processos, conforme proposto pelo PMI (2004) ver tópico 2.2;
- b) O *Gerenciamento Ágil de Projetos*, advindo de uma evolução do conceito dos Métodos Ágeis de Desenvolvimento de Software, trazendo em si uma maior agilidade e flexibilidade (HIGHSMITH, 2004; CHIN, 2004) ver tópico 2.5.
- Por enfoque de gerenciamento de projetos "mais apropriado" entende-se aquele apontado por profissionais que classificaram como "bem-sucedido" seu projeto de desenvolvimento de software conduzido com o uso de um Método Ágil.

#### 1.3 Objetivos e Contribuições

Os objetivos primários deste trabalho, que visa a responder a pergunta de pesquisa apresentada anteriormente, são:

- a) Realizar, à luz do referencial bibliográfico, um estudo comparativo entre o gerenciamento clássico e o gerenciamento ágil de projetos;
- b) Identificar o enfoque de gerenciamento de projetos mais apropriado clássico ou ágil para o desenvolvimento de software conduzido com o uso de um Método Ágil.

E, como objetivos secundários da pesquisa, podem ser citados:

- a) Investigar os fatores críticos de sucesso dos projetos de desenvolvimento de software realizados com o emprego dos Métodos Ágeis;
- b) Levantar subsídios para a realização de uma pesquisa futura mais aprofundado sobre o tema;
- c) Contribuir para a ampliação do conhecimento em Administração de Projetos.

#### 1.4 Delimitação do Estudo

Pela amplitude do tema escolhido – Gerenciamento de Projetos – foi necessário delimitar o campo e a dimensão do estudo, buscando garantir o perfeito entendimento do problema e o cumprimento dos objetivos primários e secundários da pesquisa. Procedeu-se, então, a um recorte do estudo em termos dos enfoques de gerenciamento de projetos (caracterizados por seus conceitos e técnicas) para o desenvolvimento de software conduzido com a utilização de um Método Ágil.

Vale ressaltar que dentre os vários setores passíveis de estudo, o setor de Tecnologia de Informação, em especial o segmento de desenvolvimento de software, foi escolhido por sua inegável contribuição ao processo de transformação sócio-econômico mundial e por sua importância no mercado nacional. Além disto, a escolha de projetos de desenvolvimento de

software se deu, também, em virtude dos baixos índices de sucesso alcançados por projetos desta natureza.

Ainda com relação ao estudo, deve-se salientar que, devido às características estipuladas para a condução da pesquisa, descritas no Capítulo 3 – *Metodologia de Pesquisa*, as conclusões alcançadas devem ficar limitadas ao âmbito deste trabalho, não podendo ser generalizadas ou automaticamente inferidas para situações distintas das aqui abordadas. Ao mesmo tempo, entende-se que ao atingir seus objetivos, este trabalho gera os subsídios necessários para a realização de estudos posteriores, mais amplos, contribuindo para o avanço contínuo da ciência da administração.

#### 1.5 Modelo Conceitual

Uma vez delimitado o estudo, parte-se para a apresentação do modelo conceitual seguido durante a realização deste projeto de pesquisa, que é exposto na Ilustração 1.

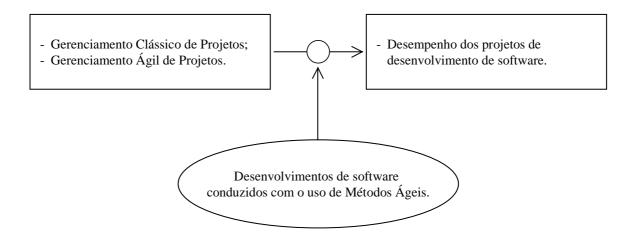


Ilustração 1 – Modelo conceitual da pesquisa

Neste modelo, como variáveis independentes têm-se os enfoques clássico e ágil de gerenciamento de projetos. O desempenho dos projetos de desenvolvimento de software corresponde à variável dependente. As variáveis intervenientes correspondem às características e técnicas dos desenvolvimentos de software realizados com o uso de Métodos

Ágeis. A descrição de cada uma destas variáveis, assim como a apresentação completa da metodologia da pesquisa, é feita no Capítulo 3.

#### 1.6 Estrutura do Trabalho

Com relação à estrutura, esta dissertação está dividida nos seguintes capítulos: 1) *Introdução*, que contém a contextualização e a justificativa do trabalho, a pergunta de pesquisa, os objetivos e as contribuições, a delimitação do estudo e seu modelo conceitual; 2) *Revisão Bibliográfica*, capítulo em que se concentram as contribuições da literatura concernentes ao assunto em estudo; 3) *Metodologia de Pesquisa*, que descreve os fundamentos metodológicos que nortearam o desenvolvimento do trabalho; 4) *Resultados e Análise*, capítulo em que são apresentados e analisados os resultados obtidos na pesquisa; 5) *Conclusões*, contendo as conclusões decorrentes da análise dos resultados e as considerações finais deste trabalho.

Uma vez apresentados o contexto, a justificativa, a pergunta de pesquisa, os objetivos e as contribuições do trabalho, a delimitação do estudo e seu modelo conceitual, parte-se para a revisão bibliográfica, que contempla a base teórica relativa ao tema da pesquisa.

#### 2 REVISÃO BIBLIOGRÁFICA

Neste capítulo é feita a exposição da revisão da literatura sobre o tema da pesquisa. São abordados: o desenvolvimento de um novo produto ou novo software (software), os métodos ágeis de desenvolvimento de software, o gerenciamento clássico de projetos, o gerenciamento ágil de projetos e assuntos correlatos. Este conjunto de conhecimento foi extraído das obras de vários autores, aportando sustentação a este trabalho e trazendo uma contribuição ao meio acadêmico, quando de sua apresentação consolidada.

#### 2.1 Desenvolvimento de Software

#### 2.1.1 Processo de Desenvolvimento de um Novo Produto ou Software

O desenvolvimento de novos produtos tem se tornado um ponto focal na competição industrial. A origem desta tendência está associada a três grandes forças que emergiram no pós-guerra (II Guerra Mundial) e mais intensamente ao longo das décadas de 70 e 80, em muitas indústrias em todo o mundo: a ampliação do espaço competitivo para além das fronteiras nacionais; o surgimento de mercados fragmentados, com demandas específicas; a aceleração do processo de mudança técnica (rápida obsolescência), com o aparecimento de tecnologias transformadoras, como a microeletrônica e as tecnologias da informação e da comunicação.

Novas tecnologias e novos entendimentos sobre as tecnologias existentes têm produzido uma maior e mais profunda base de conhecimentos sobre os fenômenos subjacentes a aplicações particulares, potencializando a capacidade de criar novas opções ajustadas às necessidades específicas de clientes ou consumidores mais sofisticados.

O desenvolvimento de novos produtos tem feito diferença na competitividade da empresa e de seus produtos no longo prazo. O sucesso no desenvolvimento de novos produtos parece estar relacionado com o "[...] padrão de consistência global do sistema de desenvolvimento,

incluindo estrutura organizacional, habilidades técnicas, processo de resolução de problemas, cultura e estratégia. Tal consistência jaz não só sobre os princípios e a arquitetura do sistema, mas também sobre os detalhes no nível operacional e de gestão" (CLARK; FUJIMOTO, 1991, p.7).

Para os gerentes seniores em todo o mundo, desenvolver melhores produtos, mais rápida, eficiente e eficazmente é uma questão central na agenda competitiva. Uma evidência é que projetar e desenvolver novos produtos de forma eficiente e eficaz tem significativo impacto nos custos, na qualidade, na satisfação do consumidor e na vantagem competitiva (CLARK; FUJIMOTO, *Ibid.*).

Empresas bem-sucedidas neste padrão competitivo são aquelas que conseguem articular de forma adequada seus objetivos estratégicos e estruturar e gerir seus portafólios de Pesquisa e Desenvolvimento (P&D), de modo a ajustá-los às metas de desenvolvimento de novos produtos e aos recursos e às competências disponíveis interna e externamente. O sucesso destas empresas também depende de quão bem as áreas tecnológicas em que entram, contribuem para a sua orientação no longo prazo, permitindo-lhes construir novas capacidades essenciais críticas, reduzir progressivamente o tempo de desenvolvimento de novos produtos, atendendo satisfatoriamente aos requisitos do mercado e às especificações do produto, sem sacrificar a qualidade (SCHILLING; HILL, 1998).

Esses autores sugerem, como capacidades essenciais críticas no processo de desenvolvimento de novos produtos, fatores ligados a quatro dimensões básicas: a estratégia tecnológica, o contexto organizacional, as equipes de projetos e as ferramentas para melhorar continuamente o processo (Ilustração 2). Através deste modelo, percebe-se que Schilling e Hill (*Ibid.*) propõem a condução das iniciativas de desenvolvimento de novos produtos sob a forma de projetos. A definição detalhada de projetos é feita no tópico 2.2.1 deste trabalho.

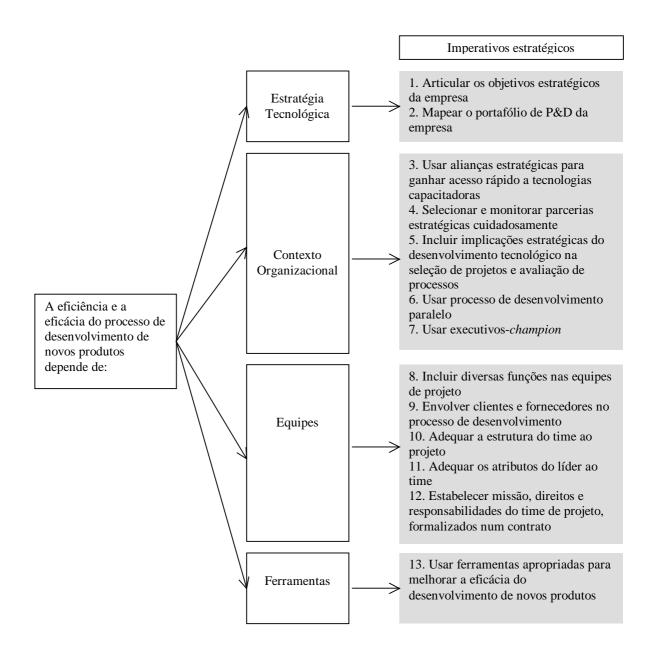


Ilustração 2 – Fatores críticos de sucesso para o desenvolvimento de novos produtos FONTE: ADAPTADO DE SCHILLING; HILL, 1998, p.70

A construção e o uso de equipes para o desenvolvimento de novos produtos têm sido alvo de muitas pesquisas que, consensualmente, apontam para a importância da multifuncionalidade, inclusive da participação de clientes e fornecedores, para o sucesso dos empreendimentos. Os motivos alegados para justificar tal hipótese recaem sobre a melhor adequação do produto em desenvolvimento às potencialidades da cadeia de suprimento, às necessidades do comprador e/ou usuário e à "manufaturabilidade" nas linhas de produção da empresa, em virtude da

superação de lapsos de comunicação existentes num processo função-a-função e do aproveitamento da "fertilização cruzada de idéias".

Ainda sobre este aspecto, Schilling e Hill (1998) frisam a importância de se adequar a estrutura da equipe – funcional, peso-leve, peso-pesado ou autônoma – ao tipo de projeto, uma vez que os níveis de coordenação e comunicação em cada tipo de projeto e estrutura são diferentes e podem ser compatibilizados. Similarmente, os atributos de liderança do gerente do projeto ou executivos-*champion* - como a habilidade de gerir conflitos e de comunicar-se com diversas áreas, entre outras - devem estar alinhados com o tipo de projeto.

Outros requisitos concernentes à formação e operação de equipes de projeto são o estabelecimento de missão, de objetivos e de responsabilidades, bem como o detalhamento das atividades a serem desenvolvidas e dos resultados a serem alcançados, garantindo que sejam claramente comunicados. Estas são maneiras de prover-lhes foco bem definido e estimular seu comprometimento com o desenvolvimento do produto, além de constituir uma ferramenta para o monitoramento e a avaliação do desempenho das equipes (SCHILLING; HILL, 1998).

Enfocando o desenvolvimento de um novo software, definido como o processo através do qual um novo software é concebido, codificado, testado e disponibilizado ao mercado (LAZAREVIC, 2003, p. 4), percebe-se que todas as colocações e preocupações anteriormente expostas, para o desenvolvimento de novos produtos, são totalmente aplicáveis, uma vez que um software não deixa de ser um produto.

Nos últimos anos, o desenvolvimento de software ganhou cada vez mais destaque nas organizações. Porter e Millar (1985) e Booch (2001) apontam a tecnologia de informação como uma ferramenta extremamente poderosa que impulsionou o processo de transformação da sociedade e da economia. Seu valor e importância podem ser percebidos à medida que "[...] a tecnologia penetra na cadeia de valores de uma empresa e extrapola as tecnologias associadas diretamente ao produto" (PORTER, 1989, p. 153). Cada atividade de valor sempre cria ou utiliza uma informação e a tecnologia de informação coordena e otimiza os elos de ligação entre os vários indivíduos e departamentos. Porter (1989, p. 153-156) enfatiza que a eficiência interna, a capacidade de operação, a habilidade de oferecer produtos e serviços, a agilidade e a flexibilidade, assim como outras vantagens competitivas tão necessárias à

sobrevivência e ao crescimento das organizações são, atualmente, fortemente influenciadas pelos sistemas de informação.

Atualmente, a indústria de software é uma das mais importantes do mundo. Sua existência tornou possível, direta ou indiretamente, o surgimento de novos negócios e o aumento da eficiência dos negócios tradicionais (BOOCH, 2001). Marcada por uma forte concorrência, movimentou cerca de US\$ 190 bilhões em 2004, entre desenvolvimento e venda de produtos prontos. Estima-se que, entre 2003 e 2008, a parcela relativa ao desenvolvimento de software apresente um crescimento anual de aproximadamente 6,1% (IDC, 2004).

De acordo com Veloso *et al* (2003), a indústria de software é dominada pelas grandes nações desenvolvidas, entre elas, os Estados Unidos, a Alemanha e o Japão, respondendo por cerca de 2% do PIB destes países. Em outros países, como Israel e Irlanda, esta contribuição é ainda maior, representando em 2001 cerca de 3,4% e 7,4% do PIB do país, respectivamente.

O advento da internet e da globalização, impulsionado pela própria tecnologia de informação, ocasionou mudanças significativas no mercado de desenvolvimento de software. Para competir na economia digital, as empresas têm de ser capazes de desenvolver um software, de alto valor agregado ao cliente, com elevado padrão de qualidade, a uma velocidade cada vez maior e ainda ser capazes de responder às mudanças constantes no mundo dos negócios (BASKERVILLE *et al*, 2003; MAURER; MARTEL, 2002; COCKBURN; HIGHSMITH, 2001a). Laitinen *et al* (2000) destacam que a internet permitiu que pequenas organizações conquistassem seu espaço, no disputado mercado de desenvolvimento de software, oferecendo produtos de valor. No Brasil, a internet propiciou a criação de milhares de empregos (BRASIL, 2001a).

#### 2.1.2 Desenvolvimento de Software no Brasil

Em 2001, o mercado brasileiro de software estava estimado em US\$ 7,7 bilhões (equivalente a 1,5% do PIB daquele ano) e se caracterizava por uma forte dependência de fornecedores externos. A maioria dos centros de tecnologia nacionais trabalhava com um volume escasso de recursos, não suficiente para propiciar o aprimoramento da prática e o incentivo à exportação (MIT; SOFTEX, 2003). Apesar deste cenário não muito favorável, o mesmo

estudo apontava uma oportunidade excepcional para o Brasil atuar como provedor de serviços de software no mercado mundial, dada a experiência adquirida com os projetos realizados em clientes exigentes e sofisticados.

A taxa de crescimento do mercado brasileiro de desenvolvimento de software tem sido estimada em um patamar superior ao da média mundial. Como exemplo, para 2004, o IDC estimou um crescimento de 7,1% para o segmento no Brasil, frente aos 6,1% projetados mundialmente (IDC, 2004). Apesar desta diferença ser explicada até o momento pelo crescimento do mercado interno, o incremento do volume de exportações pode fazer com esta variação se acentue. Segundo dados do próprio IDC (*Ibid.*), a importação de produtos e serviços de software pelos países ricos é uma tendência em franca expansão: nos Estados Unidos esta tem aumentado a taxas anuais na ordem de 25%; na Europa, os gastos com terceirização devem dobrar entre 2002 e 2005, em especial, nas áreas de administração de estoques, recursos humanos e sistemas de pagamento (neste último item, o Brasil é tido como referência mundial).

De acordo com o MCT (BRASIL, 2002), desde o ano 2000, foram criados vários programas para o aprimoramento do setor de software brasileiro e para o incentivo às exportações. Entre estes programas pode ser destacado o SOFTEX – Programa para Promoção da Excelência do Software Brasileiro (anteriormente conhecido como Programa Nacional do Software para Exportação), que reúne a maioria das empresas brasileiras exportadoras de produtos e serviços de software. Entretanto, foi a partir de 2004 que o desenvolvimento de software passou a integrar formalmente a política de desenvolvimento, indústria e comércio exterior do governo federal (BRASIL, 2005a). Além de estar totalmente alinhado à proposta defendida pelo Ministério do Desenvolvimento, Indústria e Comércio, segundo a qual o Brasil precisa evoluir em áreas de maior valor agregado e deve investir na exportação do conhecimento para poder crescer e se desenvolver, o setor de software ainda desperta interesse especial, dado o seu potencial de geração de empregos e de absorção de mão-de-obra recém-saída das universidades (BRASIL, 2005a).

O Brasil possui atualmente 3.265 empresas de software (muitas delas formadas pela parceria estratégica entre investidores com experiência no mercado de tecnologia de informação, professores e universitários da área da computação), mas apenas 2,2% delas, ou seja, 71 companhias, exportaram seus produtos ou serviços em 2004 (BRASIL, 2005a). De acordo

com o levantamento que teve apoio da Assespro, Associação das Empresas Brasileiras de Tecnologia da Informação, Software e Internet de São Paulo e do ITS, Instituto de Tecnologia de Software de São Paulo, o volume exportado no período atingiu US\$ 235 milhões, sendo que o segmento de software respondeu por US\$ 110 milhões (46,8%), enquanto os serviços e a alocação de mão-de-obra foram responsáveis pelos US\$125 milhões restantes. A estimativa de exportação para 2005, com base nas 71 exportadoras, é de 280 milhões de dólares, crescimento de cerca de 19% frente ao ano passado (BRASIL, 2005a).

Segundo informações da Agência Nacional (BRASIL, 2005a), o governo tem a intenção de que em 2007 as exportações de software e serviços de informática cheguem à cifra de US\$ 2 bilhões e, para tanto, vem tomando uma série de medidas para propiciar o cumprimento desta meta, entre as quais podem ser citadas:

- Financiamento de R\$ 100 milhões em 2004, concedido pelo BNDES, voltado à produção, comercialização e exportação de software. O recurso faz parte do novo Programa para o Desenvolvimento da Indústria Nacional de Software e Serviços Correlatos (PROSOFT);
- Novo Programa Nacional de Certificação em Software e Serviços, executado pelo Inmetro, por institutos de pesquisa e por entidades acadêmicas, que devem certificar empresas para assegurar reconhecimento de qualidade de produção;
- Programa de Exportação de Software e Serviços, voltado a projetos de terceirização, plataformas de exportação e consórcios de empresas, com contratação de estudos de mercado para definição de estratégias;
- Plano de Incentivo ao Desenvolvimento de Software Livre, ampliando oferta de soluções baseadas em código aberto;
- Biblioteca compartilhada de componentes, para tornar mais ágil o desenvolvimento de produtos e reduzir seus custos;
- Realização de fóruns de Tecnologia de Informação, reunindo governo, empresas e institutos de pesquisa;
- Lei de Inovação, que tem o objetivo de criar condições para que a taxa de investimento em P&D aumente nas empresas, integrando esforços com universidades e institutos de pesquisa.

Entretanto, aumentar o volume de exportações num mercado competitivo como o de desenvolvimento de software, considerando ainda o histórico de escassez de recursos nos pólos de tecnologia nacional, não é uma tarefa simples. Qualquer projeto de internacionalização do software brasileiro deve passar, necessariamente, por um exaustivo trabalho de qualificação do produto, de formação da identidade do produto e de valorização profissional (MIT; SOFTEX, 2003). A Índia já trilhou este caminho de forma bem-sucedida há algum tempo, associando a sua imagem à maturidade de seus processos de desenvolvimento de software, o que constitui, de certa forma, uma barreira à entrada do Brasil neste mercado. Atualmente a Índia exporta cerca de US\$ 8 bilhões anuais, enquanto o Brasil embolsa modestos US\$ 110 milhões. Em 2008, estima-se que o valor das exportações indianas de software alcance o patamar de U\$ 57 bilhões (MIT; SOFTEX, 2003; BRASIL, 2005a).

Uma vez que a estratégia de associar a imagem a um modelo de maturidade de processos<sup>5</sup> de desenvolvimento de software já foi utilizada por outro país, a valorização da competência e da criatividade dos profissionais brasileiros, aliada à adoção da agilidade como um diferencial, é apontada como um dos caminhos a trilhar pelo Brasil por estudiosos como Beck e Mee (2003). O Brasil possui mão-de-obra qualificada e criativa, a um custo bastante competitivo. Além disso, possui experiência e competência reconhecida mundialmente no desenvolvimento de software para alguns segmentos, como o bancário e de meios de pagamentos eletrônicos. Adicionalmente, as empresas têm se mostrado bastante ágeis para lidar com situações adversas. Neste contexto, Beck e Mee (*Ibid.*) apontam que a utilização dos Métodos Ágeis de Desenvolvimento de Software pode fornecer as bases para a consolidação desta estratégia.

Finalizando, apesar da última pesquisa sobre qualidade e produtividade do setor de software brasileiro (ano base 2004), conduzida pelo MCT (BRASIL, 2005b), ainda não incorporar os Métodos Ágeis de Desenvolvimento de Software, é possível verificar que estes já fazem parte

<sup>&</sup>lt;sup>5</sup> O modelo de maturidade de desenvolvimento de software aqui mencionado tem por base o SW-CMM (Software Capability Maturity Model), criado pelo SEI, Software Engineering Institute da Carnegie Mellon University, em 1986.

da realidade brasileira há algum tempo. Conforme mencionado, desde 2001, são objeto de estudo em universidades (como por exemplo, a Pontifícia Universidade Católica do Rio de Janeiro, a Universidade de São Paulo, a Universidade Federal de São Carlos, a Universidade Federal do Ceará, a Universidade de Uberlândia, entre outras), constituem tema de discussão em grupos especializados na internet, além de estar presente na agenda de congressos e seminários da área, como os Simpósios Brasileiros de Engenharia de Software – SBES, os congressos ou encontros patrocinados pela SUCESU, ou mesmo os organizados pela iniciativa privada (SUGAR LOAF PLOP, 2005; USP, 2004; UFBA, 2003; SBES, 2003; 2005; PUC-RJ, 2004; EXTREME PROGRAMMING BRASIL, 2002; SUCESU, 2005a).

#### 2.2 Gerenciamento Clássico de Projetos

Retomando os quatro fatores críticos de sucesso para o desenvolvimento de novos produtos propostos por Schilling e Hill (1998, p.70): estratégia tecnológica, contexto organizacional, equipes e ferramentas, percebe-se que o desenvolvimento de novos produtos (ou software) pode e deve ser gerenciado com um projeto. Considerando que as iniciativas de desenvolvimento de software são conduzidas sob a forma de projetos e que do sucesso destes projetos pode depender o sucesso das organizações (SCHILLING; HILL, 1998; KERZNER, 2002), são apresentados neste tópico os principais conceitos referentes ao gerenciamento de projetos, aqui denominado Gerenciamento Clássico de Projetos.

Ressalta-se que o termo "clássico" é utilizado neste momento para diferenciar esta abordagem do Gerenciamento Ágil de Projetos, a ser explorado em outro no tópico 2.5. Esta é a nomenclatura adotada por Chin (2004) e Highsmith (2004) ao se referirem o enfoque do gerenciamento de projetos estruturado por processos, como o proposto pelo PMI (2004).

## 2.2.1 Projetos e o Gerenciamento Clássico de Projetos

Para melhor entender o gerenciamento de projetos, em primeiro lugar é preciso reconhecer o que é um projeto. Apesar dos projetos existirem desde os tempos remotos, foi a partir da década de 60, que o tema passou a despertar maior interesse e ganhar popularidade (MEREDITH; MANTEL, 2000, vii). Embora a literatura apresente várias definições (LEWIS,

1997, p. 1; VERZUH, 1999, p. 3; GRAY; LARSON, 2003, p. 5; KERZNER, 2002, p. 17; MEREDITH; MANTEL, 2000, p. 9; MAXIMIANO (2002, p. 26); ISO, 1997; PMI, 2004, p. 5), uma análise mais cuidadosa nos faz perceber que há pouca ou quase nenhuma variação em termos conceituais entre elas.

Segundo o PMI (2004, p. 5) "Um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo". *Temporário* significa que todos os projetos possuem um início e um final definidos. *Exclusivo* ou singular, significa que o produto, serviço ou resultado gerado é, de alguma maneira, diferente de todos os outros produtos, serviços ou resultados já existentes. Estas entregas exclusivas podem ser:

- Um produto ou objeto produzido, quantificável, seja ele um item final ou um componente;
- Uma capacidade de realizar um serviço, como funções de negócio que dão suporte à produção ou à distribuição;
- Um resultado, como resultados finais ou documentos.

Como os projetos envolvem a realização de algo que jamais foi feito anteriormente, a eles pode ser associado um certo grau de complexidade e incerteza (*Ibid.*, p. 6). Os projetos têm atributos marcantes que os diferenciam das atividades do dia-a-dia ou das operações continuadas, distinção esta que deve ser perfeitamente assimilada. Segundo o PMI (2004, p. 7), o propósito de um projeto é alcançar o seu objetivo declarado e então ser encerrado, enquanto a operação continuada tem, normalmente, por finalidade, a sustentação do negócio. Martin e Tate (2001, p. 8) ressaltam o fato dos projetos serem temporários e produzirem resultados únicos, em contraponto com as operações continuadas, em que o mesmo processo é repetido várias vezes, com objetivo de produzir os mesmos resultados a cada execução. Ainda, o plano de trabalho de um projeto é incerto, requerendo atualizações constantes, enquanto nas operações continuadas, o plano é bem definido.

É importante mencionar ainda, que os projetos são geralmente implementados para que o plano estratégico de uma organização seja cumprido, sendo normalmente autorizados como resultado de uma ou mais definições estratégicas (PMI, 2004, p. 7; KERZNER, 2002, p. 17), a saber:

- Uma demanda de mercado;
- Uma necessidade organizacional;
- Uma solicitação de um cliente;
- Um avanço tecnológico;
- Um requisito legal.

Para que suas atividades sejam realizadas e seus objetivos atingidos, os projetos demandam um determinado esforço de gerenciamento. A partir da década de 60, o gerenciamento de projetos passou a ser reconhecido como uma disciplina e sua prática foi estruturada (MEREDITH; MANTEL, 2000; THOMSETT, 2002). Atualmente, diversas empresas a usam regularmente, como forma de atingir seus objetivos de negócio, dado um número limitado de recursos.

O Gerenciamento Clássico de Projetos foi objeto de estudo de vários autores e pesquisadores, que apresentaram suas definições e visões sobre o tema (VERZUH, 1999, p. 19; KERZNER, 2002, p. 4-5; DINSMORE; NETO, 2004, p. 1; MAXIMIANO, 2002, p. 40). Estes autores compartilham a mesma linha conceitual – a estruturação do gerenciamento de projetos por meio de processos – que está alinhada à linha divulgada pelo PMI (2004, p. 8).

Preocupado com a padronização de conceitos, mas também com a aplicação prática, o PMI (2004, p. 8) descreve o gerenciamento de projetos como "[...] a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto, a fim de atender aos seus requisitos". Destaca que o "gerenciamento de projetos é realizado através da aplicação e da integração dos seguintes processos de gerenciamento de projetos: iniciação, planejamento, execução, monitoramento e controle e encerramento" (PMI, *Ibid.*).

Segundo o PMI (*Ibid.*), gerenciar um projeto inclui:

- A identificação das necessidades;
- O estabelecimento de objetivos claros e alcançáveis;
- O balanceamento de demandas conflitantes de qualidade, escopo, tempo e custo;
- A adaptação das especificações, dos planos e da abordagem às diferentes preocupações e expectativas das diversas partes interessadas.

O gerenciamento de projetos provê a empresa de ferramentas poderosas que melhoram a habilidade de planejamento, organização, execução e controle das atividades de maneira a conseguir atingir os resultados esperados, dentro do prazo e custo previstos, mesmo em casos de grande complexidade (MEREDITH; MANTEL, 2000).

Kerzner (2002, p. 17) complementa que, para ser bem-sucedida, a gestão de projetos<sup>6</sup> demanda um fluxo de trabalho e uma coordenação horizontal (e não mais vertical como na gerência tradicional), com ênfase na comunicação, no aumento da produtividade, eficácia e eficiência, com destaque especial ao papel e às atribuições do gerente de projeto.

Uma vez conceituados projeto e gerenciamento de projetos, parte-se agora para a apresentação de uma visão geral dos processos do Gerenciamento Clássico de Projetos.

# 2.2.2 Processos do Gerenciamento Clássico de Projetos

Conforme mencionado no tópico anterior, alguns autores tratam o gerenciamento de projetos segundo um enfoque de processos (VERZUH, 1999, p. 19; KERZNER, 2002, p. 4-5; DINSMORE; NETO, 2004, p. 1; MAXIMIANO, 2002, p. 40; PMI, 2004, p. 36-37).

Verzuh (1999, p. 25) propõe uma estruturação dos projetos por meio dos processos "definição, planejamento, execução e encerramento" e menciona que estes se repetem ao longo dos vários estágios do ciclo de vida do projeto e do produto. Maximiano (2002, p. 49), compartilha a mesma idéia ao afirmar que os processos da administração de projetos – planejamento, organização, execução e controle – são necessários para o projeto como um todo e para cada fase de seu ciclo de vida.

Kerzner (2002) também trabalha esta estruturação por processos e acrescenta a importante discussão sobre o papel da cultura organizacional no processo de gerenciamento de projetos. Afirma que dificilmente duas empresas gerenciarão seus projetos da mesma forma e que, por

p.17).

<sup>&</sup>lt;sup>6</sup> Gestão de Projetos é o termo utilizado pelo autor ao se referir ao gerenciamento de projetos (KERZNER, 2002,

isso, a implementação do gerenciamento de projetos deve ter por base a cultura de cada organização.

A importância da criação de um modelo que sirva de guia ou base para o gerenciamento de projetos é explicada por Forsberg *et al* (1996, p. 14-16). Segundo os autores, os modelos ajudam a descrever como as coisas são feitas, servem para ilustrar o todo, fornecem uma base conceitual única, explicitam as regras de forma simples e auxiliam a identificação e comunicação dos relacionamentos e elementos-chave, eliminando de forma consciente fatores que geram confusão. Aplicado ao gerenciamento de projetos, um modelo bem definido garante:

- O entendimento, pela equipe do projeto, dos objetivos e da forma como o projeto será gerenciado;
- Uma perfeita comunicação entre as partes interessadas acerca dos diferentes aspectos do projeto, incluindo informações sobre a saúde e o progresso do projeto;
- A avaliação de caminhos alternativos e o aproveitamento de oportunidades que venham a surgir.

Para serem efetivos, os modelos ou os processos de gerenciamento de projetos devem possuir as seguintes características (FORSBERG *et al.*, p. 17):

- Definição explícita e operacional, utilizando estruturas, variáveis e relacionamentos;
- Serem óbvios e intuitivos a todos os interessados no projeto;
- Aplicabilidade ao ambiente de projeto, assegurando o tratamento das complexidades e do dinamismo inerentes aos processos do projeto e aos requisitos do projeto;
- Podem ser validados empiricamente no mundo real do gerenciamento de projetos.

Na visão do PMI (2004, p. 37-38) "o gerenciamento de projetos é realizado através de processos, usando conhecimento, habilidades, ferramentas e técnicas do gerenciamento de projetos que recebem entradas e geram saídas". Processo é definido como "um conjunto de ações e atividades inter-relacionadas realizadas para obter um conjunto pré-especificado de produtos, resultados ou serviços". Basicamente, existem duas categorias principais de processos em um projeto:

- a) *Processos do gerenciamento de projetos*: comuns à maioria dos projetos em grande parte do tempo, estes processos interagem entre si, às vezes de forma complexa, visando a um objetivo integrado, que é iniciar, planejar, executar, monitorar e controlar e encerrar um projeto. Estes processos descrevem, organizam e complementam as atividades do projeto;
- b) *Processos orientados ao produto:* estes processos especificam e criam o produto do projeto, variando por área de aplicação e sendo definidos no ciclo de vida do produto.

Verzuh (1999) e o PMI (2004, p. 37-38) afirmam que os processos de gerenciamento de projetos e os processos orientados ao produto se sobrepõem e interagem ao longo de todo o projeto. Entretanto, dado o foco deste trabalho, somente os processos do gerenciamento de projetos são aqui discutidos.

Os diferentes processos do gerenciamento de projetos, definidos com base em boas práticas<sup>7</sup>, com seus respectivos objetivos e integração, são apresentados no guia PMBoK (PMI, 2004, p.41). Esses processos encontram-se agregados em cinco grupos, a saber:

- Grupo de processos de iniciação: define e autoriza o projeto ou uma fase do projeto;
- *Grupo de processos de planejamento*: define e refina os objetivos e planeja as ações necessárias para atingir os objetivos e o escopo para os quais o projeto foi concebido;
- Grupo de processos de execução: integra pessoas e outros recursos visando à execução do plano de gerenciamento do projeto;
- Grupo de processos de monitoramento e controle: mede e monitora regularmente o
  progresso do projeto para identificar variações em relação ao plano, de forma a
  possibilitar a tomada de ações corretivas quando necessário, sempre com o intuito de
  atender aos objetivos do projeto;
- *Grupo de processos de encerramento*: formaliza a aceitação final do produto, serviço ou resultado e conduz o projeto, ou uma fase, a um final ordenado.

<sup>&</sup>lt;sup>7</sup> Boa prática significa que "[...] existe acordo geral de que a aplicação desses processos de gerenciamento de projetos tem demonstrado aumentar a chance de sucesso em uma ampla série de projetos" (PMI, 2004, p. 38).

Os grupos de processo propostos pelo PMI (*Ibid.*) estão ligados pelos objetivos que produzem e raramente representam eventos distintos e únicos. Em geral têm atividades sobrepostas, que demandam diferentes níveis de esforço durante todo o projeto e que interagem entre si dentro de uma mesma fase ou ao longo das várias fases do ciclo de vida. Quando um projeto é dividido em fases, os grupos de processos são normalmente repetidos dentro de cada fase, durante a vida do projeto, para conduzir o projeto ao seu término de modo eficaz. A estruturação dos grupos de processo ao longo das diferentes fases de um projeto e a forma como esses grupos de processos se sobrepõem e variam dentro de cada fase podem ser visualizadas nas Ilustrações 3 e 4.

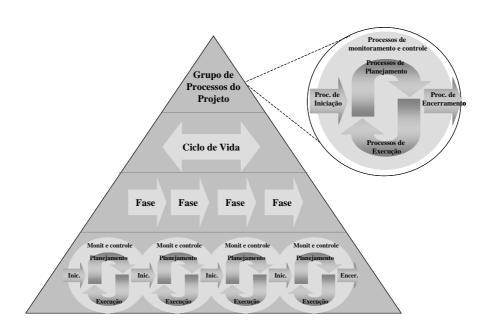


Ilustração 3 – Triângulo do grupo de processos do gerenciamento de projetos FONTE: PMI, 2004, p. 69

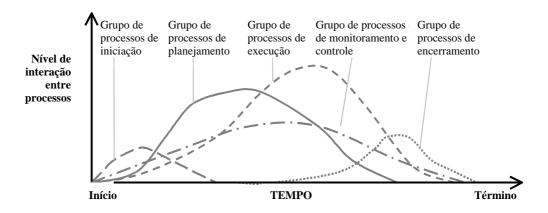


Ilustração 4 – Interação de grupos de processo em um projeto FONTE: PMI, 2004, p. 68

As interações entre os diferentes grupos de processos são, por sua vez, retratadas na Ilustração 5. Esta ilustração apresenta as principais informações e/ou documentos trocados entre os grupos de processos, que são utilizados como entradas e/ou saídas dos vários processos do Gerenciamento Clássico de Projetos.

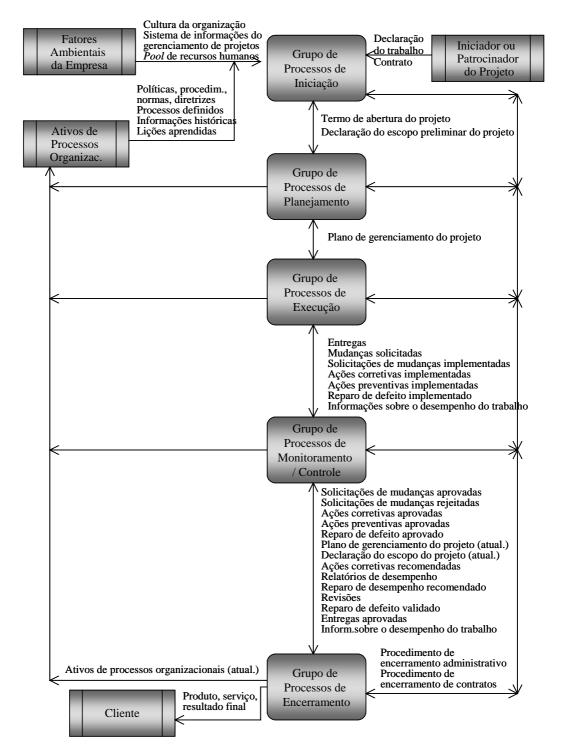


Ilustração 5 – Resumo de alto nível das interações entre os grupos de processo FONTE: PMI, 2004, p. 42

Compreendidos nos cinco grupos de processos mencionados, estão 44 processos de gerenciamento de projetos. Esses processos se distribuem ao longo de nove *áreas de conhecimento*, a saber (PMI, 2004, p 70):

- 1. Gerenciamento da integração do projeto;
- 2. Gerenciamento do escopo do projeto;
- 3. Gerenciamento de tempo do projeto;
- 4. Gerenciamento de custos do projeto;
- 5. Gerenciamento da qualidade do projeto;
- 6. Gerenciamento de recursos humanos do projeto;
- 7. Gerenciamento das comunicações do projeto;
- 8. Gerenciamento de riscos do projeto;
- 9. Gerenciamento das aquisições do projeto.

O Gerenciamento da integração do projeto inclui todos os processos e as atividades necessárias para identificar, definir, combinar, unificar e coordenar os diversos processos e atividades de gerenciamento de projetos nos grupos de processos de gerenciamento de projetos. O Gerenciamento de escopo do projeto compreende todos os processos necessários para garantir que o projeto inclua todo o trabalho necessário, e somente o necessário, para entregar o projeto com sucesso. O Gerenciamento de tempo do projeto inclui todos os processos necessários para realizar o projeto dentro do prazo estipulado. No Gerenciamento de custos do projeto são considerados todos os processos envolvidos no planejamento, estimativa, orçamento e controle de custos de modo a encerrar o projeto dentro do orçamento previsto. O Gerenciamento da qualidade do projeto compreende todas as atividades da organização executora que determinam as responsabilidades, os objetivos e as políticas de qualidade, de forma a atender às necessidades que motivaram sua realização. No Gerenciamento dos recursos humanos do projeto estão inseridos todos os processos que organizam e gerenciam a equipe do projeto. O Gerenciamento das comunicações do projeto reúne os processos necessários para garantir a geração, a coleta, a distribuição, o armazenamento, a recuperação e a destinação final das informações do projeto, de forma oportuna e adequada. O Gerenciamento dos riscos inclui os processos que tratam a identificação, a análise, a proposição de respostas, o monitoramento e o controle dos riscos de um projeto. E finalmente, o Gerenciamento das aquisições do projeto engloba os processos

para comprar ou adquirir os produtos, os serviços ou os resultados necessários, externamente à organização do projeto (PMI, 2004, p. 76, 103, 123, 157, 179, 221, 237, 269).

A Tabela 1 apresenta o mapeamento dos processos de gerenciamento de projetos e a sua associação aos grupos de processos e às áreas de conhecimento de acordo com o PMI (2004, p. 70).

Tabela 1 - Mapeamento dos processos de gerenciamento de projeto

	Grupos de processos de gerenciamento de projetos				
Processos da área de conhecimento	Grupo de processos de iniciação	Grupo de processos de planejamento	Grupo de processos de execução	Grupo de processos de monitoramento e controle	Grupo de processos de encerramento
Gerenciamento da integração do projeto	Desenvolvimento do termo de abertura do projeto Desenvolvimento da declaração de escopo preliminar do projeto	Desenvolvimento do plano de gerenciamento do projeto	Orientação e gerenciamento da execução do projeto	Monitoramento e controle do trabalho do projeto Controle integrado de mudanças	Encerramento do projeto;
Gerenciamento do escopo do projeto		Planejamento do escopo Definição do escopo Criação da EAP (Estrutura Analítica do Projeto)		Verificação do escopo Controle do escopo	
Gerenciamento de tempo do projeto		Definição da atividade Seqüenciamento de atividades Estimativa de recursos da atividade Estimativa de duração da atividade Desenvolvimento do cronograma		Controle do cronograma	
Gerenciamento de custos do projeto		Estimativa de custos Orçamento		Controle de custos	
Gerenciamento da qualidade do projeto		Planejamento da qualidade	Realização da garantia da qualidade	Realização do controle da qualidade	
Gerenciamento de recursos humanos do projeto		Planejamento de recursos humanos	Contratação ou mobilização da equipe do projeto Desenvolvimento da equipe do projeto	Gerenciamento da equipe do projeto	
Gerenciamento das comunicações do projeto		Planejamento das comunicações	Distribuição das informações	Relatório de desempenho	

	Grupos de processos de gerenciamento de projetos				
Processos da área de conhecimento	Grupo de processos de iniciação	Grupo de processos de planejamento	Grupo de processos de execução	Grupo de processos de monitoramento e controle	Grupo de processos de encerramento
				Gerenciamento das partes interessadas	
Gerenciamento de riscos do projeto		Planejamento do gerenciamento de riscos		Monitoramento e controle de riscos	
		Identificação de riscos			
		Análise qualitativa de riscos			
		Análise quantitativa de riscos			
		Planejamento de respostas a riscos			
Gerenciamento das aquisições do projeto		Planejamento das compras e aquisições	Solicitação das respostas de fornecedores	Administração de contrato	Encerramento de contrato
		Planejamento das contratações	Seleção de fornecedores		

FONTE: ADAPTADO DE PMI, 2004, p. 70.

Apesar de todo o detalhamento oferecido pelo Guia PMBoK, o PMI (2004, p.37) faz a ressalva de que o conhecimento, as habilidades e os processos de gerenciamento de projetos não devem ser aplicados uniformemente em todos os projetos. Ao gerente de projeto, em colaboração com a equipe de projeto, cabe a seleção e a determinação dos processos adequados e do grau de rigor de cada processo, a serem aplicados em um projeto específico.

Então, segundo o PMI (*Ibid.*) para que um projeto seja bem-sucedido, o gerente e a equipe de projeto devem:

- Selecionar os processos adequados dentro dos grupos de processos de gerenciamento de projetos necessários para atender aos objetivos do projeto;
- Usar uma abordagem definida para adaptar os planos e as especificações do produto de forma a atender aos requisitos do produto e do projeto;
- Atender aos requisitos para satisfazer as necessidades, os desejos e as expectativas das partes interessadas no projeto;
- Balancear as demandas conflitantes de escopo, tempo, custo, qualidade, recursos e risco para gerar um produto de qualidade.

Finalmente, ao se analisar o Guia PMBoK (PMI, 2004) percebe-se uma abordagem bastante estruturada para o gerenciamento de projetos, calcada basicamente em uma ampla e completa definição do escopo do projeto, em um planejamento prévio bem elaborado das várias áreas de conhecimento, no acompanhamento formal do progresso do projeto, em termos de escopo, prazo e custo e no controle bastante estrito das mudanças no projeto. Estas são consideradas as características principais do aqui denominado Gerenciamento Clássico de Projetos.

# 2.2.3 Projetos de Desenvolvimento de Software

Thomsett (2002, p.16-17) afirma que todos os projetos de engenharia (o autor inclui nesta categoria os projetos de desenvolvimento de software) têm de lidar basicamente com duas vertentes – uma técnica e outra gerencial – representando a dualidade "gerenciamento técnico" x "gerenciamento de projeto". Segundo o autor (*Ibid.*), até os anos 50, as atribuições gerenciais nesses projetos (a elaboração do cronograma, do orçamento, o controle de mudanças, o gerenciamento de contratos, citados a título de exemplo) ficavam a cargo do arquiteto do projeto. No entanto, a concentração dos aspectos técnicos e da visão do projeto como um todo nas mãos de uma única pessoa acabava por gerar, em alguns casos, situações de conflito de interesse. Thomsett (*Ibid.*) comenta que demorou algum tempo até o gerenciamento de projetos ser reconhecido como uma disciplina em si e os papéis e responsabilidades de âmbito técnico e gerencial passassem a ser tratados de forma separada.

Ao longo de várias décadas, o desenvolvimento de software passou por um processo de melhoria, cuja ênfase inicial recaiu sobre aspectos técnicos, ou seja, sobre os métodos de desenvolvimento propriamente ditos. Este aprimoramento, como muitos processos evolutivos, teve por base a análise dos problemas encontrados em processos e modelos anteriores e a definição de novas formas de atuação.

De acordo com Beck (1999), o *Modelo em Cascata*<sup>8</sup> foi o primeiro a buscar a construção de um software voltado para as necessidades dos usuários. Tem por base a elaboração de uma

\_

<sup>&</sup>lt;sup>8</sup> Waterfall Model (Royce, 1970).

documentação completa e exaustiva dos requisitos e das funcionalidades do software. A documentação é obtida através de interações com usuários e clientes, as quais podiam perdurar por meses, em uma etapa denominada *Análise*. A partir desta documentação abrangente, engenheiros de software, especialistas em base de dados e outros profissionais definem a arquitetura do software, na etapa denominada *Desenho*. Nas etapas subseqüentes, os programadores desenvolvem o código e promovem o teste e a entrega do software completo e perfeito – etapas de *Implementação* e *Teste*.

Beck (1999) afirma que na teoria, o modelo parece perfeito, mas na prática, vários são os problemas. O primeiro deles, é que, quase sempre, os desejos dos usuários mudam. Após meses ou mesmo anos de levantamento, especificação e construção de diagramas e fluxos, alguns usuários ainda não estão certos do que desejam, sabendo apenas que não querem o que está em produção. Um segundo problema diz respeito às mudanças de requisitos no meio do processo de desenvolvimento. Bohem (2002) destaca que, em qualquer método clássico de desenvolvimento, as primeiras dificuldades surgem quando das solicitações de alteração de código, dado que para atendê-las há a necessidade de reunir programadores, arquitetos de software, analistas de base de dados, para análise e alteração da documentação e posterior implementação da mudança. O *Modelo em Cascata* é uma tentativa para resolver esta situação, defendendo o congelamento dos requisitos do software, ou seja, não permitindo que qualquer mudança seja feita no decorrer do projeto. Isto, entretanto, se mostrou inviável na prática (BECK, *op. cit.*). Do ponto de vista do gerenciamento de projetos, o *Modelo em Cascata* pressupõe o congelamento do escopo do produto e do projeto (THOMSETT, 2002, p. 136).

Numa tentativa de aprimorar este modelo, técnicas iterativas e incrementais surgiram, propondo quebras no ciclo de desenvolvimento do software, com repetições do *Modelo em Cascata* ao longo do processo. Este novo enfoque tem por objetivo reduzir o prazo de desenvolvimento. Como no *Modelo em Cascata*, todos os requisitos do software são analisados antes do início da codificação, entretanto, são divididos em incrementos da funcionalidade básica. Os desenvolvimentos podem ser feitos paralelamente, visando a uma economia de tempo (BECK, 1999).

Enquanto os modelos incrementais têm por foco a redução de prazo, outros métodos, como o desenvolvimento iterativo e o *Modelo em Espiral*<sup>9</sup>, buscam lidar melhor com as mudanças de requisitos e trabalhar o gerenciamento de riscos. Segundo Cohen *et al* (2003, p. 3), estes modelos trabalham os fatores de risco de uma forma estruturada e planejada ao invés de tentar mitigá-los apenas quando surgissem.

Os autores afirmam que o desenvolvimento iterativo propõe uma quebra do projeto de desenvolvimento de software em iterações de prazo variável, sendo que em cada uma delas, um conjunto completo de funcionalidades, codificado com base em uma documentação elaborada previamente, é entregue (COHEN et al, 2003, p. 3.). Na primeira iteração são geradas as funcionalidades básicas, que sofrem incrementos ao longo das demais iterações. Apesar de cada iteração passar por todas as etapas do *Modelo em Cascata*: "Análise, Desenho, Implementação e Teste", o desenvolvimento iterativo lida melhor com as mudanças, uma vez que se tem apenas um conjunto completo e fixo de requisitos para a iteração em andamento. Para as demais iterações é possível fazer mudanças até a fase de "Análise". Este modelo permite, até certo ponto, a absorção de alterações de tecnologia ou de mudanças solicitadas pelo cliente (COHEN et al, Ibid.).

De forma similar, o *Modelo em Espiral* evita a necessidade de detalhamento e de definição dos requisitos finais do software logo no início do projeto. Contudo, difere do desenvolvimento iterativo por priorizar os desenvolvimentos, levando em consideração o fator risco e não as funcionalidades propriamente ditas. Considerando o gerenciamento de projetos, os modelos iterativos e em *Espiral* implicam em um projeto com o escopo inicial aberto, que é detalhado ao longo das várias iterações (THOMSETT, 2002, p. 139-142).

O surgimento destes novos modelos de desenvolvimento de software promoveu um avanço na área técnica, mas pouco ainda havia sido feito no campo gerencial. Cohen *et al (op. cit.)* afirmam que a grande mudança na forma como as organizações conduziam o gerenciamento de seus projetos de desenvolvimento de software ocorreu com o advento do SW-CMM –

\_

<sup>&</sup>lt;sup>9</sup> Espiral Model (BOEHM, 1988).

Software Capability Maturity Model. Criado em 1986, pelo Software Engineering Institute (SEI) na Carnegie Mellon University, como um modelo para promover o amadurecimento das organizações no processo de desenvolvimento de software<sup>10</sup>, o SW–CMM foi amplamente difundido nos últimos anos em organizações de diferentes países do mundo (SEI, 1995; Paulk, 2001, p. 19).

Composto por cinco níveis de maturidade<sup>11</sup> (inicial, reproduzível<sup>12</sup>, definido, gerenciado, otimizado), o SW-CMM está estruturado em 52 objetivos distribuídos entre 18 áreas-chave, que descrevem as melhores práticas de engenharia (desenvolvimento de software) e de gerenciamento, além de apontar o caminho para o aprimoramento dos processos de construção de software nas organizações (PAULK, 2001, p. 19.). Esta estrutura pode ser visualizada na Tabela 2 abaixo. Cohen *et al* (*op. cit.*) afirmam que o objetivo maior do SW-CMM é fazer com que uma organização atinja um nível ótimo de consistência, previsibilidade e confiabilidade nos processos de desenvolvimento de software (nível cinco na escala de maturidade).

Tabela 2 - Visão geral do SW-CMM

Nível	Foco Foco	Áreas-chave de Processos
5. Otimizado	Melhoria contínua dos processos	- Prevenção de defeitos
		- Gerenciamento de mudanças na tecnologia
		- Gerenciamento de mudanças nos processos
4. Gerenciado	Qualidade dos produtos e dos	- Gerenciamento quantitativo do processo
	processos	- Gerenciamento da qualidade do software
3. Definido	Processos de engenharia	- Foco no processo organizacional
	(desenvolvimento) e apoio à	- Definição do processo organizacional
	organização	- Programa de treinamento
		- Gerenciamento integrado de software
		- Engenharia de produto de software
		- Coordenação entre grupos
		- Revisões entre pares
2. Reproduzível	Processos de gerenciamento de	- Gerenciamento dos requisitos
	projetos	- Planejamento do projeto de desenvolvimento
		de software

<sup>&</sup>lt;sup>10</sup> Ressalta-se que o SW-CMM não é um modelo de desenvolvimento de software, mas sim um modelo que busca promover o amadurecimento dos processos de desenvolvimento de software.

<sup>&</sup>lt;sup>11</sup> A escala de maturidade do SW-CMM é utilizada para classificar as organizações em função do grau de consistência, previsibilidade e confiabilidade de seus processos de desenvolvimento de software (PAULK, 2001, p.19).

<sup>12</sup> Repeatable.

Nível	Foco	Áreas-chave de Processos
		- Acompanhamento e rastreamento do projeto de
		desenvolvimento de software
		- Gerenciamento das subcontratações
		- Garantia de qualidade do software
		- Gerenciamento da configuração do software
1. Inicial	Competência das pessoas	N/A

FONTE: PAULK, 2001, p. 20.

Apesar da ampla disseminação entre a comunidade de desenvolvimento de software, Paulk (2001, p. 25) destaca que a maioria das organizações ainda se encontra no primeiro nível de maturidade – *Inicial*, enquanto pouquíssimas pertencem ao nível cinco – *Otimizado*. O autor ainda comenta que o formalismo inerente à maioria dos processos de melhoria propostos pelo SW-CMM faz com que o modelo tenha por foco projetos de grande porte, caracterizados por severos requisitos de confiabilidade. Mas, ressalta também que, apesar disto, os princípios do SW-CMM podem ser adaptados para atender empresas e projetos menores.

Os modelos de desenvolvimento de software até aqui apresentados (*Modelos em Cascata*, iterativos e em *Espiral*), são exemplos do chamado enfoque clássico de desenvolvimento de software, que tem por características principais o formalismo e a preparação de uma extensa documentação. Sob o aspecto do gerenciamento de projetos, o aporte gerencial e de processos propiciado pelo SW-CMM (SEI, 1995) traz um alinhamento aos conceitos do gerenciamento clássico de projetos, com grande ênfase no planejamento e controle (PMI, 2004).

Por fim, ressalta-se que apesar dos modelos em *Espiral* e iterativos permitirem uma maior agilidade ao desenvolvimento de software, estes ainda são criticados por não conseguirem promover respostas às mudanças em uma velocidade adequada à realidade dos negócios. O formalismo e a manutenção de uma documentação exaustiva típicos destes modelos, somados ao foco no planejamento e no controle do gerenciamento de projetos são considerados empecilhos à verdadeira agilidade no desenvolvimento de software (COHEN *et al*, 2003, p.3).

#### 2.3 Sucesso e Fracasso no Gerenciamento de Projetos

Retomando a colocação de Schilling e Hill (1998) e Kerzner (2002) de que do sucesso dos projetos pode depender o sucesso das organizações e considerando o modelo conceitual desta

pesquisa, faz-se necessário discutir a questão relativa ao sucesso e fracasso no gerenciamento de projetos e, em seguida, avaliar o desempenho dos projetos de desenvolvimento de software.

Assim como vários conceitos ligados a projetos, a definição de sucesso de um projeto sofreu alterações ao longo do tempo. Sendo assim, em um primeiro momento é recuperada a visão tradicional de sucesso e analisado o desempenho dos projetos de desenvolvimento de software e, em seguida, é apresentada a visão "moderna" de sucesso, que serve de base para a constituição dos chamados Métodos Ágeis de Desenvolvimento de Software e do Gerenciamento Ágil de Projetos.

## 2.3.1 Visão Tradicional do Sucesso de um Projeto

Para empreender uma jornada, é indispensável ter um destino e um plano para chegar até ele. O destino precisa ser claramente identificado, caso contrário nunca se saberá quando a viagem acaba. De certa forma, as mesmas questões são válidas para as práticas de gerenciamento de projetos. (KERZNER, 2002, 43).

Como reconhecer um projeto bem-sucedido? Deveria o sucesso de um projeto ser medido em termos da satisfação do cliente, de lucratividade, de geração de novos negócios ou aumento da fatia do mercado? Ou deveria ser definido como o atendimento das especificações / expectativas do cliente, ou sua superação? Ou ainda, pelo "simples" cumprimento do planejamento estabelecido?

Perguntas como estas, nem sempre são respondidas com propriedade nas organizações. De fato, poucas são as empresas que se preocupam em definir para seus gerentes, o que entendem por sucesso e, quando o fazem, oferecem uma definição muito pobre deste objetivo. Expectativas mal trabalhadas podem representar um grande problema no processo de gerenciamento de projetos (KERZNER, 2002, p. 43).

Segundo o Kerzner (2002, p.44), nos primórdios da gestão de projetos, o sucesso era medido apenas em termos técnicos, ou seja, o produto gerado era, ou não, adequado. Não havia qualquer preocupação quanto ao custo ou ao prazo de execução, nem qualquer definição de

sucesso em termos empresariais. À medida que as empresas passaram a aprimorar o gerenciamento de projetos, esta postura, inicialmente defendida pelos gerentes seniores das organizações, mudou. O sucesso de um projeto passou a ser compreendido como a conclusão da programação no prazo, no custo e no nível de qualidade pré-estabelecido (KERZNER, 2002, p.44). Entretanto, o autor ressalta que esta era ainda uma medida incompleta, uma vez que estes indicadores retratavam uma definição interna de sucesso.

Dando continuidade à evolução do conceito, Kerzner (2002, p. 44-45) afirma que a "melhor explicação de sucesso é aquela que o mensura em termos de fatores primários e secundários". São considerados *fatores primários*, o cumprimento do prazo, do custo e do nível de qualidade pré-estabelecidos, definição de qualidade esta, feita pelo cliente. Por *fatores secundários* entende-se a aceitação do projeto pelo cliente e se este concorda na divulgação de seu nome como referência. Segundo o autor, a definição absoluta de sucesso só pode ser visualizada quando o cliente está tão satisfeito com os resultados, que permite a utilização de seu nome como referência (KERZNER, *Ibid.*).

Kerzner (*Ibid.*) salienta ainda, que os fatores primários podem ser alterados de acordo com a organização, aumentando ou reduzindo sua abrangência, englobando, por exemplo, o atendimento às normas de segurança, às regulamentações governamentais, à legislação ambiental, entre outros. A orientação de uma empresa a projetos também pode alterar a definição de sucesso.

Ao abordar o tema, Verzuh (1999, p. 17) aponta três componentes como a base para o sucesso de um projeto: o cumprimento do prazo previsto, o cumprimento do custo esperado e a alta qualidade do produto gerado, incluindo requisitos de funcionalidade e de desempenho. O autor menciona que estas são as três variáveis principais de um projeto, sendo extremamente interdependentes. Qualquer alteração em uma delas tem efeito em uma ou em todas as demais. Verzuh (*Ibid.*) afirma que o grande desafio do gerente de projeto é buscar o equilíbrio ótimo entre "prazo-custo-qualidade", idéia esta compartilhada por Meredith e Mantel (2000, p. 4). Estes autores a representam no que chamam de *três objetivos do projeto*, conforme mostra a Ilustração 6.

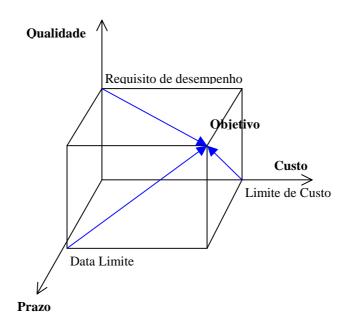


Ilustração 6 – Os três objetivos do projeto: custo-prazo-qualidade FONTE: MEREDITH; MANTEL, 2000, p. 4

Verzuh (1999, p. 18) complementa, porém, que não basta atingir os objetivos de custo, prazo e qualidade para que o projeto seja considerado um sucesso. É importante que a mesma visão do equilíbrio "prazo-custo-qualidade" seja compartilhada entre o cliente e o gerente de projetos, assegurando o alinhamento das expectativas. Reconhecer que o sucesso do projeto está também estritamente relacionado à percepção de outras pessoas envolvidas, é um grande incentivo a buscar um entendimento comum em torno dos seus objetivos.

Anteriormente, Pinto e Kharbanda (1995a), também defenderam a idéia de que o sucesso de um projeto não pode ser mensurado somente em termos da tradicional restrição tripla – prazo, custo e qualidade (desempenho). Considerando o contexto de negócio e a crescente ênfase na satisfação do cliente, os autores afirmam que há a necessidade de se expandir o conceito de sucesso para incluir uma quarta variável: a satisfação e a entrega de valor agregado ao cliente. Pinto e Kharbanda (1995a) ressaltam que todo esforço realizado num projeto deve ter por foco ouvir e entender as preocupações e as opiniões do cliente, de forma a gerar um produto final adequado às suas necessidades, assegurando assim uma transição tranqüila e o aceite do projeto por parte dele.

Mais recentemente, Dinsmore e Neto (2004, p. 15) salientam que "[...] executar projetos dentro do prazo e orçamento previstos, atender à qualidade especificada e satisfazer às

expectativas da organização responsável pelo projeto são indicadores de sucesso na gerência de programas e projetos, independentemente da natureza dos mesmos".

Cabe salientar, porém, que Meredith e Mantel (2000, p. 4) refutam a idéia da criação de uma quarta variável de sucesso (além dos objetivos de prazo, custo e qualidade), que represente as expectativas do cliente, ao afirmar que estes requisitos são "parte inerente às especificações do projeto". Os autores são bastante enfáticos ao mencionar que, considerar os requisitos ou expectativas dos clientes em separado das especificações do projeto significa inserir e estimular o conflito entre o cliente e a equipe de projeto, o que é antagônico ao que se espera do bom gerenciamento de projetos.

Partindo do pressuposto que um projeto é considerado bem-sucedido caso atenda a quatro critérios – tempo, custo, eficácia e satisfação do cliente, Pinto e Slevin (1988), após pesquisa realizada com gerentes de projeto, estabeleceram um modelo de dez fatores considerados críticos para o sucesso de um projeto. Jiang *et al* (1996), ao conduzir um estudo com profissionais da área de sistemas de informação, chegou a conclusões similares. Estes fatores são apresentados, em ordem de importância, na Tabela 3.

Tabela 3 - Fatores críticos de sucesso em ordem de importância

	Tabela 5 - 1 atores criacos de sucesso em ordem de importanca			
Fator crítico		Descrição		
1.	Missão do projeto	Definição clara de objetivos no início do projeto		
2.	Apoio da alta administração	Comprometimento da alta administração em prover os recursos e poder /		
		autoridade necessários para o sucesso do projeto		
3.	Cronograma detalhado do	Especificação detalhada de todos os passos necessários para a		
	projeto	implementação do projeto		
4.	Cliente consultor	Comunicação e habilidade em ouvir todas as partes envolvidas no projeto		
5.	Pessoal	Recrutamento, seleção e treinamento dos recursos necessários para formar a		
		equipe de projeto		
6.	Ações técnicas	Disponibilidade da tecnologia, habilidades e conhecimentos necessários para		
		a realização das atividades técnicas requeridas		
7.	Aceite do cliente	Ação de "vender" os resultados finais do projeto aos usuários finais		
8.	Monitoramento e feedback	Capacidade de fornecer feedback em todos os estágios do projeto		
9.	Comunicação	Criação de uma rede de contatos e fornecimento de informações a todos os		
		atores-chave no âmbito do projeto		
10.	Resolução de problemas	Habilidade em lidar com situações inesperadas e desvios do plano		

FONTE: ADAPTADO DE PINTO: SLEVIN. 1988.

Apesar dos autores (PINTO; SLEVIN, 1988; JIANG *et al*, 1996) encontrarem resultados similares em seus trabalhos, estes não devem ser generalizados. De acordo com Meredith e Mantel (2000, p. 548), pesquisas realizadas apontam que os fatores críticos de sucesso dos projetos variam em função do tipo de indústria.

Do exposto até o momento, pode-se dizer que sob a ótica tradicional de definição do sucesso de um projeto, o estabelecimento dos objetivos de custo, prazo e qualidade do projeto, refletidos num planejamento realista e adequado, assim como o correto entendimento dos requisitos e das expectativas dos clientes e demais interessados no projeto, constituem os fatores básicos para o sucesso de um projeto.

# 2.3.2 Desempenho dos Projetos de Desenvolvimento de Software

A avaliação do desempenho de projetos de desenvolvimento de software tem sido uma preocupação de vários autores e organizações nos últimos anos. Várias pesquisas foram conduzidas visando à identificação do índice de desempenho dos projetos e dos fatores críticos de sucesso destes projetos (Jiang *et al*, 1996; GIGA INTERNATIONAL GROUP, 2002; STANDISH GROUP INTERNATIONAL, 1999, 2001 e 2003). De forma geral, os resultados de tais pesquisas apontam para um cenário pouco animador, retratando elevados índices de insucesso dos projetos de desenvolvimento de software.

Dado que, usualmente, os relatórios da série *The Chaos Report*, publicados pelo STANDISH GROUP INTERNATIONAL (1999; 2001; 2003) são os mais citados nas obras de estudiosos e praticantes do desenvolvimento de software (THOMSETT, 2002; UDO; KOPPENSTEINER, 2003; HIGHSMITH, 2004; JOHNSON, 2005; AMBLER, 2005), estes servirão de base para a retratação do desempenho dos projetos de desenvolvimento de software neste trabalho.

Nas várias edições da pesquisa do STANDISH GROUP INTERNATIONAL (*op. cit.*), o desempenho dos projetos foi classificado em três modalidades:

- a) Sucesso compreendendo os projetos entregues no prazo, dentro do orçamento previsto e com padrão de qualidade aceito pelo cliente;
- b) *Insucesso* englobando projetos cancelados antes de sua conclusão ou nunca implementados;

c) Sucesso parcial – reunindo projetos encerrados e operacionais, mas cujos indicadores de prazo e custo tiveram uma variação negativa frente ao previsto, ou projetos que foram entregues com qualidade inferior à especificada.

De acordo com o relatório divulgado em 2003 (STANDISH GROUP INTERNATIONAL, 2003), consolidando a análise de cerca de 50.000 projetos de TI de diferentes empresas distribuídas ao redor do mundo realizados na última década, os índices de projetos encerrados com resultados insatisfatórios — insucesso ou sucesso parcial — continuam relativamente elevados (66% em 2002). Isto, apesar de os resultados gerais dos projetos de desenvolvimento de software apresentarem uma melhora significativa no período, com o percentual de projetos bem-sucedidos variando de 16% em 1994, para 34% em 2002.

Se por um lado, os estouros de orçamento foram reduzidos de 180% em 1994 para 43% em 2002, os indicadores de desempenho de prazo e qualidade pioraram após um período de melhores resultados. Os acréscimos não previstos de prazos passaram a 82% em média, em 2002, contra um patamar de 63% reportado em 2000; apenas 52% dos projetos cumpriram as especificações de qualidade, frente a 67% em 2000. Estas variações demonstram uma certa instabilidade nos resultados dos projetos, com ganhos ainda não consistentes ao longo do tempo.

No Gráfico 1 é possível visualizar de forma clara a evolução histórica do desempenho dos projetos de desenvolvimento de software entre 1994 e 2002. Este gráfico tem por base os dados de classificação e os resultados das pesquisas realizadas pelo STANDISH GROUP INTERNATIONAL (1999; 2001; 2003). Percebe-se claramente um movimento entre os extremos, com uma redução do índice de projetos mal-sucedidos e um incremento do percentual de projetos bem-sucedidos. Entretanto, à exceção de 1996, quase não há alteração no índice de projetos classificados como de "sucesso parcial", que gira em torno de 50%.

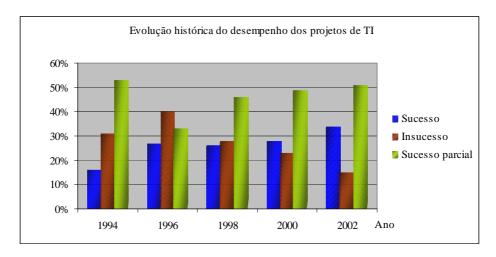


Gráfico 1 – Evolução histórica do desempenho dos projetos de TI FONTE: STANDISH GROUP INTERNATIONAL, 2003.

Outro aspecto importante das pesquisas realizadas pelo STANDISH GROUP INTERNATIONAL (2001; 2003) foi a determinação de fatores críticos de sucesso em projetos de desenvolvimento de software. Esses fatores, pontuados de acordo com sua influência no sucesso de um projeto, são apresentados na Tabela 4.

Tabela 4 - Fatores críticos de sucesso em projetos de desenvolvimento de software

	Critérios de Sucesso	Pontuação
1.	Apoio executivo	18
2.	Envolvimento dos usuários	16
3.	Experiência do gerente de projeto	14
4.	Entendimento claro dos objetivos do negócio	12
5.	Escopo reduzido e controlado	10
6.	Infraestrutura padrão de sistemas	8
7.	Requisitos básicos bem estabelecidos	6
8.	Metodologia formal de gerenciamento de projetos	6
9.	Estimativas confiáveis e realistas	5
10.	Outros	5

FONTE: ADAPTADO DE STANDISH GROUP INTERNATIONAL, 2001.

De acordo com os relatórios publicados, entre 1994 e 2002, muitas das organizações que participaram da pesquisa investiram no aprimoramento de seus processos e metodologias de gerenciamento de projetos. Relata-se que a melhora apresentada nos resultados dos projetos pode ser atribuída, em parte, à adoção de boas práticas relacionadas ao gerenciamento de projetos (STANDISH GROUP INTERNATIONAL, 2001; 2003). Nesta linha, entre os principais fatores responsáveis por esta transformação, podem ser citados: a realização de projetos menores, a utilização de melhores ferramentas para o monitoramento e controle dos

projetos e também, a designação de gerentes de projeto mais qualificados, com experiência e domínio dos processos de gerenciamento de projeto.

Apesar dos ganhos obtidos, o STANDISH GROUP INTERNATIONAL (2003) sugere que muitos dos projetos fracassaram ou estavam fadados ao insucesso, não pela ausência de verba ou pela falta de domínio tecnológico mas sim, pelo desconhecimento ou emprego incorreto de práticas de gerenciamento de projetos. Para Thomsett (2002), Chin (2004) e Highsmith (2004) o enfoque clássico do gerenciamento de projetos não se mostrou plenamente efetivo para o desenvolvimento de software. Estes autores explicam esta diferença pelo fato de que, usualmente, os projetos de desenvolvimento de software estão inseridos em ambientes de negócio bastante dinâmicos, sujeitos a mudanças constantes, o que foge aos padrões do gerenciamento clássico de projetos. Fowler (2003) ainda complementa que há uma grande dificuldade de mensuração do desempenho dos projetos de desenvolvimento de software.

A análise destas informações parece indicar alguma inconsistência ou a existência de um hiato entre as práticas do gerenciamento de projetos clássico adotadas pelas organizações e suas necessidades, no que tange ao desenvolvimento de software. Este desconforto foi sentido por vários estudiosos e praticantes do desenvolvimento de software que acabaram por buscar soluções alternativas para o problema. Ratificando esta percepção, pesquisa realizada por Reifer (2002, p. 14-16) afirma que, surpreendentemente, empresas com certa maturidade nos processos de desenvolvimento de software (classificadas como SW-CMM nível 2 ou superior) iniciaram a busca por algo novo, por enfrentarem problemas recorrentes no atendimento aos objetivos dos projetos.

Em suma, tal como ilustra o próprio título da pesquisa mencionada, denominada *The Latest Chaos Report on Project Management* (STANDISH GROUP INTERNATIONAL, 2003), os projetos de desenvolvimento de software permanecem, invariavelmente, associados ao insucesso, sendo que parcela deste mau desempenho é justificada pelo emprego incorreto ou pela inadequação das práticas de gerenciamento de projetos, o que sugere a necessidade de se repensar a disciplina de gerenciamento de projetos aplicada ao desenvolvimento de software.

Neste momento de reflexão da prática, alguns autores, como Thomsett (2002) e Cohen e Graham (2002) propõem que sejam alterados também os critérios de mensuração do sucesso dos projetos, argumentando que a forma de avaliação tradicional pode ser uma das

responsáveis pelo fracasso de muitos projetos. Esta nova abordagem para a avaliação do sucesso dos projetos é apresentada a seguir.

#### 2.3.3 Visão Moderna de Sucesso de um Projeto

Partidários da visão de que o gerenciamento de projetos não é apenas um processo técnico, Cohen e Graham (2002, p. vii) afirmam que gerenciar projetos "[...] transformou-se em um processo de negócio de importância crítica" e que os gerentes devem sempre buscar melhorar a contribuição de seus projetos nos resultados da organização. Ressaltam que todas as decisões do dia-a-dia dos gerentes de projetos devem ser orientadas pela necessidade de se atender às expectativas dos acionistas e que os projetos também devem ser gerenciados tendose em mente a lucratividade futura da empresa.

Para tratar a questão de sucesso no gerenciamento de projetos, Cohen e Graham (2002, p. vii-ix) introduzem um conceito inovador, denominado *ciclo de vida dos resultados do projeto* ou *project outcome life cycle* (POL). Nesta abordagem, também defendida por Thomsett (2002, 27-29), a responsabilidade dos gerentes de projeto não mais se encerra ao final da fase de desenvolvimento do projeto, mas se prolonga por todo o período de pós-implementação, quando da geração dos resultados, conforme mostra a Ilustração 7.

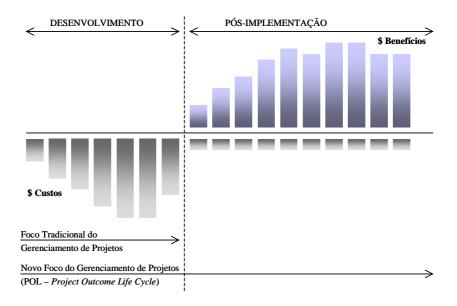


Ilustração 7 – Perspectiva do ciclo de vida dos resultados do projeto FONTE: THOMSETT, 2002, p. 28

Neste modelo, é fundamental que os gerentes e a equipe de um projeto passem a entender e a adotar a perspectiva da alta administração – a necessidade de geração de valor econômico – no planejamento e execução de seus projetos (COHEN; GRAHAM, 2002, p. vii-ix.).

Em sua obra, Cohen e Graham (2002) mostram que a visão "antiga" de sucesso, mensurada através da obtenção dos resultados esperados e do cumprimento das metas de prazos e custos, perdeu a validade. Em seu lugar, propõem o aumento do valor econômico como principal critério de avaliação do sucesso da gestão de projetos.

Numa visão ainda mais extrema, Thomsett (2002, p. 69) chega a mencionar que "[...] o conceito da restrição tripla<sup>13</sup> levou mais projetos ao insucesso, do que qualquer outro mito do gerenciamento de projetos". Segundo o autor, inúmeros são os projetos cerceados durante seu desenvolvimento, para atendimento de metas de prazo e custos, que acabam por obter resultados insatisfatórios ao seu final. Não se restringindo ao conceito de valor econômico, apesar de defender a perspectiva do ciclo de vida dos resultados do projeto, Thomsett (2002, p. 69-77), propõe que o sucesso seja definido em termos do atendimento às expectativas dos clientes, critério este totalmente vinculado ao entendimento do ambiente do cliente, ou seja, ao entendimento de sua cultura, das pressões a que seu negócio está sujeito, de suas preocupações e de seus objetivos. Explorando um pouco mais o conceito de expectativas, Thomsett (*Ibid.*) afirma que estas podem ser definidas em termos de sete indicadores principais, abaixo apresentados:

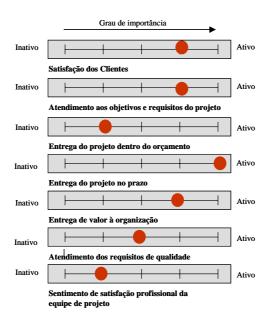
- Grau de satisfação dos clientes: Como os clientes se sentem em relação ao projeto?
- Atendimento aos objetivos e requisitos do projeto: O que os clientes desejam com o projeto?
- Entrega do projeto dentro do orçamento: Quanto os clientes estão dispostos a pagar pelo projeto?
- Entrega do projeto no prazo: Para quando os clientes desejam o projeto?
- Entrega de valor à organização: Por que os clientes desejam o projeto?
- Atendimento aos requisitos de qualidade: Quão bem o produto deve ser construído?

<sup>&</sup>lt;sup>13</sup> Restrição tripla: "prazo-custo-qualidade" (MEREDITH; MANTEL, 2000, p. 4)

- Satisfação da equipe de projeto: Quão bem a equipe tem de se sentir a respeito do projeto?

O grau de importância de cada um dos indicadores apresentados acima varia de projeto para projeto. Sendo assim, Thomsett (*Ibid.*) propõe a utilização de uma escala que sinalize se determinado indicador é válido ou não para o projeto em questão e que informe a sua importância (ver Ilustração 8). A posição de cada indicador deve ser determinada em conjunto pelos clientes, patrocinadores e gerente do projeto e comunicada à equipe do projeto. O planejamento e a execução do projeto devem ser norteados, então, por estas diretrizes, aumentando as chances de sucesso.

A adoção destas novas formas de mensuração de sucesso dos projetos cria um novo paradigma do gerenciamento de projetos, convertendo o conceito da restrição tripla (MEREDITH; MANTEL, 2000, p. 4) em possibilidades, conforme mostram as Tabelas 5 e 6, e, principalmente, promovendo uma mudança radical no sistema de controle (COHEN; GRAHAM, 2002, p. 6-23; THOMSETT, 2002, p. 69-77).



**Ilustração 8 – Indicadores de sucesso de um projeto** FONTE: THOMSETT, 2002, p. 75

Tabela 5 - Novo paradigma da gestão de projetos

Fator	De	Para
Estratégia da	Não é minha preocupação	Determinante número um do sucesso do
Organização		projeto
Processos de	Os processos de programação são	Os fatores comportamentais condicionam
Gestão de	importantes. Os fatores comportamentais	o sucesso do projeto e o valor econômico.
Projetos	são "perfumarias"	Como tal, são essenciais
Marketing	Algo de responsabilidade do pessoal do	Algo que deve ser compreendido durante
	Departamento de Marketing	o projeto, como condição de sucesso
		(Gestão de projetos movida a Marketing)
Custos	Apenas os custos do projeto são	Os custos são importantes durante o
	importantes	projeto e durante todo o ciclo de vida de
		seus resultados
Finanças	Algo sem consequências para o projeto	Algo importante na determinação da
		contribuição do projeto para o valor
		econômico

FONTE: COHEN; GRAHAM, 2002, p.14.

Tabela 6 - Das velhas restrições aos novos processos

Velhas Restrições	Novos Processos	Produtos
Resultado	Aprimoramento do desempenho de mercado	Desempenho de mercado
Duração (prazos fixos, cronogramas)	Escolha dos momentos certos	Maximização da geração de caixa
Custo (orçamento)	Investimento	Vantagem competitiva

FONTE: COHEN; GRAHAM, 2002, p.16.

As consequências desta nova orientação se traduzem numa alteração da forma de mensuração do sucesso dos projetos (COHEN; GRAHAM, 2002, p. 23):

- Do atendimento às especificações rígidas à satisfação dos clientes e execução da intenção estratégica;
- Da observação de um orçamento rígido à gestão do fluxo de caixa de modo a aumentar o valor para o acionista;
- Do cumprimento de prazos fixos à escolha do melhor momento para entrar no mercado e reduzir o prazo até o ponto de equilíbrio;
- Do foco interno no projeto ao foco externo no cliente, no mercado, na concorrência e em todo o ciclo de vida do projeto;
- Da simples execução do projeto à ajuda na implementação da estratégia organizacional.

Os gerentes de projeto têm de compreender o que significam estes novos critérios e descobrir como gerenciar seus projetos para atingir resultados favoráveis, com foco na lucratividade do negócio. Lembrando ainda que esses resultados não são estáticos e que os projetos são

influenciados pelas turbulências do ambiente organizacional e, sobretudo, pelas forças dos mercados globais (COHEN; GRAHAM, *Ibid.*; THOMSETT, 2002).

O dinamismo do ambiente de projetos pode ser ilustrado pela máxima de Von Moltke de que "nenhum plano sobrevive ao contato com o inimigo" (MITCHELL, 1979, p. 22 *apud* COHEN; GRAHAM, 2002, p. 21), transferida para a gestão de projetos: "poucas especificações sobrevivem ao contato com o mercado" (COHEN; GRAHAM, *Ibid.*).

Neste novo contexto, os gerentes de projetos devem estar aptos a tomar decisões instantâneas à medida que o ambiente sofra mudanças e a pensar no projeto como se fosse um empreendimento completo (COHEN; GRAHAM, 2002; THOMSETT, 2002). Ou seja, os autores defendem a transformação da visão do gerenciamento clássico de projetos num modelo mais ágil e que atribua maior autonomia ao gerente e à equipe de projetos.

# 2.4 Métodos Ágeis de Desenvolvimento de Software

Como uma resposta às crescentes pressões por inovação em prazos cada vez mais reduzidos, às necessidades de constantes mudanças de requisitos e ao mau desempenho de grande parte dos projetos de desenvolvimento de software, houve um movimento na comunidade de desenvolvimento de software, que deu origem aos Métodos Ágeis. Posteriormente, o conceito-base deste movimento evoluiu, de uma abordagem técnica para o âmbito gerencial, criando um novo enfoque de gerenciamento de projetos – o Gerenciamento Ágil de Projetos.

Neste capítulo são apresentados o histórico, o conceito, os valores e os princípios que norteiam os Métodos Ágeis, assim como são descritos os métodos mais freqüentemente utilizados pelas organizações. Também são exploradas as suas limitações, as indicações de aplicação e os resultados obtidos por empresas que já os adotaram. Por fim, são discutidos os fatores críticos de sucesso de projetos de desenvolvimento de software conduzidos com o uso desses métodos. O Gerenciamento Ágil de Projetos será abordado no tópico 2.5 deste trabalho.

# 2.4.1 Definição e Origem dos Métodos Ágeis de Desenvolvimento de Software

Os Métodos Ágeis de Desenvolvimento de Software surgiram como uma reação aos métodos clássicos de desenvolvimento de do reconhecimento da necessidade premente de se criar uma alternativa a estes "processos pesados 15", caracterizados pelo foco excessivo na criação de uma documentação completa (BECK, *et al*, 2001). Em meados dos anos 90, integrantes da comunidade de desenvolvimento de software, começaram a questionar estes processos, julgando-os pouco efetivos e, muitas vezes, impossíveis de serem colocados em prática (HIGHSMITH, 2002).

Sintetizando o pensamento deste grupo, Highsmith (*Ibid.*) menciona que a indústria e a tecnologia sofrem modificações tão aceleradas que acabam por "atropelar" os métodos clássicos. Highsmith *et al* (2002) ainda acrescentam que os clientes, na maioria das vezes, são incapazes de definir de forma clara e precisa, os requisitos do software, logo no início de um projeto de desenvolvimento, o que inviabiliza a adoção dos métodos clássicos em muitos projetos.

Como resposta a esta situação, muitos especialistas criaram métodos próprios para se adaptar às constantes mudanças exigidas pelo mercado e às indefinições iniciais dos projetos. O agrupamento desses métodos deu origem à família dos Métodos Ágeis de Desenvolvimento de Software. Sendo assim,

"[...] os Métodos Ágeis podem ser considerados uma coletânea de diferentes técnicas e métodos, que compartilham os mesmos valores e princípios básicos, alguns dos quais remontam de técnicas introduzidas em meados dos anos 70, como os desenvolvimentos e melhorias iterativos" (COHEN et al, 2003, p.2).

\_

<sup>&</sup>lt;sup>14</sup> Entre os métodos clássicos de desenvolvimento de software podem ser citados o *Modelo em Cascata* e os modelos *iterativos* e *em Espiral* (COHEN *et al*, 2003), apresentados no tópico 2.2.4 deste trabalho.

<sup>&</sup>lt;sup>15</sup> Heavyweight processes.

De fato, Cockburn e Highsmith (2001a) já haviam afirmado que a maioria das práticas propostas pelos Métodos Ágeis não tem nada de novo e que a diferença recai principalmente sobre o foco e os valores que os sustentam.

Segundo Cohen *et al* (2003), um dos primeiros questionamentos aos métodos clássicos de desenvolvimento de software foi feito por Schwaber, criador do *Scrum* (Método Ágil a ser apresentado no tópico 2.4.3). Para entender melhor os métodos clássicos de desenvolvimento de software baseados no SW-CMM, Schwaber (2002) elaborou um estudo junto aos cientistas da DuPont, que tinha por objetivo responder a seguinte pergunta: "Por que os processos definidos e defendidos pelo SW-CMM não promovem entregas consistentes"? Após analisarem seus processos de desenvolvimento de software, os cientistas chegaram à conclusão que, apesar do SW-CMM buscar a consistência, a previsibilidade e a confiabilidade dos processos de desenvolvimento de software, muitos destes processos ainda eram, de fato, imprevisíveis e impossíveis de serem repetidos. A explicação para tal recaía na complexidade dos processos propostos pelo SW-CMM, na conseqüente dificuldade de aplicação e também na necessidade de mudanças constantes e difíceis de serem antecipadas.

Schwaber (*op. cit.*) percebe que para que o desenvolvimento de software seja realmente ágil, deve-se aceitar as mudanças, ao invés de dar foco extremo à previsibilidade. Quase que simultaneamente, outros especialistas no assunto chegam à conclusão de que métodos que respondam às mudanças, tão rapidamente quanto estas venham a surgir e que incentivem a criatividade, são a única maneira de enfrentar e gerenciar os problemas do desenvolvimento de software em ambientes complexos (COCKBURN; HIGHSMITH, 2001a, SCHWABER, 2002).

Neste mesmo período, modelos de processos aplicados a outras indústrias, começam a ser analisados para servir como fonte de inspiração ao aprimoramento do processo de desenvolvimento de software (POPPENDIECK, 2001). O Modelo Toyota de Produção <sup>16</sup> foi alvo de atenção especial: enquanto as unidades fabris americanas trabalhavam a 100% de sua capacidade e mantinham grandes volumes de inventário de matérias-primas e de produtos

<sup>&</sup>lt;sup>16</sup> Modelo Toyota de Produção – para maiores informações ver Correa e Gianesi (1993) e Ferreira et al (2002).

acabados, a fábrica da Toyota mantinha o nível de estoque suficiente para um dia de operação e produzia somente o necessário para atender aos pedidos já colocados. Este modelo traduzido no princípio da *Lean Manufacturing*, visava à utilização mais eficiente dos recursos e a redução de qualquer tipo de desperdício e estava totalmente alinhado à filosofia da Administração da Qualidade Total<sup>17</sup>, criada pelo Dr. Edwards Deming (POPPENDIECK, 2001; FERREIRA *et al*, 2002). Deming (1990) acreditava que as pessoas desejavam fazer um bom trabalho e que os gerentes deveriam permitir que os trabalhadores do chão-de-fábrica tivessem autonomia para a tomada de decisões e a resolução de problemas. Além disso, estimulava o estabelecimento de uma relação de confiança com os fornecedores e defendia uma cultura de melhoria contínua dos processos e dos produtos. Enquanto as unidades fabris japonesas geravam produtos cada vez melhores e mais baratos, as fábricas americanas não conseguiam fazer o mesmo.

Com base nesta avaliação, Poppendieck (2001) listou 10 práticas que tornavam a *Lean Manufacturing* tão bem-sucedida e que, em seu entendimento, poderiam ser adaptadas e aplicadas ao desenvolvimento de software:

- Eliminação de gastos eliminar ou reduzir diagramas e modelos que não agregam valor ao produto final;
- Minimização de inventário minimizar artefatos intermediários, como documentos de requisitos e de desenho;
- 3. Maximização do fluxo utilizar o desenvolvimento iterativo para redução do prazo de entrega do software;
- 4. Atendimento à demanda atender às mudanças de requisitos;
- 5. Autonomia aos trabalhadores compartilhar a documentação e dizer aos programadores "o que" precisa ser feito e não "como" deve ser feito;
- 6. Atendimento aos requisitos dos clientes trabalhar perto dos clientes, permitindo que eles mudem suas opiniões ou seus desejos;

<sup>17</sup> A Administração da Qualidade Total ou *Total Quality Management* pode ser entendida como a extensão do planejamento de negócios de uma empresa, abrangendo o planejamento da qualidade (JURAN; GRYNA, 1991,

p. 210 - 214).

\_\_

- 7. Fazer certo da primeira vez testar o quanto antes e refazer o código se necessário;
- 8. Abolição da otimização local gerenciar o escopo de forma flexível;
- 9. Desenvolvimento de parceria com os fornecedores evitar relações conflitantes, tendo em mente que todos devem trabalhar juntos para gerar o melhor software;
- 10. Cultura de melhoria contínua permitir que o processo seja melhorado, que se aprenda com os erros e se alcance o sucesso.

Highsmith (2002) afirma, que de forma independente, Kent Beck e Ron Jeffreis percebem a importância dos princípios defendidos por Poppendieck (2001) durante um projeto de desenvolvimento de software na Chrysler e criam o projeto *Extreme Programming* (XP), Método Ágil de maior expressão atualmente. Simultaneamente, outras histórias começam a ecoar pelo mundo, como a vivenciada por Alistair Cockburn, que entrevistando profissionais do *IBM Consulting Group*, percebe que equipes de projetos bem-sucedidos se desculpavam por não ter seguido os processos formais, por não utilizar as ferramentas de alta tecnologia e por ter "simplesmente" trabalhado de forma próxima e integrada, enquanto membros de projetos mal-sucedidos afirmavam ter seguido as regras e processos e que não entendiam o que havia dado errado. Com base nesta experiência, Cockburn desenvolveu o *Crystal Method*, outro Método Ágil (HIGHSMITH, 2002).

Face ao exposto, percebe-se que o mundo do desenvolvimento de software passa por uma importante transformação: os métodos clássicos são vistos como não adequados a todas as situações e os especialistas reconhecem a necessidade de criação de novas práticas, orientadas a pessoas e flexíveis o suficiente para fazer frente a um ambiente de negócio dinâmico (COCKBURN; HIGHSMITH, 2001a). Os principais desafios enfrentados e que devem ser endereçados pelos novos métodos de desenvolvimento de software são assim sumarizados por Cockburn e Highsmith (*Ibid.*):

- 1. A satisfação dos clientes passar a ter precedência frente à conformidade aos planos;
- 2. As mudanças sempre ocorrem o foco deixa de ser como evitá-las e passa a ser como abraçá-las e como minimizar o seu custo ao longo do processo de desenvolvimento;
- 3. A eliminação das mudanças pode significar menosprezar condições importantes do negócio, ou seja, pode levar ao insucesso de uma organização;
- 4. O mercado espera um software inovador, com alta qualidade, que atenda aos requisitos do negócio e que esteja disponível em prazos cada vez menores.

#### 2.4.2 Manifesto para o Desenvolvimento Ágil de Software

No início de 2001, criadores e representantes dos chamados Métodos Ágeis de Desenvolvimento de Software - Extreme Programming, Scrum, Dynamic Systems Development Method, Adaptative Software Development, Crystal Methods, Feature-Driven Development, Lean Development, entre outros – se reuniram nos Estados Unidos para discutir alternativas aos tradicionais "processos pesados" de desenvolvimento de software (BECK et al, 2001). Estes especialistas foram enfáticos em dizer que não eram contra métodos, processos ou metodologias, sendo que muitos até mencionaram o desejo de resgatar o verdadeiro significado e a credibilidade destas palavras. Defendiam a modelagem e a documentação, mas não em excesso. Planejavam, mas reconheciam os limites do planejamento e da previsibilidade num ambiente turbulento (BECK et al, 2001).

A essência deste movimento é a definição de novo enfoque de desenvolvimento de software, calcado na agilidade, na flexibilidade, nas habilidades de comunicação e na capacidade de oferecer novos produtos e serviços de valor ao mercado, em curtos períodos de tempo (HIGHSMITH, 2004, p. xix). Como agilidade deve-se entender "a habilidade de criar e responder a mudanças, buscando a obtenção de lucro, em um ambiente de negócio turbulento" (HIGHSMITH, 2004, p. 16); ou ainda, "a capacidade de balancear flexibilidade e estabilidade" (Id., 2002). A agilidade não deve ser vista como falta de estrutura, mas está diretamente relacionada à capacidade de adaptação a situações diversas e inesperadas. Highsmith (2004, p. 16) enfatiza que a ausência de estrutura ou de estabilidade pode levar ao caos, mas que estrutura em demasia gera rigidez.

Como resultado do encontro, foi criada a Agile Alliance<sup>18</sup>, sendo publicado o Manifesto para Desenvolvimento Ágil de Software ou o Manifesto for Agile Software Development (BECK et al, 2001), cujo conteúdo19 é apresentado abaixo:

<sup>&</sup>lt;sup>18</sup> Agile Alliance – Organização sem fins lucrativos criada para auxiliar indivíduos e organizações que utilizam os Métodos Ágeis para o desenvolvimento de software.

"Nós estamos descobrindo melhores maneiras para desenvolver software, praticando e auxiliando os outros a fazê-lo. Através deste trabalho nós valorizamos:

- Os indivíduos e suas interações acima de processos e ferramentas;
- Software em produção acima da documentação exaustiva;
- Colaboração do cliente acima da negociação de contratos;
- Respostas às mudanças acima da execução de um plano.

Ou seja, embora haja valor nos itens à direita, nós valorizamos mais os itens à esquerda."

Segundo Cohen *et al* (2003, p. 6), este *Manifesto* tornou-se a peça-chave do movimento pelo desenvolvimento ágil de software, uma vez que reúne os principais valores dos Métodos Ágeis, que os distingue dos métodos clássicos de desenvolvimento.

Além do *Manifesto*, foram definidos os princípios que regem a maioria das práticas dos chamados Métodos Ágeis de Desenvolvimento de Software (AGILE ALLIANCE, 2005). Estes princípios são apresentados na Tabela 7 abaixo, de acordo com a ordem originalmente proposta:

Tabela 7 - Princípios dos métodos ágeis de desenvolvimento de software

	Princípios
1.	Nossa maior prioridade é satisfazer o cliente através da entrega rápida e contínua
	de um software de valor
2.	Pessoas de negócio e programadores devem trabalhar juntos, diariamente, ao
	longo de todo o projeto
3.	Aceite as mudanças de requisitos, mesmo que numa etapa avançada do
	desenvolvimento
4.	Entregue novas versões do software freqüentemente
5.	O software em funcionamento é a medida primária de progresso do projeto
6.	Construa projetos com pessoas motivadas. Ofereça a elas o ambiente e todo o
	apoio necessários e acredite em sua capacidade de realização do trabalho
7.	As melhores arquiteturas, requisitos e projetos emergem de equipes auto-

- Individuals and interaction over process and tools;
- Working software over comprehensive documentation;
- Customer collaboration over contract negotiation
- Responding to change over following a plan.

That is, while there is a value in the items on the right, we value the items on the left more."

<sup>&</sup>lt;sup>19</sup> "We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Princípios			
organizadas			
8. O método mais eficiente e efetivo de distribuir a informação para e entre uma			
equipe de desenvolvimento é a comunicação face a face			
9. Processos ágeis promovem desenvolvimento sustentado			
10. A atenção contínua na excelência técnica e num bom projeto aprimora a			
agilidade			
11. Simplicidade é essencial			
12. Equipes de projeto avaliam seu desempenho em intervalos regulares e ajustam			
seu comportamento de acordo com os resultados			

FONTE: AGILE ALLIANCE, 2005.

Pode-se dizer, então, que os Métodos Ágeis se caracterizam por serem incrementais, cooperativos, diretos e adaptativos. *Incrementais*, dadas as pequenas versões e ciclos rápidos de desenvolvimento; *cooperativos*, por estimular a proximidade com o cliente e a interação entre os programadores; *diretos*, pela simplicidade de aprendizado e de documentação; e, finalmente, *adaptativos*, pela habilidade de acomodar mudanças ao longo do projeto (ABRAHAMSSON *et al*, 2003; FOWLER, 2003).

Como principais diferenças entre os Métodos Ágeis e os clássicos de desenvolvimento de software podem ser citadas (FOWLER, 2003; CHIN, 2004):

- Os Métodos Ágeis são *adaptativos* e não preditivos: diferentemente dos enfoques clássicos que defendem o planejamento integral do escopo no início do projeto e um controle rígido de mudanças, os planos dos Métodos Ágeis são elaborados e adaptados ao longo do projeto, permitindo e, algumas vezes estimulado, a incorporação das mudanças requeridas pelo cliente;
- Os Métodos Ágeis são *orientados a pessoas* e não a processos: os processos clássicos têm desenvolvimento de software têm, em geral, a pretensão de funcionar independentemente de quem os executa. Já os Métodos Ágeis levam em consideração os indivíduos, sendo elaborados para auxiliá-las.

Um ponto interessante a salientar é que enquanto alguns defensores "radicais" dos Métodos Ágeis são categóricos em criticar e apontar as falhas dos métodos clássicos de desenvolvimento de software, outros especialistas, representantes tanto dos métodos clássicos quanto dos ágeis, têm uma postura mais amena, enxergando até mesmo uma possibilidade de integração entre as duas abordagens (GLASS, 2001; COHEN *et al*, 2003; PAULK, 2002;

GLAZER, 2001; HIGHSMITH, 2004). Glass (2001) apresenta uma análise do *Manifesto para o Desenvolvimento Ágil de Software* e menciona que, apesar de concordar com os pontos defendidos pelos praticantes dos Métodos Ágeis, não se deve desprezar os modelos clássicos e que ambos os lados têm pontos importantes a serem considerados e balanceados. Paulk (2002) defende que os princípios dos Métodos Ágeis devem ser seguidos por todos os profissionais que desenvolvem software, mas lembra que formalismo e disciplina devem ser levados em conta, especialmente quando o software a ser desenvolvido envolve severos requisitos de confiabilidade. Paulk (*Id.*, 2001) analisa o Método Ágil *Extreme Programming* (XP) sob a ótica do SW-CMM, apontando como este poderia auxiliar as organizações a alcançar os objetivos propostos pelo SW-CMM.

## 2.4.3 Principais Métodos Ágeis de Desenvolvimento de Software

Dentre os principais Métodos Ágeis de Desenvolvimento de Software podem ser citados: Extreme Programming, Scrum, Dynamic Systems Development Method, Adaptative Software Development, Crystal Method, Feature-Driven Developmen e Lean Development (COHEN et al, 2003; UDO; KOPPENSTEINER, 2003). A seguir é feita uma breve explanação sobre as suas principais características.

#### 2.4.3.1 Extreme Programming (XP)

De acordo com Cohen *et al* (2003, p.12) "o *Extreme Programming* (XP) é, indubitavelmente, o Método Ágil de maior expressão, criado nos últimos anos". Por se tratar do Método Ágil mais difundido é abordado com mais de detalhe neste trabalho.

O XP é um Método ágil para pequenas e médias equipes desenvolverem software, em ambientes com requisitos instáveis. Criado em 1998 por Kent Beck, Ron Jeffries e Ward Cuninghan, a partir de um projeto piloto na Chrysler (BECK *et al*, 1998), o XP vem ganhando cada vez mais adeptos, ampliando sua participação no mercado (HIGHSMITH, 2002). Beck (2000, p.xv) explica que o termo *Extreme* (extremo) é utilizado, dado que o XP reúne um conjunto de práticas de desenvolvimento já existentes e reconhecidas como "boas práticas" no desenvolvimento de software, mas as leva ao extremo, ao limite. Juric (2002) comenta que o XP é uma maneira eficiente, de baixo risco, científica e divertida de desenvolver software.

A premissa básica do XP é que, ao contrário do que se pensava há 30 anos, o custo de mudança de um software não aumenta exponencialmente com o avançar do projeto (BECK, 2004, p. 21-23). As novas técnicas desenvolvidas pelos especialistas em software, como os bancos de dados relacionais e a programação modular, entre outras, permitiram uma redução no custo da mudança (ver Ilustração 9). Desta forma, não se faz mais necessário evitar mudanças durante o desenvolvimento, sendo possível abandonar o *Modelo em Cascata* e usar o XP.

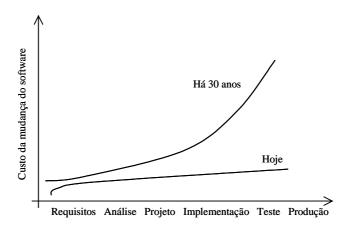


Ilustração 9 – Custo da mudança do software FONTE: BECK, 2004, p. 21-23

O XP baseia-se em 12 práticas ou regras concisas e diretas, listadas abaixo (BECK, *Ibid.*; COHEN *et al*, 2003, p.12, BECK; FOWLER, 2001, p. 72):

- Jogo do planejamento: no início de cada interação, clientes, gerentes e programadores se encontram para definir, estimar e priorizar os requerimentos. A idéia é que se elabore um plano aproximado no início no projeto e se faça um refinamento à medida que as necessidades e requisitos se tornem mais conhecidos;
- 2. *Programação em pares*: dois programadores utilizando o mesmo equipamento escrevem o código;
- 3. *Pequenas versões*: as versões devem ser tão pequenas quanto possível e trazerem valor para o negócio. Uma versão inicial do software deve ser colocada em produção após um pequeno número de iterações e, em seguida, outras versões devem ser disponibilizadas tão logo faça sentido;
- 4. *Metáforas*: clientes, gerentes e programadores criam metáforas ou conjunto de metáforas para modelagem do sistema;

- 5. *Projeto simples*: os programadores são estimulados a desenvolver o código do software o mais simples possível;
- 6. *Testes*: os programadores devem criar os testes de unidade antes ou mesmo durante o desenvolvimento do código do sistema. Os clientes, por sua vez, escrevem os testes de aceitação. Ao final de cada iteração a bateria de testes deve ser conduzida;
- 7. Reengenharia de software<sup>20</sup>: técnica que permite a melhoria de código sem a mudança de funcionalidade (BRASIL, 2001b, p. 186), deve ser executada pela equipe do projeto, o tempo todo;
- 8. *Integração contínua*: os programadores devem integrar os novos códigos ao software tão rapidamente e com a maior freqüência possível;
- 9. *Propriedade coletiva do código*: o código do programa deve ser propriedade de toda a equipe e qualquer integrante pode fazer alterações sempre que for necessário;
- 10. Cliente no local: o cliente deve trabalhar com a equipe de projeto a todo o momento, respondendo perguntas, realizando testes de aceitação e assegurando que o desenvolvimento do software esteja sendo feito a contento;
- 11. Semana de 40 horas: como trabalhar por longos períodos reduz o desempenho, o conteúdo de cada iteração deve ser planejado de forma a não haver necessidade de realização de horas extras, fazendo com que os programadores estejam renovados e ansiosos a cada manhã e cansados e satisfeitos à noite;
- 12. Padrão de codificação: no início do projeto deve ser criado um padrão de codificação, simples e aceito por toda a equipe, que deverá ser seguido de forma a não permitir a identificação de quem desenvolveu determinada parte do código e a auxiliar a condução do trabalho.

Especialistas concordam que o XP não é decorrência da aplicação destas práticas isoladamente, mas sim do resultado de sua combinação (COHEN *et al*, 2003, p.13). HIGHSMITH (2002) ainda ressalta que cinco princípios constituem a base do XP: comunicação, simplicidade, *feedback*, coragem e qualidade de trabalho.

<sup>&</sup>lt;sup>20</sup> Refactoring.

Cohen *et al* (2003, p.13) apresentam um resumo de características principais que norteiam a aplicação do *Extreme Programming*, retratado na Tabela 8.

Tabela 8 - Características principais para utilização do XP

Característica	Valores sugeridos	
Tamanho da Equipe	Equipes formadas por 2 a 10 integrantes	
Duração das iterações	Duração usual de 2 semanas por iteração	
Equipes distribuídas	Dada que a equipe deve trabalhar preferencialmente no mesr local físico, o XP não é indicado para equipes distribuídas	
Aplicações de alta criticidade	Pode ser utilizado no desenvolvimento de software de baixa, média ou alta criticidade	

FONTE: COHEN et al, 2003, p.13.

#### 2.4.3.2 *Scrum*

Criado por Ken Schwaber e Jeff Sutherland em 1996, como um método que aceita que o desenvolvimento de software é imprevisível e formaliza a abstração, o *Scrum* é aplicável a ambientes voláteis (SCHWABER, 2002). É uma abordagem empírica baseada na flexibilidade, adaptabilidade e produtividade, em que a escolha das técnicas de desenvolvimento fica a cargo dos programadores. Segundo Udo e Koppensteiner (2003), o *Scrum* se destaca dos demais Métodos Ágeis pela maior ênfase dada ao gerenciamento do projeto. Há atividades específicas de monitoramento e *feedback*, em geral, reuniões rápidas e diárias com toda a equipe, visando à identificação e à correção de quaisquer deficiências e/ou impedimentos no processo de desenvolvimento (SCHWABER; BEEDLE, 2001).

Schwaber (2002) sumariza assim os princípios-chave do Scrum:

- Equipes pequenas de trabalho, buscando a maximização da comunicação e da troca de conhecimento tácito e informal e minimização de *overhead*;
- Adaptação às solicitações de mudanças técnicas ou do cliente / usuário, assegurando a entrega do melhor software possível;
- Entregas freqüentes de versões que podem ser testadas, ajustadas, executadas, documentadas e liberadas para produção;
- Divisão do trabalho e das responsabilidades da equipe de projeto em pequenas entregas;
- Habilidade em entregar um software pronto quando da necessidade do cliente ou do negócio.

As características principais que norteiam a aplicação do *Scrum* são apresentadas na Tabela 9 (COHEN *et al*, 2003, p.15).

Tabela 9 - Características principais para utilização do Scrum

Característica	Valores sugeridos	
Tamanho da Equipe	O trabalho é dividido em equipes de até 7 pessoas	
Duração das iterações	Duração usual de 4 semanas por iteração	
Equipes distribuídas	Como o projeto pode ser constituído por várias pequenas equipes, há a possibilidade de que estas estejam distribuídas (descentralizadas)	
Aplicações de alta criticidade	Não há menção específica quanto à aplicabilidade do método para desenvolvimento de software de alta criticidade	

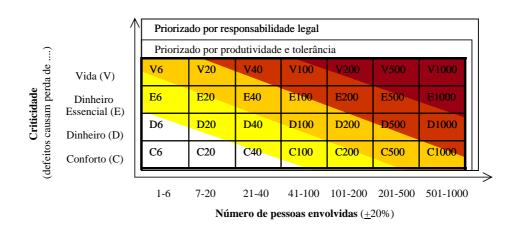
FONTE: COHEN et al, 2003, p.16-17.

## 2.4.3.3 Crystal Methods

Criado por Alistair Cockburn no início dos anos 90, a partir da crença de que os principais obstáculos enfrentados no desenvolvimento de produtos recaíam sobre os problemas de comunicação, os *Crystal Methods* dão grande ênfase às pessoas, à comunicação, às interações, às habilidades e aos talentos individuais, deixando os processos em segundo plano (UDO; KOPPENSTEINER, 2003; COCKBURN, 2004, COHEN *et al*, 2003). Correspondem a uma família de métodos organizados por cores, de acordo com o número de pessoas envolvidas (tamanho do projeto x necessidade de comunicação), com as prioridades do negócio e com a complexidade e a criticidade do software a ser desenvolvido, conforme mostra a Ilustração 10.

Apesar da estrutura proposta servir como um guia dos processos mais adequados a uma determinada situação, nos *Crystal Methods*, a definição final dos processos a serem utilizados é responsabilidade da equipe de projeto. Mas duas regras principais são sempre seguidas: ciclos de desenvolvimento incrementais com duração de no máximo quatro meses e reuniões de reflexão que estimulam a colaboração entre integrantes da equipe de projeto (UDO; KOPPENSTEINER, 2003; COCKBURN, 2004; COHEN *et al*, 2003).

#### Metodologias diferentes para diferentes ocasiões (tamanho do projeto, criticidade do sistema e prioridades)



**Ilustração 10** — **Esquema dos** *Crystal Methods* FONTE: COHEN *et al*, 2003, p. 16

Cohen *et al* (2003, p.15) apontam as características principais de projetos que utilizam os *Crystal Methods* (ver Tabela 12):

Tabela 10 - Características principais para utilização dos Crystal Methods

F F F F F F F F F F F F F F F			
Característica	Valores sugeridos		
Tamanho da Equipe	A família dos Crystal Methods acomoda equipes de qualquer tamanho,		
	preferencialmente compostas por pessoas talentosas		
Duração das iterações	Até 4 meses para projetos grandes e altamente críticos		
Equipes distribuídas	Os <i>Crystal Methods</i> foram concebidos para atender ao conceito de equipes distribuídas		
Aplicações de alta criticidade	Os <i>Crystal Methods</i> atendem a projetos críticos, incluindo aqueles que envolvem risco de vida e de valores monetários		

FONTE: COHEN et al, 2003, p.15.

## 2.4.3.4 Dynamic Systems Development Method (DSDM)

Originário da Inglaterra, em meados dos anos 90, o *Dynamic Systems Development Method* é controlado por um consórcio de empresas. Criado a partir do RAD – *Rapid Application Development*<sup>21</sup>, o DSDM é o único método ágil compatível com a ISO 9000. Seu ciclo de vida é divido nos seguintes estágios: a) Pré-projeto; b) Análise de Aderência; c) Estudo de

<sup>&</sup>lt;sup>21</sup> Desenvolvimento Rápido de Aplicações.

Negócio; d) Modelagem Funcional; e) Projeto e Desenvolvimento; f) Implementação, e, g) Pós-implementação. A idéia central do DSDM é que se deve primeiramente fixar o prazo e os recursos para, em seguida, definir e ajustar o número de funcionalidades a serem desenvolvidas (COHEN *et al*, 2003; UDO; KOPPENSTEINER, 2003).

Dadas a sua natureza, o DSDM não endereça um tamanho de equipe específico e não possui durações pré-determinadas para suas iterações (COHEN *et al*, op. *cit.*). Não foram encontradas na literatura recomendações específicas para utilização no desenvolvimento de aplicações de determinada criticidade ou para equipes centralizadas ou descentralizadas.

## 2.4.3.5 Feature-Driven Development (FDD)

O Feature-Driven Development, criado por Peter Coad e Jeff DeLuca em 1999, é um método de desenvolvimento de software específico para aplicações críticas de negócio (PALMER; FELSING, 2001). Diferentemente de outros Métodos Ágeis, o FDD se baseia em processos bem definidos e que podem ser repetidos. Sua abordagem se concentra nas fases de projeto e construção, com maior ênfase na modelagem, em um ciclo de vida iterativo e também em atividades de gerenciamento de projetos (UDO; KOPPENSTEINER, 2003). Os princípios-base do FDD são apontados abaixo (HIGHSMITH, 2002):

- Necessidade de se automatizar a geração de software para projetos de grande escala;
- Um processo simples e bem definido é fundamental;
- As etapas de um processo devem ser lógicas e óbvias para cada integrante da equipe de desenvolvimento;
- Bons processos atuam na retaguarda, permitindo que a equipe se dedique ao alcance dos resultados;
- Ciclos de vida curtos e iterativos são mais indicados.

Um projeto conduzido pelo método FDD possui as seguintes etapas: a) Desenvolvimento de um modelo geral; b) Construção da lista de funcionalidades; c) Planejamento por funcionalidades; d) Projeto e desenvolvimento por funcionalidades.

A Tabela 11 apresenta as principais características de um projeto desenvolvido pelo método FDD (COHEN *et al*, 2003, p.18):

Tabela 11 - Características principais para utilização do FDD

Característica	Valores sugeridos	
Tamanho da Equipe	Variável de acordo com a complexidade das funcionalidades a serem	
	desenvolvidas	
Duração das iterações	Até 2 semanas	
Equipes distribuídas	O FDD foi criado para trabalhar com equipes múltiplas e, apesar o	
	não haver indicação formal para equipes distribuídas, é passível de	
	adaptação	
Aplicações de alta criticidade	Indicado para desenvolvimento de software crítico	

FONTE: COHEN et al, 2003, p.15.

## 2.4.3.6 Lean Development (LD)

Com raízes na indústria automotiva dos anos 80, em especial no Modelo Toyota de Produção, o *Lean Development* é considerado o Método Ágil com maior foco estratégico. Iniciado por Bob Charette, o LD tem como principais objetivos reduzir em um terço o prazo, o custo e o nível de defeitos no desenvolvimento de software. Para tanto, requer um grande comprometimento da alta administração e uma predisposição a mudanças radicais (COHEN *et al*, 2003; UDO; KOPPENSTEINER, 2003). HIGHSMITH (2002) aponta como princípios fundamentais do *Lean Development*:

- A satisfação do cliente é a prioridade principal;
- Prover sempre o maior valor possível para o dinheiro;
- O sucesso depende da participação ativa dos clientes;
- Todo projeto baseado no LD requer um esforço conjunto de toda a equipe;
- Tudo pode ser mudado;
- Domine, não aponte as soluções;
- Complete, n\u00e4o desenvolva;
- Prefira uma solução a 80% hoje, a uma solução a 100% amanhã;
- O minimalismo é essencial;
- A necessidade determina a tecnologia;
- O incremento do produto corresponde a um incremento de funcionalidade e não de tamanho;
- Nunca force o LD além de seus limites.

Cohen *et al* (2003, p. 19) afirma que "[...] como o *Lean Development* é mais uma filosofia de gerenciamento que um processo de desenvolvimento", os itens referentes ao tamanho da equipe, à duração das iterações, ao tratamento de equipes centralizadas ou distribuídas e à criticidade da aplicação não são diretamente endereçados pelo método.

### 2.4.3.7 Adaptative Software Development (ASD)

Criado por Highsmith como uma evolução do RAD – *Rapid Application Development* em 1992, o *Adaptative Software Development* propõe uma forma alternativa de se enxergar o desenvolvimento de software nas organizações (HIGHSMITH, 2002). O ASD foi projetado para lidar com ambientes repletos de incertezas e complexos. Segundo Udo e Koppensteiner (2003), o método estimula a aprendizagem durante o processo de desenvolvimento e a adaptação constante às novas realidades do negócio e do projeto. Além disso, encoraja o desenvolvimento iterativo e incremental, com a liberação constante de novas versões. O ASD oferece estrutura e orientação suficiente para evitar que os projetos se tornem caóticos, sem, entretanto, trazer uma rigidez indesejada que venha a suprimir a criatividade. Neste método, o papel do gerente de projetos é favorecer a colaboração entre a equipe de desenvolvimento e o cliente.

De acordo com Highsmith (2002), o ASD é indicado para equipes pequenas, mas pode ser adaptado para equipes maiores. Com relação aos demais atributos – duração das iterações, apoio a equipes distribuídas e desenvolvimento de aplicações de alta criticidade – não foi encontrada uma indicação precisa na literatura.

# 2.4.3.8 Resumo das Características Principais dos Métodos Ágeis

Nos tópicos anteriores foram apresentadas as principais características de alguns dos chamados Métodos Ágeis de Desenvolvimento de Software. É importante mencionar que estes métodos foram selecionados por serem os mais citados na literatura, servindo de base para a definição das variáveis da pesquisa.

Como visto, apesar dos Métodos Ágeis terem uma essência ou valores em comum, há algumas diferenças significativas entre eles. A Tabela 12 apresenta um resumo das principais características dos Métodos Ágeis analisados, com uma linha específica destinada à análise da incorporação ou não de práticas relacionadas ao gerenciamento de projetos:

Tabela 12 - Características principais dos métodos ágeis selecionados

Característica	XP	Scrum	Crystal	FDD	ASD	LD	DSDM
			Methods				
Tamanho da Equipe	2-10	1-7	Variável	Variável	Variável		
Duração das iterações	2 semanas	4 semanas	< 4 meses	< 2 semanas			
Equipes distribuídas	Não	Adaptável	Sim	Adaptável			
Aplicações de alta	Adaptável	Adaptável	Todos os	Adaptável		Não def	finido
criticidade			tipos			_	
Práticas de	Poucas	Muitas	Poucas	Muitas	Muitas		
gerenciamento de							
projetos							

FONTE: Adaptado de COHEN et al, 2003, p. 23.

# 2.4.4 Aplicação dos Métodos Ágeis nas Organizações

Há uma grande variedade de Métodos Ágeis de Desenvolvimento de Software, cada qual sugerindo práticas específicas, cuja adoção traz mudanças às rotinas das organizações. De acordo com Nerur *et al* (2005, p.74), "[...] as alterações nos processos de desenvolvimento de software representam um fenômeno complexo de mudança organizacional que não pode ser alcançado pela mera substituição de ferramentas e tecnologias". Estas mudanças têm impacto significativo em vários aspectos da organização: na estrutura, na cultura e nas práticas gerenciais. Conhecer a amplitude deste fenômeno é fator crítico para o planejamento e o gerenciamento destas mudanças. Sendo assim, antes de uma organização adotar e implementar qualquer um dos Métodos Ágeis é fundamental que avalie a dimensão do impacto e a sua prontidão para tal (AMBLER, 2002c; COHEN *et al*, 2003; FOWLER, 2003; NERUR *et al*, 2005).

Para facilitar o entendimento e retomar as principais características e diferenças entre os enfoques clássico e ágil de desenvolvimento de software, já discutidas ao longo deste trabalho, é traçado paralelo entre eles, na Tabela 13. Estas diferenças entre as abordagens sugerem que as organizações devem repensar suas metas, seus objetivos e reconfigurar seus componentes humanos, gerencial e tecnológico, de forma a alcançar uma implementação

bem-sucedida dos métodos ágeis (NERUR *et al*, 2005, p.75). Enfocando o componente gerencial, estas diferenças podem demandar até mesmo uma alteração no enfoque de gerenciamento de projetos utilizado para as iniciativas de desenvolvimento de software.

Tabela 13 - Comparação entre os enfoques clássico e ágil de desenvolvimento de software

	Enfoque Clássico	Enfoque Ágil (Métodos Ágeis)
Premissa Fundamental	O software é totalmente especificável e previsível e pode ser construído através de um planejamento meticuloso e extensivo	Software adaptativo e de alta qualidade pode ser construído por pequenas equipes, com o uso de princípios como a melhoria contínua do projeto e o desenvolvimento e testes baseados no rápido <i>feedback</i> e na aceitação de mudanças
Estilo de Gerenciamento Comando e controle Liderança e colabo		Liderança e colaboração
		Equipes auto-organizadas, encorajando o intercâmbio de papéis
Papel do Cliente Importante		Crítico
Modelo de Desenvolvimento	Modelo de ciclo de vida – Modelo em Cascata, Espiral e iterativos	Métodos Ágeis
Tecnologia	Sem restrições	Favorece a tecnologia orientada a objetos

FONTE: ADAPTADO DE NERUR et al, 2005, p. 75.

Ambler (2002c) discute os fatores que influenciam a adoção bem-sucedida de Métodos Ágeis, enfatizando que o ponto mais importante para que isto aconteça, é a existência de uma compatibilidade entre os conceitos e valores da organização e dos Métodos Ágeis, ou seja, o autor discute claramente a questão da cultura da organização. Na mesma linha, Nerur *et al* (2005) destacam que os valores, as normas e os padrões estabelecidos e reforçados pelas organizações ao longo do tempo se refletem nas políticas e nas rotinas da empresa e exercem considerável influência nos processos de tomada de decisão, nas estratégias de solução de problemas, nas práticas voltadas à inovação, nos filtros de informação, nos relacionamentos interpessoais e nas negociações. Como a cultura e a forma de pensar das pessoas não são facilmente modificáveis, pode-se dizer que a adoção de Métodos Ágeis seja indicada apenas a algumas organizações (AMBLER, 2002; NERUR *et al*, 2005).

Um segundo fator importante a ser considerado quando da decisão pela implementação de Métodos Ágeis, de acordo com Ambler (*op. cit.*) é o projeto e as características do negócio. Questões como – Os trabalhos têm sido conduzidos de forma incremental? Qual é a motivação da equipe de projeto? Qual o nível de apoio que a equipe de desenvolvimento pode esperar? Os recursos adequados estão disponíveis? Quão voláteis são os requisitos do projeto?

- são fundamentais para esta avaliação. Com relação ao último ponto, Bohem (2002) chega a sugerir que os métodos clássicos deveriam ser preferidos quando o índice de alteração de requisitos do projeto for inferior a 1% ao mês.

Um terceiro aspecto destacado por Ambler (*op. cit.*) é a necessidade de definição de um *champion*, ou seja, de um profissional que assuma os desafios do projeto, de forma que a equipe de desenvolvimento possa trabalhar com tranqüilidade. Ampliando esta análise, uma vez que os Métodos Ágeis valorizam e depositam elevado grau de confiança no conhecimento tácito e na capacidade de tomada de decisões de cada indivíduo, Bohem (*op. cit.*) enfatiza a importância de se ter também uma equipe de projeto bem treinada e composta por especialistas. Com relação a este aspecto, apesar de concordar com a necessidade de formação de uma equipe especializada, Nerur *et al* (2005) chama a atenção para que não se crie uma cultura de elitismo dentro do grupo de desenvolvimento, que pode afetar o moral e o comprometimento de outros profissionais da empresa, não integrantes da equipe.

É inegável também que os Métodos Ágeis trazem consigo uma mudança radical na relação cliente – fornecedor. Cockburn e Higsmith (2001a), Bohem (2002) e Nerur *et al* (2005) apontam o envolvimento e a participação dos clientes como fatores imprescindíveis para o sucesso da implementação dos métodos ágeis. Os clientes devem estar totalmente comprometidos com o projeto, trabalhar com espírito de colaboração, possuir o conhecimento necessário, ter autonomia para a tomada de decisões, além de estar disponíveis para sanar as dúvidas quando necessário. Entretanto, por se tratar de um item extremamente crítico no processo e que demanda tempo, paciência e esforço consideráveis para o estabelecimento de um ambiente de respeito e confiança mútua entre clientes e fornecedores, é mencionado por alguns autores, entre eles Nerur *et al* (*Ibid.*), como um dos obstáculos à utilização dos Métodos Ágeis.

Compartilhando os pontos discutidos nos parágrafos anteriores, Cohn e Ford (2003, p. 74) acrescentam que transição de um processo clássico com ênfase no "planejamento – execução – controle", para um processo ágil, afeta não apenas a equipe de desenvolvimento de software, mas também outras equipes, departamentos, assim como o corpo gerencial da organização. Tomando por base a experiência empírica de implementações de Métodos Ágeis em várias organizações, abrangendo tanto projetos simples como projetos complexos, equipes centralizadas ou distribuídas, os autores sugerem uma abordagem diferenciada junto aos

principais envolvidos nos processos – programadores, alta gerência e a área de Recursos Humanos – para que se garanta uma implementação bem-sucedida deste novo enfoque de desenvolvimento de software.

Segundo Cohn e Ford (2003, p. 74 - 75), usualmente os programadores respondem à implementação de Métodos Ágeis com um misto de ceticismo, entusiasmo, otimismo, ou mesmo, resistência. Como este novo enfoque desenvolvimento de software normalmente prioriza a entrega do código à geração de uma documentação extensiva, a adaptação ao planejamento completo e, as pessoas e suas interações aos processos e ferramentas (BECK *et al*, 2001), Cohn e Ford (*op. cit.*) afirmam que há sentimentos controversos entre os integrantes da equipe durante o processo de transição: alguns se sentem perdidos ou sem um direcionamento, pela ausência de um cronograma formal ou de uma documentação completa de requisitos; outros empolgados, pelo reconhecimento de sua capacidade e pela autonomia concedida; alguns crêem erroneamente que a adoção dos Métodos Ágeis tem o intuito de estimular a microgerência, dados os encontros e as reuniões constantes entre gerentes e equipe, em métodos como o *Scrum* e XP.

Em face desta situação, Cohn e Ford (2003, p. 75), assim como Ambler (2002c), defendem uma passagem gradual dos processos clássicos de desenvolvimento para os Métodos Ágeis, tornando o período de transição mais tranquilo.

A idéia de uma equipe formada por talentos, citada por Boehm (2002) é ratificada por Cohn e Ford (2003, p. 75) e por Ambler (2002c), que explicam ainda que diferenças grandes de produtividade numa equipe podem trazer impactos negativos ao projeto, uma vez que num Método Ágil, a equipe se dedica apenas às atividades essenciais para a entrega do software. Não se deve esquecer, contudo, que uma pequena queda de produtividade é esperada durante a transição, até que toda a equipe esteja ambientada com a nova dinâmica de trabalho. Fowler (2003) destaca que a equipe técnica não pode ser responsável por todo o trabalho por si só, sendo fundamental o papel dos gerentes, fornecendo o direcionamento, auxiliando a definição das prioridades de negócio, removendo obstáculos e negociando os prazos de entrega. Com esta colocação, Fowler (*Ibid.*) enfatiza a importância do gerenciamento de projetos mesmo no desenvolvimento de software conduzido com o uso de Métodos Ágeis.

Ao considerar a questão de equipes distribuídas, Cohn e Ford (*op. cit.*) são enfáticos em dizer que, mesmo havendo a necessidade de se organizar um projeto de forma descentralizada, deve-se fazer o possível para que nas primeiras semanas de trabalho, a maior parte da equipe esteja reunida e somente após este período as equipes sejam distribuídas. Os autores justificam este ponto ao afirmar que equipes que trabalham segundo os Métodos Ágeis necessitam tomar decisões de forma mais rápida que nos processos tradicionais, mas para tanto, recorrem a uma comunicação mais freqüente e normalmente informal. Sem esta proximidade inicial, a comunicação pode ser comprometida.

Analisando um outro grupo de interessados, Cohn e Ford (2003, p.76) apontam que normalmente a alta gerência representa um desafio singular à adoção dos Métodos Ágeis. Basicamente as suas preocupações recaem sobre quatro pontos fundamentais, extremamente vinculados à visão do Gerenciamento Clássico de Projetos:

- 1. Como é possível prometer novas funcionalidades aos clientes?
- 2. Como mensurar o progresso do projeto?
- 3. Quando o projeto acaba?
- 4. Como a utilização de um Método Ágil afetará outros grupos?

Apesar de pertinentes, a origem destes questionamentos está na dificuldade que muitos gerentes têm em abrir mão do processo clássico de planejamento e controle, da solicitação de compromissos formais de entrega, mesmo sabendo que raramente estes objetivos são cumpridos pelas equipes de desenvolvimento<sup>22</sup>. Cohn e Ford (2003, p.76) e Highsmith (2004) salientam que gerentes que temem que com os Métodos Ágeis não seja possível a fixação de datas de entrega de produtos a clientes, deveriam lembrar que, no passado, os planos formais provedores de "garantias" de prazo, custo e qualidade estavam usualmente errados, superestimados, ou ambos. Em organizações com histórico de estimativas incorretas de projetos, convencer a alta gerência sobre os benefícios dos Métodos Ágeis pode não ser algo

<sup>&</sup>lt;sup>22</sup> Pesquisa publicada pelo STANDISH GROUP INTERNATIONAL (2003), referente a projetos de desenvolvimento de software, aponta que usualmente os projetos ultrapassam em 82% os prazos inicialmente previstos. Em 1999, este índice de atraso chegou ao valor aproximado de 222%.

difícil, bastando uma avaliação dos resultados passados frente aos objetivos iniciais de prazo, custo e qualidade e a apresentação dos benefícios potenciais da nova abordagem. Já nos casos em que a equipe de desenvolvimento entrega seus projetos sistematicamente no prazo e no custo, a utilização dos Métodos Ágeis pode ser justificada com vistas à redução de prazos de entrega, à redução das equipes de trabalho, o que significaria um ganho para a organização.

Apesar dos Métodos Ágeis proporem uma mudança do paradigma de "comando e controle" para "liderança e colaboração", conforme explica Nerur *et al* (2005, p.75), isto não significa que sejam alheios à necessidade de mensuração do progresso dos projetos. Segundo Cohn e Ford (2003, p.77), os Métodos Ágeis podem oferecer um acompanhamento adequado do projeto, por meio da utilização de relatórios específicos a cada iteração, que contêm dataschave, comparação de resultados reais *versus* planejados, métricas principais e até mesmo a identificação dos riscos do projeto. Estas são práticas do gerenciamento de projetos, mas que neste caso, são aplicadas a cada interação e não ao projeto como um todo.

Outra preocupação bastante comum à alta gerência, refletida na terceira questão, é que aparentemente um projeto realizado segundo um Método Ágil tende a não ter fim, uma vez que as iterações podem continuar enquanto houver itens prioritários, definidos pelo cliente, a serem desenvolvidos. Cohn e Ford (2003, p.77) apontam como solução para esta questão, a definição de um intervalo de prazo e custo, vinculado a um escopo macro, que servem como uma estimativa preliminar do projeto, a ser considerada durante a execução do projeto. A equipe deve buscar entregar uma versão básica e funcional do software, no prazo mínimo deste intervalo e, a partir daí, são negociadas entregas complementares (outras iterações). Desta forma, Cohn e Ford (2003, p. 77), afirmam que é possível minimizar o desconforto quanto à finalização do projeto.

Considerando a quarta questão, a implementação de um Método Ágil pode afetar além da equipe de desenvolvimento, os gerentes, os clientes e, até mesmo, áreas ou departamentos internos à empresa que, numa análise inicial, não têm nenhum vínculo aparente com o projeto. Sendo assim, é fundamental que a alta gerência tenha ciência de quais grupos ou departamentos serão afetados pela adoção dos Métodos Ágeis, para que se estabeleça um consenso quanto à forma de trabalho e se evite desgastes ou problemas futuros. Neste contexto, deve-se destacar a importância da participação da área ou departamento de Recursos Humanos no processo de transição para a utilização dos Métodos Ágeis (COHN; FORD,

2003, p. 78). Representantes desta área devem ser envolvidos logo no início da implementação, para que tenham ciência da nova forma de trabalho e forneçam todo o apoio necessário, sanando dúvidas e amenizando ansiedades dos envolvidos no processo de mudança.

Complementando esta análise, COHEN *et al* (2003, p. 31) selecionam um conjunto de lições aprendidas, que consideram úteis quando da decisão pela adoção de um Método Ágil, algumas das quais rebatem as críticas mais comuns atribuídas a este novo enfoque de desenvolvimento de software:

- Os Métodos Ágeis podem ser aplicados a equipes de qualquer tamanho, entretanto, deve-se ter em mente que quanto maior a equipe, maior a dificuldade de comunicação;
- Experiência é importante para que um projeto ágil seja bem-sucedido, mas a experiência técnica em desenvolvimento de software é muito mais importante que a experiência prévia com os Métodos Ágeis;
- Os Métodos Ágeis requerem menos treinamento formal que os métodos clássicos. Técnicas como a programação em pares minimizam a necessidade de treinamento, pois as pessoas aprendem umas com as outras. Os treinamentos formais podem ser realizados por meio de auto-estudo;
- Um software crítico e de alta confiabilidade pode ser construído com a utilização de Métodos Ágeis. Neste caso, é fundamental que os requisitos de desempenho sejam explicitados logo no início do projeto e os níveis adequados de teste planejados. Ao solicitar *feedback* constante, incentivar a participação dos clientes e a definição de prioridade por eles, os Métodos Ágeis tendem a antecipar resultados e minimizar os riscos do projeto;
- Os três principais fatores críticos de sucesso para um projeto ágil são: cultura, pessoas e comunicação. Os Métodos Ágeis exigem uma compatibilidade cultural para serem bemsucedidos; profissionais competentes são fundamentais: os Métodos Ágeis usam menos técnicos, porém mais qualificados; equipes centralizadas facilitam a comunicação e a interação próxima com cliente é fator base para seu funcionamento;
- Os Métodos Ágeis permitem a identificação rápida de eventuais problemas no projeto, através de pequenos sinais, como a queda da motivação da equipe nas reuniões diárias, atrasos nas entregas das iterações e a geração de documentação desnecessária;

- A documentação deve ser considerada um custo e sua extensão deve ser determinada pelo cliente. Deve-se buscar desenvolver uma comunicação mais efetiva, mantendo a documentação formal em um patamar mínimo necessário.

A forma como um Método Ágil é introduzido em uma organização e os cuidados tomados durante este processo podem determinar o sucesso ou o fracasso da iniciativa. Outro importante desafio enfrentado pelos gerentes das organizações refere-se à escolha do Método Ágil mais apropriado para o momento e o projeto em questão, em face da variedade de métodos atualmente disponíveis no mercado (NERUR et al, 2005, p.77). Ambler (2002c) e Cohn e Ford (op. cit.) mencionam que, se após uma análise detalhada, os Métodos Ágeis não se apresentarem totalmente compatíveis com o projeto ou com os princípios da organização, mas suas idéias ainda assim despertarem interesse, pode-se partir para uma adoção parcial. Nesta estratégia, deve-se identificar os pontos de melhoria prioritários no processo clássico de desenvolvimento de software e aplicar algumas práticas dos Métodos Ágeis, visando ao aprimoramento. Obtendo-se um resultado positivo, procede-se a uma nova seleção de áreas de melhoria e implantação de novas técnicas, até que todo o Método Ágil tenha sido adotado.

Por fim, Highsmith (2004), Cohn e Ford (2003) e Nerur *et al* (2005) ressaltam que muitas das inseguranças e incertezas inerentes a este processo de transição podem ser minimizadas com a adoção de um enfoque de gerenciamento de projetos adequado e compatível com o desenvolvimento de software conduzidos com o uso de Métodos Ágeis.

# 2.4.5 Resultados da Aplicação dos Métodos Ágeis

Como os Métodos Ágeis de Desenvolvimento de Software são relativamente recentes, poucos são os dados empíricos disponíveis referentes aos resultados de sua aplicação nas organizações, alguns dos quais são apresentados a seguir.

Pesquisa realizada por Reifer (2002, p. 14-17) em 32 organizações, representando 28 empresas de dez segmentos de mercado distintos, apontou que dentre elas, 14 organizações adotavam Métodos Ágeis, todas motivadas por um histórico de baixo desempenho dos projetos de desenvolvimento de software quanto ao cumprimento dos objetivos de prazo e custo. Surpreendentemente, a maioria das empresas analisadas era classificada como nível

dois ou superior, na escala de maturidade nos processos de desenvolvimento de software proposta pelo SW-CMM, ou seja, possuíam processos relativamente maduros. No entanto, estas empresas buscavam algo novo, que pudesse reverter o quadro de mau desempenho consistente de seus projetos. A distribuição destas empresas entre os segmentos de atuação, assim como alguns detalhes quanto ao início da prática, à quantidade de projetos realizados com o uso dos Métodos Ágeis e ao estágio de implementação do novo método são apresentados na Tabela 14.

Tabela 14 - Características das empresas pesquisadas

Indústria	# Empresas que utilizam Métodos Ágeis	# de Projetos	Início da Utilização de Métodos Ágeis	Estágio da Implementação
Aeroespacial	1	1	2001	Descoberta
Computação	2	3	2000	Piloto
Consultoria	1	2	2000	Piloto
Comércio Eletrônico	5	15	2000	Produção
Pesquisa	1	1	2000	Piloto
Software	2	4	2000	Produção
Telecomunicações	2	5	2000	Produção
Total	14	31	n/a	n/a

FONTE: ADAPTADO DE REIFER, 2002, p. 15.

Os projetos pesquisados são qualificados como de pequeno porte (com menos de 10 participantes), em geral, internos, de baixo risco, com duração menor que 12 meses, mas que exigiam elevado grau de flexibilidade.

Com relação aos resultados, 50% das organizações que responderam a pesquisa conseguiram mensurar os ganhos obtidos com a utilização de Métodos Ágeis de Desenvolvimento de Software de forma concreta, sendo que muitas, por possuírem métricas anteriores, realizaram comparações bastante efetivas. Reifer (2002, p.15) aponta como principais ganhos, já normalizados entre todos os participantes, os seguintes itens:

- Incremento de produtividade: ganhos de 15% a 23% de produtividade frente aos indicadores da indústria;
- Redução de custos: 5% a 7% de redução de custos quando comparado aos indicadores da indústria;

- Redução do tempo de entrega: 25% a 50% de redução de prazo comparando-se com projetos anteriores realizados pelas empresas;
- Melhoria de qualidade: cinco empresas que possuíam dados concretos apontaram que a taxa de defeitos estava no mesmo nível dos projetos anteriores quando da liberação de produtos e aplicações.

As demais organizações analisadas, que não possuíam indicadores formais, mensuraram seus ganhos através da opinião dos principais interessados nos projetos, listando como benefícios da utilização dos Métodos Ágeis, o alto grau de motivação dos profissionais envolvidos, a elevação da moral da equipe e alguns outros argumentos intangíveis.

De forma unânime, todas as organizações afirmaram sua intenção de continuar ou mesmo de estender a utilização dos Métodos Ágeis, considerando-a uma experiência bastante proveitosa e bem-sucedida. Mas apesar dos resultados animadores, Reifer (2002, p.15), chama a atenção para que não sejam tiradas conclusões precipitadas, nem se faça uma generalização dos resultados, dados o tamanho da amostra (14 organizações) e o tipo de projetos envolvidos (projetos pequenos, de baixo risco, em ambiente relativamente controlado).

Maurer e Martel (2002) relataram o caso de uma companhia – Bitonic Solutions Inc. – sediada no Canadá, cuja equipe de sistemas, composta por nove profissionais, desenvolveu uma aplicação para comércio eletrônico. O processo de desenvolvimento foi iniciado segundo uma abordagem orientada a objetos *ad hoc*, sendo adotado o método ágil *Extreme Programming* (XP), depois de determinado período. Para a mensuração da produtividade do desenvolvimento, os autores definiram três indicadores principais, cujos valores foram divididos pelo esforço despendido para execução (em horas), a saber:

- Novas linhas de código (NLOC)
- Número de novos métodos (# métodos);
- Número de novas classes (# classes).

Maurer e Martel (2002) coletaram métricas do desenvolvimento nos dois períodos (anterior e posterior ao uso do XP) e reportaram ganhos de produtividade que variaram de 66,3% a 302,1%, conforme mostra a Tabela 15.

Tabela 15 - Ganhos de produtividade com XP em projetos Web

Indústria	NLOC / esforço	# Métodos / esforço	# Classes / esforço
Média anterior ao uso do XP	10,2	0,36	0,05
Média com uso do XP	17	1,45	0,21
% de Variação	66,3%	302,1%	282,6%

FONTE: ADAPTADO DE MAURER: MANTEL, 2002.

Poole e Huisman (2001) relataram o uso bem-sucedido do *Extreme Programmming* (XP) na companhia Iona Technologies, no desenvolvimento de um *middleware*. Os autores mencionaram que a empresa não possuía um processo de desenvolvimento e manutenção de software definido, sendo que a falta de qualidade no processo acabava por se refletir na qualidade do produto. Um processo de reengenharia de software foi iniciado em 1997 sem, entretanto, alcançar os resultados desejados. Em 2000, a companhia decidiu implementar, de forma gradativa, as práticas do XP em seu processo de manutenção de software. A Iona Technologies alcançou excelentes resultados, reportando ganhos de produtividade na ordem de 67%, o que possibilitou uma redução no seu quadro de pessoal de 36 para 25 profissionais. A empresa obteve também ganhos de qualidade, com uma queda bastante significativa do número de erros encontrados pelos clientes, que passaram de 33 erros ao mês para quatro erros ao mês. Estes resultados foram atribuídos, principalmente, à adoção da prática de programação em pares.

Em 2001, pesquisa conduzida pelo Cutter Consortium (COCKBURN; HIGHSMITH, 2001b), com mais de 200 profissionais de diferentes empresas dos Estados Unidos, Europa, Índia e Austrália, sobre o uso dos métodos clássicos e ágeis de desenvolvimento de software chegou a três conclusões interessantes:

- Um aumento do número de empresas que utilizavam algum Método Ágil de Desenvolvimento de Software, quando comparado à pesquisa similar realizada em 2000;
- Os projetos de desenvolvimento realizados segundo o enfoque ágil apresentaram melhor desempenho em termos de atendimento aos requisitos do negócio, satisfação do cliente e qualidade, que os projetos conduzidos nos moldes clássicos;
- Os Métodos Ágeis receberam melhor pontuação no quesito "moral dos profissionais", resultado este considerado, na época, surpreendente pelo fato de apenas 12% dos respondentes serem analistas ou programadores.

Similarmente a Reifer (2002), Cockburn e Highsmith (2001b) não generalizam estas conclusões, mas consideram, em especial o terceiro item, um indício bastante positivo da contribuição dos Métodos Ágeis em relação à motivação das equipes de projeto.

A aplicação bem-sucedida do método *Extreme Programming* (XP) no desenvolvimento de um sistema de missão crítica<sup>23</sup>, voltado à coordenação de equipes da Hewlett-Packard distribuídas ao redor do mundo, foi relatada por Gawlas (2004, p. 18-24). O autor aponta que o processo de desenvolvimento, pelo método XP, foi dividido em sete etapas, cada qual com um resultado e um prazo inicial pré-determinados. Várias adaptações foram feitas ao longo do projeto. Além de destacar práticas relacionadas ao XP, Gawlas (2004, p. 18-24) chama a atenção para a abordagem de gerenciamento de projetos utilizada, caracterizada como uma combinação entre o conhecimento e a experiência da equipe de projeto e a necessidade de balanceamento entre o enfoque ágil e o Gerenciamento Clássico de Projetos adotado pela companhia. Após nove meses e dentro do prazo previsto, o software entrou em produção com poucos defeitos, que puderam ser corrigidos de forma rápida e tranqüila. Gawlas (2004, p. 18-24) salienta também que um ponto de destaque foi a motivação da equipe de projeto.

Bonato (2004, p. 107-172), por sua vez, apresenta um caso de adoção do *Extreme Programming* (XP) em renomada empresa jornalística brasileira, para o desenvolvimento de uma aplicação que visava à reformulação do pagamento de bonificação às empresas publicitárias, garantindo maior controle ao processo. O projeto foi iniciado segundo o enfoque clássico de desenvolvimento de software, utilizado pela companhia. Contudo, logo no início do projeto, a equipe se deparou com problemas como a impossibilidade de detalhamento prévio de requisitos e com a dificuldade de entregar o software no prazo, dada a documentação bastante extensiva exigida pelo processo. Neste momento, houve uma decisão pela adoção do XP, buscando aliar qualidade e produtividade diante de requisitos instáveis. Ao final, o projeto foi considerado um sucesso, com ganhos de qualidade (mensurados por

<sup>&</sup>lt;sup>23</sup> Sistema (ou software) de missão crítica é aquele em que a vida, a saúde ou a segurança das pessoas ou organizações podem ser comprometidas se a qualidade do software não for extremamente alta (TURK *et al*, 2005, p.29).

uma redução de 62,9% no número de erros reportados pelos usuários após a implementação, quando comparados com outros projetos conduzidos pelo Jornal) e com ganhos de produtividade estimados em 4%, quando comparados à média da indústria (BONATO, 2004, p. 131-135). Assim como nos, Bonato (*Ibid.*) destaca o elevado grau de motivação da equipe de projeto.

Apesar dos vários estudos expostos, não foi encontrada na literatura uma pesquisa extensa o suficiente, que permita a generalização dos benefícios resultantes do emprego dos Métodos Ágeis. No entanto, não se pode negar que os estudos realizados até o momento, alguns dos quais aqui retratados, apontam que determinadas organizações estão de fato auferindo resultados positivos, quanto à qualidade, à produtividade e à motivação de seus profissionais, com a adoção dos Métodos Ágeis para o desenvolvimento de software.

# 2.4.6 Limitações dos Métodos Ágeis

Neste trabalho, tão importante quanto listar a aplicação e os benefícios obtidos com o uso dos Métodos Ágeis, é apresentar as limitações e obstáculos à sua utilização. Dentre os autores que escreveram sobre os Métodos Ágeis de Desenvolvimento de Software, alguns apresentaram e discutiram os principais conceitos e valores que regem este novo enfoque, como Abrahamsson *et al* (2003), Bohem e Turner (2003), Glass (2001), Cohen *et al* (2003), Cockburn e Highsmith (2001a; 2001b), Udo e Koppensteiner (2003), Fowler (2003); outros, como Turk *et al* (2003; 2005), Nerur *et al* (2005), Boger *et al* (2001) e McBreen (2003), introduziram uma visão mais crítica, apontando limitações ao emprego dos Métodos Ágeis.

Dentre as obras com enfoque crítico acima citadas, a de Turk *et al* (2005) se destaca das demais por sua ampla abordagem, voltada à identificação tanto dos pressupostos básicos sobre os quais os princípios e práticas dos Métodos Ágeis estão ancorados, quanto das limitações, decorrentes de sua aceitação, conforme mostra a Ilustração 11. Com base nestes pressupostos, identificados após análise detalhada de publicações referentes aos diferentes Métodos Ágeis e dos valores e princípios defendidos pela *Agile Alliance* e retratados no documento *Manifesto para o Desenvolvimento Ágil de Software* (BECK *et al*, 2001), Turk *et al* (2003, p. 43-46) pontuam as limitações dos Métodos Ágeis.

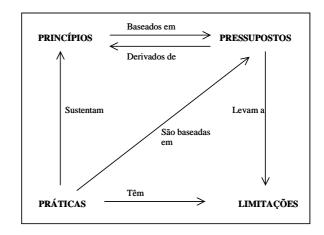


Ilustração 11 – Relacionamento entre princípios, práticas, pressupostos e limitações FONTE: TURK  $et\ al,\ 2005,\ p.\ 8.$ 

Sendo assim, para que se possa entender tais limitações, deve-se primeiramente visualizar os pressupostos identificados pelos autores (TURK *et al*, 2005, p. 22), que se encontram retratados na Tabela 16.

Tabela 16 - Sumário dos pressupostos relativos aos princípios propostos pela Agile Alliance

	Pressupostos	Detalhamento
1.	Pressuposto da visibilidade	A visibilidade do projeto só pode ser alcançada através da entrega de um software funcionando
2.	Pressuposto da iteração	Um projeto pode ser sempre estruturado em iterações curtas e de períodos fixos
3.	Pressuposto da interação com o cliente	A equipe do cliente está disponível para interação frequente, quando solicitado pelos programadores
4.	Pressuposto da comunicação da equipe	Os programadores estão localizados em local que permita a comunicação freqüente e intensiva entre si
5.	Pressuposto da comunicação face a face	A comunicação face a face é o método mais produtivo de comunicação com o cliente e entre os programadores
6.	Pressuposto da documentação	Desenvolver uma documentação extensiva e consistente (relativamente completa) e modelos de software é contraproducente
7.	Pressuposto da mudança dos requerimentos	Requisitos sempre sofrem modificações, devido a mudanças de tecnologia, necessidades dos clientes, domínios de negócio ou mesmo aquisição de novos clientes
8.	Pressuposto do custo da mudança	O custo da mudança não aumenta dramaticamente com o passar do tempo
9.	Pressuposto da experiência da equipe	Os programadores têm a experiência necessária para definir e adaptar seus processos apropriadamente
10.	Pressuposto da auto-avaliação	As equipes almejam a auto-avaliação
11.	Pressuposto da auto-organização	As melhores arquiteturas, requisitos e projetos emergem de equipes auto-organizadas
12.	Pressuposto da garantia de qualidade	A avaliação dos artefatos de software (produtos e processos) pode se restringir a entrevistas freqüentes e informais, revisões e testes de codificação

Pressupostos	Detalhamento
13. Pressuposto do desenvolvimento da aplicação específica	O reuso e a generalidade não devem ser objetivos do desenvolvimento de uma aplicação específica.
14. Pressuposto do redesenho contínuo	O software pode ser continuamente redesenhado e assim mesmo, manter sua estrutura e integridade conceitual.

FONTE: TURK et al, 2005, p. 22.

Ao abordar as limitações dos Métodos Ágeis, Turk *et al* (2005, p. 23), iniciam a discussão afirmando que "[...] nenhum processo ágil é uma bala de prata (apesar dos altos brados dos entusiastas)". Os autores defendem que, uma vez que os pressupostos apresentados acima não são atendidos por todas as organizações ou por todos os ambientes de desenvolvimento, os Métodos Ágeis em seus formatos atuais apresentam sim limitações. Para minimizá-las, a depender do contexto, determinados métodos necessitam de extensões ou incorporações de práticas e princípios, usualmente associados os enfoques preditivos e clássicos de desenvolvimento de software. As principais limitações apontadas por Turk *et al* (2005, p. 23-34) são apresentadas nos parágrafos a seguir.

A primeira limitação identificada diz respeito ao suporte a equipes distribuídas. A dispersão geográfica acarreta vários problemas que inexistem quando os integrantes da equipe de desenvolvimento e do cliente encontram-se próximos. A comunicação se torna mais difícil, há necessidade de ferramentas e tecnologia especiais. Em ambientes distribuídos, os pressupostos de interação com o cliente, comunicação da equipe, comunicação face a face ficam comprometidos. O uso de documentação formal pode se fazer necessário, para organizar o trabalho realizado em várias localidades por diferentes equipes (TURK *et al*, 2005, p. 25-26).

Turk et al (2005, p. 26-27) apontam a questão da subcontratação como outra limitação dos Métodos Ágeis, sendo que Nerur et al (2005, p. 76) estendem a discussão para a negociação de contratos de forma geral. A transferência do desenvolvimento de software a um subcontratado pressupõe o estabelecimento de um contrato de prestação de serviços bem definido. No entanto, as bases de um contrato tradicional não são as mesmas utilizadas pelos Métodos Ágeis. Mesmo seguindo um enfoque iterativo e incremental, os contratos de forma geral prevêem marcos, um número fixo de iterações, com durações e entregáveis prédefinidos e cláusulas restritivas a mudanças, o que não se adequa a vários pressupostos dos

Métodos Ágeis. Turk *et al* (2005, p. 26-27) apontam como saída, a criação de um contrato contendo uma parte fixa e outra variável para acomodar ambas as expectativas.

A dificuldade de apoio a grandes projetos e grandes equipes é outra limitação apontada por alguns autores. Vários autores apontam que quanto maior a equipe, maior a complexidade da comunicação e menos ágil se torna o processo (HIGHSMITH, 2004; COHEN *et al*, 2003, TURK *et al*, 2005, p. 27-28; NERUR *et al*, 2005, p.76). O tamanho das equipes restringe a eficiência e a freqüência das interações face a face, havendo necessidade de uma maior estruturação e organização da equipe e da definição de várias linhas e formas de comunicação. O volume de documentação requerido é maior e, de forma geral, a agilidade tende a diminuir. Isto não quer dizer que grandes projetos não possam utilizar Métodos Ágeis, mas há que se adotar práticas complementares para garantir o bom andamento dos trabalhos.

Discutindo outro item, a geração de componentes total ou parcialmente reutilizáveis (códigos de programas, documentos de análise e desenho, entre outros) pressupõe a visão completa da solução (e não apenas do software a ser desenvolvido naquele momento) e a ênfase no controle de qualidade. Os Métodos Ágeis têm por premissa a manutenção de uma documentação mínima, o que dificulta a determinação do potencial de reuso de um determinado artefato, além de ter por foco o desenvolvimento de soluções para problemas específicos e não o desenvolvimento de códigos mais genéricos e reutilizáveis. Desta forma, os Métodos Ágeis de Desenvolvimento de Software não são os mais adequados à geração de artefatos reutilizáveis (TURK *et al*, 2005, p. 28-29).

A adoção de Métodos Ágeis no desenvolvimento de sistemas de missão crítica é outro ponto comentado por Turk *et al* (2005, p. 29-30). Aplicações desta natureza requerem que a qualidade de todos os seus componentes seja intensivamente testada e que estes sejam projetados de forma a não haver falhas que impossibilitem a correção ou o uso seguro de um determinado equipamento. Neste contexto, os pressupostos relacionados à documentação, à garantia da qualidade e ao aprimoramento contínuo dos Métodos Ágeis não são mais válidos. Uma especificação formal, testes intensivos e abrangentes e outras técnicas de análise e avaliação dos métodos clássicos de desenvolvimento de software se tornam necessárias, apesar de os autores ressaltarem que a adoção de algumas práticas dos Métodos Ágeis seja interessante, para aumentar a qualidade e a confiabilidade do produto final (TURK *et al, Ibid.*).

O desenvolvimento de um software grande e complexo, com numerosas linhas de códigos (milhares ou milhões) e alto grau de integração entre seus vários componentes é outra situação em que a aplicabilidade dos Métodos Ágeis é contestada. Segundo Turk *et al* (2005, p. 30-31), projetos deste porte requerem elevado esforço de gerenciamento e controle, processos estruturados e formais, para garantir o perfeito entendimento do software e a integração de todas as suas partes. Os testes também devem ser cuidadosamente planejados. De forma geral, não é possível o desenvolvimento deste tipo de software de forma incremental e a adoção da premissa de desenvolvimento contínuo pode ser comprometida, haja vista a complexidade e a extensão do código.

Complementando a análise, Nerur et al (2005, p.76) e TURK et al (2005, p. 13-15) chamam a atenção à questão da dependência entre as organizações e suas equipes ágeis de desenvolvimento, em especial no que diz respeito à gestão do conhecimento, vital nos dias de hoje, e à relação de poder. Os Métodos Ágeis de Desenvolvimento de Software encorajam o pensamento direto e o corte em todo e qualquer esforço desnecessário, inclusive na documentação. Sendo assim, no desenvolvimento ágil, o conhecimento é tácito e reside apenas na mente de cada integrante de equipe. Em alguns casos, as organizações tornam-se dependentes desta equipe e muitas vezes há uma mudança no balanço do poder, entre os gerentes e os programadores, sendo os últimos os detentores das informações. Para minimizar este impasse, Nerur et al (op. cit.) e TURK et al (op. cit.) sugerem que seja claramente definido o que deve ser documentado e o que pode ser mantido como conhecimento tácito.

O pressuposto de interação com os clientes é outro ponto amplamente discutido e criticado por vários autores. Turk *et al* (2005, p. 12) esclarece que os Métodos Ágeis pressupõem que os clientes estão sempre disponíveis para participar, esclarecer dúvidas e tomar decisões junto à equipe de desenvolvimento, o que na prática nem sempre se mostra viável. Nerur *et al* (2005, p. 76) complementam que o ambiente pluralista de tomada de decisões (que envolve clientes e equipe de desenvolvimento), estimulado pelos Métodos Ágeis, torna o processo decisório mais lento e difícil, quando comparado ao enfoque clássico, em que o gerente do projeto é o responsável por tal. Nerur *et al* (*Ibid.*) destacam que o sucesso dos Métodos Ágeis depende de se encontrar clientes que almejem e tenham disponibilidade para uma participação ativa no processo de desenvolvimento, que sejam colaborativos, representativos e comprometidos e que disponham da autonomia e do conhecimento adequados ao projeto

Há que se mencionar ainda, a questão da manutenção de software, qualificada como algo tão problemática quanto o desenvolvimento propriamente dito e ainda pouco abordada pelos Métodos Ágeis (RUS *et al*, 2002; COHEN *et al*, 2003).

Todas estas limitações apontadas por autores e estudiosos do assunto (TURK et al, 2003 e 2005; NERUR et al, 2005; BOGER et al, 2001; McBREEN, 2003) devem ser avaliadas quando da definição método de desenvolvimento de software a ser utilizado. Contudo, apesar de sua importância, não podem ser consideradas como única verdade, uma vez que a literatura aponta exemplos de projetos conduzidos com o uso de Métodos Ágeis que foram bemsucedidos, apesar das condições teoricamente adversas à sua utilização. Entre estes exemplos podem ser citados: o desenvolvimento de um software de missão crítica com o uso do XP (GAWLAS, 2003) e a utilização do XP e do *Scrum* em projetos com mais de 100 pessoas distribuídas em mais de um continente (Fowler, 2003).

# 2.4.7 Fatores Críticos de Sucesso de Projetos de Desenvolvimento de Software Realizados com o Uso dos Métodos Ágeis

Poucas são as referências na literatura que discutem a questão dos fatores críticos de sucesso de projetos conduzidos com o uso de Métodos Ágeis. Cohen *et al* (2003, p. 31) destacam que são três os principais fatores críticos de sucesso para um projeto conduzido segundo um enfoque ágil, a saber: cultura, pessoas e comunicação.

Cockburn e Highsmith (2001b), autores que defendem uma visão dos Métodos Ágeis centrada nas pessoas, identificam a competência individual como o fator primordial para o sucesso de projetos conduzidos de acordo com os Métodos Ágeis. Mencionam que "[...] se as pessoas forem boas o suficiente, podem usar praticamente qualquer processo e atingir seus objetivos. Se as pessoas não forem boas o bastante, não há processo que possa reparar esta inadequação" (COCKBURN; HIGHSMITH, 2001b, p. 131). A falta de apoio executivo e dos principais usuários, por outro lado, é apontada como um grande empecilho para o alcance do sucesso de um projeto, levando até mesmo excelentes profissionais ao não cumprimento de seus objetivos.

Em 2003, um estudo de caráter exploratório desenvolvido por Lazaveric (classificado pelo próprio autor como "sem similar" na área até aquele momento) identificou os principais fatores críticos de sucesso de projetos de desenvolvimento de software conduzidos com o uso de Métodos Ágeis (LAZAREVIC, 2003). A pesquisa foi realizada por meio da aplicação de um questionário a profissionais que haviam gerenciado ou participado de projetos com tais características, integrantes de cinco grupos da internet especializados na troca de experiência e na discussão dos principais Métodos Ágeis. Apesar de obter uma taxa de resposta de apenas 4% (o autor acredita que este número deva ser maior, dado que muitos dos integrantes dos referidos grupos encontravam-se inativos), a pesquisa permitiu a identificação de fatores críticos de sucesso para tais projetos que se mostraram bastante alinhados às colocações de autores como Cockburn e Higsmith (2001b), Cohen *et al* (2003) e Reifer (2002).

A existência de programadores motivados foi identificada como a chave para o sucesso dos projetos pesquisados, sendo considerado o fator crítico mais importante para projetos conduzidos com o uso de Métodos Ágeis. O segundo fator crítico de sucesso identificado foi o grau de propriedade coletiva do código. A propriedade coletiva pressupõe que qualquer integrante da equipe pode alterar o código desenvolvido por outrem e contribuir para a solução e encoraja a sinergia, a colaboração e o trabalho em equipe. A descoberta deste segundo fator está totalmente alinhada ao primeiro item. Lazarevic (2003, p. 28) destaca que "[...] atingir um estágio em que os resultados do grupo sejam melhores que a contribuição de cada indivíduo é o coração de muitos Métodos Ágeis". Ainda, de acordo com Sutherland (2001), quando os indivíduos se ajudam mutuamente e contribuem para o todo, a equipe de desenvolvimento atinge o estado de hiper-produtividade. O terceiro fator encontrado diz respeito à entrega de um protótipo logo no início do projeto. Para Lazarevic (*Ibid.*), a maioria dos respondentes interpretou esta questão à luz das necessidades de mudanças de requisitos, ou seja, da necessidade de um desenvolvimento incremental do software.

Nesse mesmo estudo, Lazaveric (2003, p. 24) menciona que os projetos desenvolvidos segundo os métodos *Extreme Programming* (XP) e *Dynamic Software Development Model* (DSDM) foram mais bem-sucedidos, mas este resultado não se mostrou estatisticamente significativo.

Como informação adicional desta pesquisa, tem-se que a maioria dos respondentes possuía de um a quatro anos de experiência em Métodos Ágeis, o que confirma que a utilização destes métodos no desenvolvimento de software é um fenômeno relativamente novo. O tamanho das equipes dos projetos analisados variava entre quatro e vinte integrantes, sendo que apenas 10% dos projetos tiveram mais de 50 participantes (LAZAREVIC, 2003, p. 26).

Como muitos outros autores (REIFER, 2002; COCKBURN; HIGHSMITH, 2001b), Lazarevic (2003, p. 29) não considera seu estudo representativo, não devendo assim ser generalizado. Mas o enfoque diferenciado de seu trabalho, em especial, sua abrangência (obtenção de dados de diferentes projetos, de profissionais provenientes de distintas organizações), certamente contribui para a ampliação do conhecimento na área.

Ampliando a visão dos fatores críticos de sucesso, Chin (2004), assim como Cohn e Ford (2003), Highsmith (2004) e Nerur *et al* (2005), ressaltam que não se pode esquecer o inegável valor do gerenciamento de projetos na entrega de um projeto de desenvolvimento de software bem-sucedido. Apesar de alguns Métodos Ágeis já possuírem componentes de gerenciamento de projetos inseridos em suas práticas, como por exemplo o *Scrum*, o FDD e o ASD, Thomsett (2002, p. 17) indica a necessidade de uma abordagem mais ampla e, preferencialmente, em separado das questões técnicas.

Analisando os fatores críticos de sucesso identificados, percebe-se que estão estritamente relacionados à valorização da competência individual e à comunicação (REIFER, 2002; COCKBURN; HIGHSMITH, 2001b; LAZAREVIC, 2003) e à entrega rápida de valor (LAZAREVIC, 2003). Todos estes itens correspondem a características marcantes dos Métodos Ágeis de Desenvolvimento de Software e o também do Gerenciamento Ágil de Projetos, o que pode sugerir que talvez este seja o enfoque de gerenciamento de projetos mais mais apropriado para o desenvolvimento de software conduzido com o uso de um Método Ágil.

# 2.5 Gerenciamento Ágil de Projetos

Este tópico apresenta o histórico, a definição, os valores, os princípios e a estrutura de práticas do *Gerenciamento Ágil de Projetos*.

# 2.5.1 Evolução do Conceito dos Métodos Ágeis para o Gerenciamento de Projetos

O Gerenciamento Ágil de Projetos ou Agile Project Management (APM) foi criado a partir dos valores e princípios dos Métodos Ágeis de Desenvolvimento de Software, retratados no Manifesto para o Desenvolvimento Ágil de Software (BECK et al, 2001). Conectado a iniciativas correlatas das indústrias de construção civil e aeroespacial (uma vez que o cenário de complexidades e incertezas era comum a todos), o movimento para o desenvolvimento ágil de software ampliou sua abrangência, sendo estendido ao gerenciamento de projetos (HIGHSMITH, 2004, p. xix).

Chin (2004, p.1) afirma que "[...] em projetos que envolvem tecnologia, balancear os requisitos do gerenciamento clássico de projetos e as necessidades de uma equipe altamemente capacitada e criativa é mais uma arte que uma ciência". Explica que, se por um lado o excesso de formalidade e de processos tende a inibir a equipe e bloquear a inovação, por outro, a falta de estrutura pode fazer com que os objetivos do projeto nunca sejam atingidos.

Por muitos anos, o gerenciamento de projetos se desenvolveu sobre uma sólida plataforma, capaz de dar suporte a diferentes tipos de projeto, nos mais variados ambientes. As organizações adotaram o Gerenciamento Clássico de Projetos, propuseram e implementaram melhorias e adaptações, criando seus próprios modelos e expandindo a plataforma clássica de gerenciamento de projetos (CHIN, 2004, p. 1-3). No entanto, Chin (*Ibid.*) explica que, como qualquer plataforma, o Gerenciamento Clássico de Projetos apresenta restrições e ao se aproximar do limiar de sua abrangência, estas restrições se tornam mais visíveis e o seu desempenho, menos efetivo. Continuar o desenvolvimento desta plataforma clássica, segundo o autor, em alguns momentos é mais difícil do que estruturar uma nova idéia.

Neste contexto, Chin (2004, p. 1-3) propõe a criação de um novo modelo e não uma simples expansão do Gerenciamento Clássico de Projetos. O Gerenciamento Ágil de Projetos surge, então, como uma nova plataforma de gerenciamento de projetos (ver Ilustração 12), aplicável a ambientes voláteis e desafiadores, sujeitos a mudanças freqüentes, em que o processo prescritivo e padronizado não mais se adequa (CHIN, 2004, p. 1-3; HIGHSMITH, 2004, p. 5).

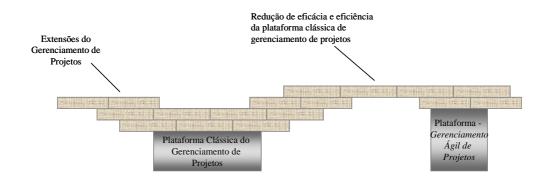


Ilustração 12 – Relacionamento entre as plataformas de gerenciamento clássico e ágil de projetos FONTE: CHIN, 2004, p.3

Segundo Highsmith (2004) e Chin (2004), o Gerenciamento Ágil de Projetos se desfaz da postura antecipatória, fortemente calcada no planejamento prévio de ações e atividades, típica do gerenciamento clássico de projetos, e busca o desenvolvimento da visão do futuro e da capacidade de exploração.

Enfocando este último ponto, Highsmith (2004, p. 6) afirma que são cinco os objetivos principais para um bom processo de exploração, que acabam por constituir a base para o Gerenciamento Ágil de Projetos:

- 1. *Inovação contínua*: entrega de produtos que atendam aos requisitos dos clientes e agreguem valor ao negócio. A idéia de inovação é associada a um ambiente organizacional cuja cultura estimule o autogerenciamento e a autodisciplina;
- 2. Adaptabilidade do produto: os produtos desenvolvidos devem não só oferecer valor ao cliente no presente, mas ser também adaptáveis às novas necessidades do futuro;
- 3. Tempos de entrega reduzidos: foco, direcionamento preciso e capacidade técnica da equipe são fundamentais para a redução dos prazos de desenvolvimento de produtos visando ao melhor aproveitamento das oportunidades de mercado e um melhor retorno sobre o investimento;

- 4. Capacidade de adaptação do processo e das pessoas: formação de equipes cujos integrantes estejam confortáveis com mudanças e que não as enxerguem como obstáculos e sim como desafios de um ambiente dinâmico; estabelecimento de processos que respondam rapidamente às alterações do negócio;
- 5. *Resultados confiáveis*: entrega de produtos de valor ao cliente, garantindo a operação, o crescimento e aumento da lucratividade da empresa.

O Gerenciamento Ágil de Projetos pode ser visto como uma resposta de âmbito gerencial às crescentes pressões por constantes inovações, à concorrência acirrada, à necessidade de redução dos ciclos de desenvolvimento e implantação de novos produtos ou serviços e à necessidade de adaptação a um ambiente de negócio bastante dinâmico (HIGHSMITH, *Ibid.*).

# 2.5.2 Definição, Valores e Princípios do Gerenciamento Ágil de Projetos

Highsmith (2004, p.16) define o Gerenciamento Ágil de Projetos como "[...] um conjunto de valores, princípios e práticas que auxiliam a equipe de projeto a entregar produtos ou serviços de valor em um ambiente desafiador". Os valores e os princípios descrevem o porquê do Gerenciamento Ágil de Projetos e as práticas descrevem como realizá-lo. Apesar de ratificar a necessidade de entrega produtos confiáveis aos clientes dentro de restrições de prazo e custos, comum ao Gerenciamento Clássico de Projetos, Highsmith (2004, p. 6) menciona que o Gerenciamento Ágil de Projetos pode ser considerado "mais uma atitude que um processo, mais ambiente que metodologia".

Os valores principais deste novo enfoque de gerenciamento de projetos endereçam tanto a necessidade de criação e entrega de produtos ágeis, adaptáveis e de valor, como a necessidade de desenvolvimento de equipes de projeto com as mesmas características (HIGHSMITH, 2004, p.10). Os quatro valores centrais do Gerenciamento Ágil de Projetos são:

- 1. As respostas às mudanças são mais importantes que o seguimento de um plano;
- 2. A entrega de produtos está acima da entrega de documentação;
- 3. Priorização da colaboração do cliente sobre a negociação de contratos;
- 4. Os indivíduos e suas interações são mais importantes que os processos e as ferramentas.

Com relação ao primeiro valor, Highsmith (2004, p. 10-11) afirma que os projetos exploratórios, alvo do Gerenciamento Ágil de Projetos, são caracterizados por processos que enfatizam a visão do futuro e a exploração, ao invés do planejamento detalhado e a respectiva execução. Highsmith (*Ibid.*) e Chin (2004, p. 1-3) mencionam que não se trata apenas de absorver pequenas alterações de escopo, prazo ou custo, mas sim, de dar abertura à mudança completa de planos, requisitos, tecnologia e arquitetura no decorrer do projeto. Highsmith (*op. cit.*) embasa seu ponto de vista, ao apresentar o caso de uma companhia que, erroneamente, se recusou a mudar o plano traçado inicialmente para um projeto de desenvolvimento de software, orçado em aproximadamente US\$125 milhões, o que a levou a um grande e custoso desastre. Para Highsmith (*Ibid.*), "[...] planejar o trabalho e executar o plano" não é um lema adequado a uma grande variedade de projetos, em especial para os projetos de desenvolvimento de novos produtos (ou software) ou àqueles relacionados a qualquer tipo de inovação tecnológica. Esta posição está alinhada às proposições de Thomsett (2002) e Chin (2004).

Abordando o segundo valor, ao estabelecer a prioridade da entrega de produtos sobre a preparação de uma documentação exaustiva, Highsmith (2004, p. 11-12) argumenta que não se trata de menosprezar a importância da documentação, mas sim de assumir que os resultados só podem ser avaliados pela equipe de projeto e pelo cliente quando algo concreto é apresentado, ou seja, quando se tem um produto (ou software) operante. Highsmith (*Ibid.*) defende a manutenção de um patamar mínimo e suficiente de documentação para auxiliar o processo de comunicação e colaboração, propiciar a transferência de conhecimento, preservar a informação histórica, servir de base para a melhoria de produtos e processos e, algumas vezes, atender aos requisitos legais.

O terceiro valor, por sua vez, tem por pressuposto o estabelecimento de uma parceria entre o cliente e a equipe de projeto, na qual cada um tem papéis e responsabilidades específicos e bem definidos, sendo cobrados por isso. Esta parceria deve ser marcada por uma forte colaboração e não por disputas contratuais (HIGHSMITH, 2004, p. 13). Apesar de reconhecer a existência de diferentes partes interessadas no projeto, Highsmith (*Ibid.*) menciona que o cliente deve "reinar soberano" e apresenta a seguinte definição de cliente: "[...] um indivíduo ou grupo de indivíduos que usa os produtos ou serviços para gerar valor ao negócio".

Considerando o quarto valor, Highsmith (2004, p. 13-14) afirma que os processos servem como guias ou trilhas e as ferramentas são utilizadas para melhorar a eficiência, mas sem pessoas com qualificações técnicas e comportamentais adequadas, nenhum processo ou ferramenta produz resultado algum. O movimento *ágil* deposita elevado valor no indivíduo e reconhece sua capacidade de auto-organização, autodisciplina e sua competência.

Enfocando o sucesso na ótica do Gerenciamento Ágil de Projetos, Highsmith (2004, p. 27) menciona que o sucesso é direcionado pelas pessoas e suas interações e não por processos e estruturas. As competências e habilidades individuais, as interações entre pessoas ou equipes tecnicamente qualificadas e a capacidade da equipe aprender e aplicar os conhecimentos adquiridos são determinantes para o sucesso ou o fracasso de um projeto, estando estritamente ligados ao atendimento das expectativas do cliente. Como as pessoas são normalmente guiadas por um conjunto interno de valores, desenvolver a agilidade depende do perfeito alinhamento entre o ambiente e este sistema de valor de cada indivíduo. Esta é a razão pela qual o Gerenciamento Ágil de Projetos é fortemente calcado em valores e também o motivo por que muitas vezes é impossível a aplicação do Gerenciamento Ágil de Projetos em determinadas organizações ou equipes (tal como ocorre com a adoção dos Métodos Ágeis de Desenvolvimento de Software discutida no tópico 2.4.4).

Além dos valores, o Gerenciamento Ágil de Projetos possui um conjunto de princípios mestres, que norteiam sua aplicação. De acordo com Larson e LaFasto (1989), a liderança baseada em princípios é uma das características mais importantes das equipes de alto desempenho. Highsmith (2004, p. 27-28) estabelece seis princípios, divididos em duas categorias (ver Tabela 17), que auxiliam as equipes de projeto a determinar como proceder, ou seja, ajudam-nas a definir quais práticas são mais apropriadas, a gerar outras práticas quando necessário, ou mesmo, a avaliar algumas que venham a surgir e implementá-las com agilidade. Embora cada princípio seja útil se aplicado isoladamente, o sistema formado por eles cria um ambiente que encoraja e produz resultados mais efetivos (HIGHSMITH, *Ibid.*).

Tabela 17 - Princípios do gerenciamento ágil de projetos

Categoria	Princípio	
Valor ao cliente	Entregar valor ao cliente	
	Gerar entregas iterativas e baseadas em funcionalidades	
	3. Buscar a excelência técnica	
Estilo de gerenciamento	4. Encorajar a exploração	
baseado na liderança e colaboração	5. Construir equipes adaptativas (auto- organizadas e autodisciplinadas)	
Colabolação	6. Simplificar	

FONTE: ADAPTADO DE HIGHSMITH, 2004, p. 27.

## 2.5.3 Ciclo de Vida e Fases do Gerenciamento Ágil de Projetos

Apesar dos processos não serem tão importantes quanto as pessoas no Gerenciamento Ágil de Projetos, isto não significa que não se dê importância a eles. Highsmith (2004, p. 79) e Chin (2004, p. 14) defendem que não se deve atribuir aos processos uma conotação negativa, vinculada ao excesso de documentação e padronização, à característica estática e prescritiva, difícil de ser mudada, conforme alardeiam alguns seguidores do movimento ágil. Segundo os autores, os processos, como qualquer outro elemento de uma organização, devem estar estritamente alinhados aos objetivos de negócio. Em se tratando de operações contínuas e repetitivas, processos prescritivos são plenamente justificáveis. Por outro lado, se o ambiente for dinâmico e volátil, a estrutura de processos deve ser orgânica, flexível e facilmente adaptável (HIGHSMITH, op. cit..; CHIN, op. cit.).

Sendo assim, a estrutura de processos para o Gerenciamento Ágil de Projetos deve ter por foco o conceito de agilidade e englobar os princípios apresentados no tópico anterior, assim como assegurar o alcance dos objetivos de negócio. Para tanto, deve:

- Favorecer a exploração e a cultura adaptativa;
- Permitir a auto-organização e a autodisciplina;
- Promover a confiabilidade e a consistência possíveis, dados o grau de incertezas e a complexidade inerentes ao projeto;
- Ser flexível e facilmente adaptável;
- Permitir visibilidade ao longo do processo;
- Incorporar o aprendizado;
- Englobar as práticas específicas de cada fase;

Prover pontos de verificação.

Segundo Udo e Koppensteiner (2003, p. 5) e Highsmith (2004, p. 81), um projeto típico do Gerenciamento Ágil de Projetos possui uma etapa inicial, seguida por vários ciclos ou iterações. A cada ciclo é feito um novo planejamento de escopo, prazo, custo e qualidade, visando à entrega de produtos ou resultados e possibilitando incrementos de funcionalidades conforme a necessidade do negócio. Ao final das várias iterações tem-se o término do projeto. Este padrão ou fluxo típico dos projetos *ágeis*, bem como o nível de atividade, são apresentados na Ilustração 13. Esta estruturação favorece a entrega de valor e a geração de um ambiente que propicie o aprendizado contínuo (UDO; KOPPENSTEINER, *op. cit.*).

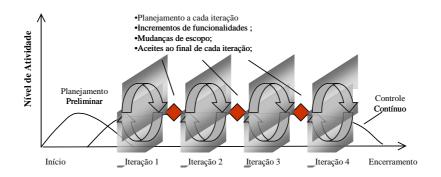


Ilustração 13 – Fluxo do gerenciamento ágil de projetos FONTE: ADAPTADO DE UDO; KOPPENSTEINER, 2003, p. 5

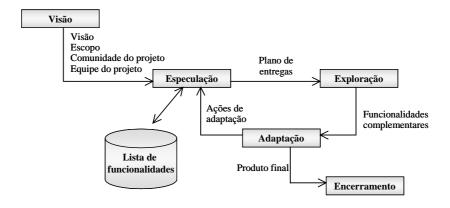
O Gerenciamento Ágil de Projetos está estruturado em cinco fases, assim descritas (HIGHSMITH, 2004, p. 81),:

- Visão: compreende a determinação da visão do produto e do escopo do projeto, a identificação da comunidade do projeto (clientes, gerente de projeto, equipe de projeto e outras partes interessadas) e a definição de como a equipe de projeto trabalhará em conjunto;
- 2. *Especulação:* engloba a identificação dos requisitos iniciais para o produto, a definição das atividades, o desenvolvimento de um plano de projeto (contendo entregas, marcos, várias iterações, cronograma de trabalho e alocação de recursos), a incorporação de estratégias para mitigação de riscos, as estimativas de custos e outras informações

financeiras relevantes. Nesta etapa é elaborado um planejamento preliminar, seguido por planejamentos específicos a cada iteração;

- 3. *Exploração*: corresponde à entrega dos produtos planejados, através do gerenciamento da carga de trabalho e do emprego de práticas e estratégias de mitigação de risco apropriadas. Estas entregas são feitas sob a forma de incrementos de funcionalidades do produto e são divididas entre os ciclos do projeto;
- 4. Adaptação: compreende a revisão dos resultados entregues, a análise da situação atual e a avaliação do desempenho da equipe e do projeto, para eventual adaptação. Esta revisão objetiva não apenas uma comparação do "realizado *versus* planejado", mas principalmente do "realizado *versus* informações e requisitos atualizados do projeto", para a identificação das mudanças necessárias;
- 5. *Encerramento*: referente à finalização das atividades do projeto, à transferência dos aprendizados-chave e à celebração.

O relacionamento entre estas fases pode ser visualizado na Ilustração 14.



**Ilustração 14 – Fases do gerenciamento ágil de projetos** FONTE: ADAPTADO DE HIGHSMITH, 2004, p. 81.

Após a fase *Visão*, as fases *Especulação*, *Exploração* e *Adaptação* se alternam a cada iteração, no intuito de refinar o produto do projeto. A etapa de *Especulação* é retomada com objetivo de planejar o novo ciclo, levando em consideração os resultados até então alcançados no projeto e os incrementos ou alterações de escopo solicitados até o momento. Algumas vezes, o retorno à fase Visão pode ser necessário, em especial quando se têm modificações

muito significativas no escopo projeto. Uma vez obtido o produto final, segue-se o *Encerramento* do projeto (HIGHSMITH, 2004, p. 85).

# 2.5.4 Práticas do Gerenciamento Ágil de Projetos

Assim como o PMI (2004) sugere uma série de processos de gerenciamento de projetos, para cada fase do Gerenciamento Ágil de Projetos existe um conjunto de práticas, que em sua totalidade, formam um sistema de práticas. Este sistema não só alinha os valores e os princípios do Gerenciamento Ágil de Projetos, mas permite também a sua implementação (HIGHSMITH, 2004, p. 86). O autor, entretanto, é avesso à idéia da adoção de "melhores práticas", ao afirmar que as práticas são apenas maneiras de se executar algo e que podem ser boas ou ruins, a depender do contexto em que estão inseridas. Highsmith (*Ibid.*) menciona que as práticas do Gerenciamento Ágil de Projetos se mostram úteis em uma grande diversidade de situações e que também podem ser aplicadas no Gerenciamento Clássico de Projetos. Por outro lado, admite que outras práticas, não mencionadas no modelo em questão, podem trazer benefícios aos projetos ágeis.

#### 2.5.4.1 Práticas da Fase Visão

O propósito da fase *Visão* é identificar de maneira clara "o que" precisa ser feito e "como" o trabalho será realizado. Para tanto, suas práticas estão estruturadas em quatro categorias: *visão do produto*, *escopo do projeto*, *comunidade do projeto* e *abordagem* (HIGHSMITH, 2004, p. 88-92), que são apresentadas na Tabela 18.

Tabela 18 - Práticas da fase "Visão" do gerenciamento ágil de projetos

Categoria	Prática	Descrição
Visão do produto	Visão do produto e "declaração de elevador"  Desenvolvimento de um conceito de alto nível do pro projeto, de uma forma concisa e visual, com a utilizaç um texto simples, garantindo a uniformidade de enter e de discurso entre todos os envolvidos no projeto	
	Arquitetura do produto	Desenvolvimento de uma arquitetura técnica do produto, que facilite a exploração e assegure um direcionamento à condução dos trabalhos e à organização da equipe de projeto, visando ao atendimento das expectativas dos clientes
Escopo do projeto (objetivos e restrições)	Folha de dados do projeto	Elaboração de um documento que sumarize os pontos-chave do projeto em termos de escopo, prazo, custos, recursos, qualidade e riscos

Categoria	Prática	Descrição
Comunidade do projeto	Escolha das pessoas certas	Montagem da equipe do projeto e definição de sua organização, buscando sempre atrair e selecionar os melhores talentos
	Identificação dos participantes	Identificação de todas as partes interessadas no projeto, de forma a melhor entender e gerenciar suas expectativas
	Interface entre a equipe do cliente e a equipe do projeto	Estabelecimento do processo de colaboração entre a equipe do projeto e a equipe do cliente
Abordagem	Adaptação de processos e práticas	Definição de uma estrutura de processos e práticas com base em uma estratégia de auto-organização, que estabelece a forma de trabalho em conjunto da equipe do projeto

FONTE: ADAPTADO DE HIGHSMITH, 2004, p. 92-124.

#### 2.5.4.2 Práticas da Fase Especulação

Segundo Highsmith (2004, p. 128-131), a fase *Especulação* tem por objetivo a elaboração de um plano de entregas que represente o melhor entendimento da equipe de projeto, em um dado momento do projeto. A utilização da palavra especulação no lugar de planejamento está estritamente relacionada ao reconhecimento do elevado grau de incertezas e de mudanças que caracteriza um projeto-alvo do Gerenciamento Ágil de Projetos. O plano resultante desta etapa é construído a partir de informações da especificação e da arquitetura do produto, dos riscos envolvidos, do padrão de qualidade ou do nível de defeito estabelecido, das restrições do negócio e dos marcos do projeto.

As práticas de fase Especulação, repetidas a cada ciclo ou iteração, são divididas em duas categorias: *estrutura analítica do produto* e *planejamento de entregas* (HIGHSMITH, 2004, p. 132-164), sendo retratadas na Tabela 19.

Tabela 19 - Práticas da fase "Especulação" do gerenciamento ágil de projetos

Tabela 17-11aucas da fase Especulação do gerenciamento agil de projetos		
Categoria	Prática	Descrição
Estrutura analítica do produto	Lista de funcionalidades do produto, através da definição de seu requisitos, gerando uma lista de funcionalidades do produto	
	Cartões de funcionalidades	Criação de um documento padrão para o registro das funcionalidades e requisitos do produto e das estimativas de prazo para seu desenvolvimento
	Cartões de requisitos de desempenho	Documentação dos requisitos de desempenho e operação do produto a ser entregue
Planejamento das entregas	Plano de entregas, marcos e iterações	Desenvolvimento de um plano que apresente o caminho a ser seguido pela equipe de projeto para entregar o produto do projeto, de acordo com os objetivos de escopo, prazo e restrições delineadas na Folha de Dados do Projeto. São planejados vários ciclos ou iterações para refinamento do produto do produto final do projeto

FONTE: ADAPTADO DE HIGHSMITH, 2004, p. 131-164.

### 2.5.4.3 Práticas da Fase Exploração

A fase *Exploração* visa à entrega dos produtos planejados, a cada ciclo do projeto, através do gerenciamento da carga de trabalho e do emprego de práticas e estratégias de mitigação de risco apropriadas (HIGHSMITH, 2004, p. 166-168). De acordo com Highsmith (*Ibid.*) e Chin (2004, p. 69-81), a essência do papel do gerente de projetos no Gerenciamento Ágil de Projetos não é a elaboração de um cronograma detalhado e a preparação dos relatórios de progresso (apesar de ambos fazerem parte de seu trabalho), mas sim a criação e o desenvolvimento de uma equipe de alto desempenho a partir de um grupo de pessoas e estabelecer um estilo de liderança e gerenciamento que encoraje a inovação e incentive a aceitação de situações de risco e de mudanças constantes.

Neste contexto, Highsmith (2004, p. 169-209) propõe um conjunto de práticas divididas em três categorias: *entrega segundo os objetivos*, *práticas técnicas* e *comunidade do projeto*. Estas práticas podem ser visualizadas na Tabela 20.

Tabela 20 - Práticas da fase "Exploração" do gerenciamento ágil de projetos

Tubelu 20 Trusteus du ruse Emploração do gerenetamiento agri de projetos			
Categoria	Prática	Descrição	
Entrega segundo os objetivos	Gerenciamento da carga de trabalho	Atribuição da responsabilidade pelo gerenciamento das atividades do dia-a-dia, necessárias para a entrega das funcionalidades a cada iteração, aos próprios integrantes da equipe de projeto	
Práticas técnicas	Mudanças de baixo custo	Redução dos custos das mudanças ao longo das diferentes fases do ciclo de vida do produto, alinhada à necessidade de entrega de valor ao cliente "hoje", mas com vistas ao futuro	
Comunidade do projeto	Aconselhamento e desenvolvimento da equipe	Desenvolvimento contínuo das competências, habilidades e domínio técnico e de negócio dos integrantes da equipe de projeto, enfatizando a autodisciplina e o espírito de equipe	
	Reuniões diárias de integração da equipe de projeto	Realização de reuniões de integração com a equipe de projeto visando a coordenação diária das atividades do projeto	
	Decisões participativas	Fornecimento à comunidade do projeto de práticas específicas para entender, analisar e/ou tomar decisões ao longo do projeto	
	Interações diárias com o cliente	Realização de reuniões ou interações diárias com o cliente para garantir que os esforços do projeto sejam executados de forma a atender às suas expectativas	

FONTE: ADAPTADO DE HIGHSMITH, 2004, p. 169-209.

#### 2.5.4.4 Prática da Fase Adaptação

Se no cenário do Gerenciamento Ágil de Projetos os planos são especulações acerca do futuro, então é grande a necessidade de *feedbacks* freqüentes e efetivos para a validação das

hipóteses assumidas (HIGHSMITH, 2004, p. 213-215). As adaptações dependem do manuseio e entendimento de uma vasta gama de informações, que incluem o progresso do projeto, os riscos técnicos, a evolução dos requisitos do produto e a análise do mercado. A equipe de projeto deve avaliar e tomar decisões constantes sobre quatro pontos principais:

- Funcionalidades primárias, segundo a perspectiva da equipe do cliente;
- Qualidade do produto, de acordo com as perspectivas da equipe técnica;
- Desempenho da equipe do projeto;
- Progresso do projeto.

A fase Adaptação possui apenas uma prática – *revisão do produto / projeto / equipe e ações de adaptação* – divida em várias subpráticas, como mostra a Tabela 21 (HIGHSMITH, 2004, p. 211-231).

Tabela 21 - Práticas da fase "Adaptação" do gerenciamento ágil de projetos

1 abela 21	i - I i aticas ua iase. Auaptação (	do gerenciamento agn de projetos
Prática	Descrição	Sub-prática Sub-prática
Revisão do produto /	O objetivo desta prática é garantir	- Grupo de foco com cliente
projeto / equipe e ações de	que o feedback constante e o	- Revisões técnicas
adaptação	aprendizado de alto nível ocorram em todas as dimensões do projeto.	- Avaliação do desempenho da equipe do projeto
	em todas as differences do projeto.	- Relatório de progresso do projeto
		- Acompanhamento do escopo e do valor do trabalho realizado <sup>24</sup> a cada iteração
		- Acompanhamento do cronograma de cada iteração
		- Acompanhamento dos custos e da utilização dos recursos (custo estimado ao término do projeto)
		- Acompanhamento da qualidade do produto do projeto
		- Informação à equipe do projeto
		- Ações de adaptação

FONTE: ADAPTADO DE HIGHSMITH, 2004, p. 211-231.

\_

<sup>&</sup>lt;sup>24</sup> Highsmith (2004, p. 227) sugere a utilização do EVA – *Earnead Value Analysis* (PMI, 2004, p. 173) – para a determinação do valor do trabalho realizado.

#### 2.5.4.5 Prática da Fase Encerramento

O *Encerramento* do projeto é tanto uma fase como uma prática. Como recursos humanos qualificados são normalmente escassos, há uma forte tendência por parte das organizações de transferi-los rapidamente para outra iniciativa, quando do término de um projeto, sem estes profissionais recebam, ao menos, os créditos pela realização. Entretanto, há várias atividades a serem realizadas na etapa final de um projeto. A primeira delas, é a *celebração*, que tem dois propósitos fundamentais: reconhecer aqueles que trabalharam muito para que o projeto fosse entregue e marcar o encerramento do projeto. A segunda, menos glamurosa, diz respeito à limpeza de eventuais itens em aberto, à entrega da documentação final e à preparação dos relatórios financeiros e administrativos requeridos. Por fim, tem-se a condução de uma retrospectiva do projeto, visando à obtenção e à divulgação das lições aprendidas entre as equipes de projeto (HIGHSMITH, 2004, p. 231-232).

O encerramento de um projeto, seja ele gerenciado segundo a abordagem ágil ou clássica, é fundamental e marca a transição do produto do projeto para a operação da organização (HIGHSMITH, *Ibid.*).

# 2.5.5 Aplicações, Resultados e Limitações do Gerenciamento Ágil de Projetos

Diferentemente dos Métodos Ágeis de Desenvolvimento de Software, até o final desta dissertação, não foram encontrados estudos analisando a eficácia, ou mesmo fornecendo indícios de resultados positivos (ou negativos) da aplicação do Gerenciamento Ágil de Projetos. Uma explicação para tal situação pode ser atribuída ao fato de o assunto ser bastante recente. O que se têm são indicações de alguns autores, como Thomsett (2002), Highsmith (2004) e Chin (2004), acerca dos potenciais de aplicação deste novo enfoque de gerenciamento de projetos.

Highsmith (*op. cit.*) e Chin (*op. cit.*) sugerem a aplicação do Gerenciamento Ágil de Projetos em projetos de desenvolvimento de novos produtos, ou outros que requeiram alto grau de inovação, criatividade, que envolvam tecnologia de ponta, equipes de alto desempenho, que estejam sujeitos a incertezas ou inseridos em ambientes de negócio dinâmicos. Highsmith (2004, p. 4) indica formalmente o uso do Gerenciamento Ágil de Projetos para o desenvolvimento de software.

Chin (2004, p. 13-21) propõe um modelo, apresentado na Tabela 22, para a verificação da aplicabilidade do Gerenciamento Ágil de Projetos, levando-se em consideração duas dimensões-chave, a saber:

- Tipo de projeto: projetos operacionais (projetos simples e que ocorrem regularmente na organização); projetos de desenvolvimento de novos produtos / processos; ou projetos de desenvolvimento de novas tecnologias / plataformas;
- Organizações envolvidas: apenas uma organização interessada no projeto; várias organizações internas interessadas no projeto; ou várias organizações externas interessadas no projeto.

Tabela 22 - Aplicabilidade do gerenciamento ágil de projetos

Tabela 22 - Apricabilidade do gerenciamento agri de projetos			
	Várias organizações interessadas; externas	Várias organizações interessadas; internas	Uma única organização interessada
Projetos operacionais	Gerenciamento Clássico de Projetos	Gerenciamento <b>Clássico</b> de Projetos	Gerenciamento <b>Clássico</b> de Projetos
Projetos de desenvolvimento de novos produtos / processos	Gerenciamento Clássico de Projetos / Gerenciamento Ágil de Projetos	Gerenciamento Clássico de Projetos / Gerenciamento Ágil de Projetos	Gerenciamento <b>Ágil</b> de Projetos
Projetos de desenvolvimento de novas tecnologias / plataformas	Gerenciamento Clássico de Projetos / Gerenciamento Ágil de Projetos	Gerenciamento <b>Ágil</b> de Projetos	Gerenciamento <b>Ágil</b> de Projetos

FONTE: CHIN, 2004, p. 20.

Nas áreas onde há a possibilidade de escolha, Chin (*Ibid*.) sugere que seja feita uma análise do contexto interno e externo, considerando aspectos culturais e a prontidão da organização e dos principais interessados no projeto para a adoção de um enfoque ágil, antes da definição da melhor abordagem. Esta colocação de Chin (*Ibid*.) é corroborada por Highsmith (2004) e remete à discussão apresentada no tópico 2.4.4 – Aplicação dos Métodos Ágeis nas Organizações – ou seja, às ponderações de Ambler (2002c), Cohen *et al*, 2003; Fowler (2003) e Nerur *et al* (2005). Isto porque tanto os Métodos Ágeis como o Gerenciamento Ágil de Projetos foram construídos sobre bases, valores e princípios similares e ambos pressupõem que as organizações estejam preparadas do ponto de vista cultural, organizacional e gerencial para a sua adoção.

Apesar de não haver na literatura menção específica às limitações do Gerenciamento Ágil de Projetos, as ressalvas feitas por Turk *et al* (2005) e Nerur *et al* (2005) com relação aos Métodos Ágeis, em especial quanto à participação dos clientes, à questão da negociação de contratos, ao suporte a equipes de projeto distribuídas e à dependência da organização perante os integrantes da equipe de projeto, podem ser transpostas para o Gerenciamento de Projetos, dados os valores e princípios comuns.

Cabe fazer uma ressalva, porém, quanto ao tamanho do projeto. Highsmith (2004, p. 85) afirma que os valores e os princípios do Gerenciamento Ágil de Projetos são aplicáveis a projetos de qualquer envergadura e, similarmente, as práticas podem ser empregadas em projetos de diversos tamanhos, havendo somente a necessidade de adaptação para equipes com mais de 50 integrantes.

# 2.5.6 Comparação: Gerenciamento Clássico e Gerenciamento Ágil de Projetos

Para finalizar a explanação sobre o Gerenciamento Ágil de Projetos é interessante o estabelecimento de uma comparação entre ele e o Gerenciamento Clássico de Projetos. Um possível alinhamento entre os enfoques clássico e ágil de gerenciamento de projetos pode ser feito através da comparação entre as principais características dos grupos de processos propostos pelo PMI (2004) – Iniciação, Planejamento, Execução, Monitoramento e Controle e Encerramento – e as fases do Gerenciamento Ágil de Projetos – Visão, Especulação, Exploração, Adaptação e Encerramento (HIGHSMITH, 2004). Esta comparação tem por base o trabalho de Udo e Koppensteiner (2003, p. 3), sendo seu resultado apresentado na Tabela 23.

Tabela 23 - Alinhamento entre os enfoques ágil e clássico de gerenciamento de projetos

Grupo de Processos do Gerenciamento Clássico de Processos	Fases do Gerenciamento Ágil de Projetos
<b>Iniciação</b> : Autorização de um novo projeto ou fase e definição do escopo preliminar do projeto	<b>Visão</b> : Determinação da visão do produto e do escopo inicial do projeto
<b>Planejamento</b> : Planejamento integral e detalhado do projeto	<b>Especulação</b> : Desenvolvimento de um plano inicial do projeto, seguido por planejamentos individuais a cada iteração
Execução: Execução do plano do projeto	<b>Exploração</b> : Entrega das funcionalidades / produtos previstos a cada ciclo
Monitoramento e Controle: Ênfase no controle do progresso dos trabalhos e no controle e gerenciamento de mudanças para minimizar os impactos no projeto	Adaptação: Revisão dos resultados entregues e abertura para adaptações do escopo, visando o atendimento aos novos requisitos do negócio

Grupo de Processos do Gerenciamento Clássico de Processos	Fases do Gerenciamento Ágil de Projetos
<b>Encerramento:</b> Formalização do aceite final do projeto	<b>Encerramento</b> : Aceites do cliente a cada ciclo ou iteração e formalização do encerramento do projeto ao final dos trabalhos

FONTE: ADAPTADO DE KOPPENSTEINER e UDO, 2003, p. 3.

A partir das informações da tabela acima e do exposto ao longo deste trabalho, é possível verificar que as diferenças fundamentais entre o Gerenciamento Clássico de Projetos e o Gerenciamento Ágil de Projetos residem fundamentalmente em dois pontos: na relação "Planejamento x Especulação" e na dualidade "Monitoramento e controle x Adaptação" (UDO; KOPPENSTEINER, 2003; HIGHSMITH, 2004; CHIN, 2004).

O Gerenciamento Clássico de Projetos, estruturado segundo uma visão de processos, conforme descrito pelo PMI (2004) e, anteriormente, defendido por autores como Verzuh (1999), Kerzner (2002), Maximiano (2002), Dinsmore e Neto (2004), entre outros, deposita uma grande importância no planejamento detalhado do projeto e nos processos formais de monitoramento e controle. Por outro lado, no Gerenciamento Ágil de Projetos, a ênfase é transferida do planejamento para a execução, visando à entrega rápida de valor ao cliente e à apresentação de resultados ao longo de todo o projeto; e do controle para adaptação, permitindo alterações substanciais de escopo a cada iteração, para atender às mudanças de requisitos do negócio (THOMSETT, 2002; UDO; KOPPENSTEINER, 2003; CHIN, 2004; HIGHSMITH, 2004).

Com relação às áreas de conhecimento, também é possível traçar um paralelo entre Gerenciamento Clássico de Projetos e o Gerenciamento Ágil de Projetos, em uma adaptação do trabalho de Udo e Koppensteiner (2003, p. 6). A avaliação, apresentada na Tabela 24, aponta que praticamente todas as áreas de conhecimento são abordadas pelo Gerenciamento Ágil de Projetos, porém com um enfoque diferenciado, dada sua ênfase na entrega de valor ao cliente, na resposta rápida às necessidades de mudanças e na valorização do indivíduo.

Tabela 24 - Comparação dos enfoques clássico e ágil de gerenciamento de projetos por área de conhecimento

Áreas de Gerenciamento Clássico de Projetos Garanciamento Ágil da Projetos		
Conhecimento	Gerenciamento Ciassico de Frojetos	Gerenciamento Ágil de Projetos
Gerenciamento da Integração do projeto	Assegura a coordenação dos vários elementos do projeto	A necessidade de coordenação formal é limitada devido à redução a nível mínimo de estrutura e de processos
Gerenciamento do escopo do projeto	Assegura que o projeto contenha apenas o trabalho necessário para completá-lo de forma bem-sucedida. Foco na definição e controle do que está ou não está incluído no escopo do projeto e em um processo bem estruturado de gerenciamento de mudanças	O escopo é fixo apenas quando as iterações estão em andamento. Não há controle formal do escopo ao longo do projeto, havendo a possibilidade de inclusão / alteração das funcionalidades do produto em cada iteração
Gerenciamento de tempo do projeto	Foco na definição das atividades e estimativas de tempo para a elaboração do cronograma detalhado do projeto e no controle, para assegurar a finalização do projeto no prazo	O prazo é estabelecido apenas por iteração ou ciclo. Foco na entregar valor (funcionalidades) o mais rapidamente possível. O cronograma geral é baseado em funcionalidades e não em atividades
Gerenciamento de custos do projeto	Foco na elaboração do orçamento do projeto a partir da necessidade de recursos humanos e materiais e no controle, para garantir que o projeto seja encerrado dentro do orçamento aprovado	Determinação do orçamento em função da funcionalidade do produto requisitada. Os recursos, as funcionalidades e os prazos são balanceados e há uma preocupação em medir o custo por atraso
Gerenciamento da qualidade do projeto	Assegura que o projeto atenda às necessidades para as quais foi concebido. Foco na conformidade e na adequação às especificações e na satisfação das expectativas das partes interessadas no projeto	O sucesso do projeto é definido pelo cliente, que também apresenta seu parecer ao final de cada iteração. Foco na execução da visão e do propósito do produto e na adequação ao uso
Gerenciamento de recursos humanos do projeto	Processos para que se faça o uso mais efetivo de todas as partes interessadas no projeto	Foco na equipe e não no indivíduo. Busca o desenvolvimento de equipes de alto desempenho. Os incentivos são baseados na produtividade do grupo
Gerenciamento das comunicações do projeto	Assegura a geração, a coleta, a disseminação e o armazenamento periódicos das informações do projeto	Foco na eliminação de gastos e de todas as padronizações, documentações e relatórios desnecessários. Garantia de acesso às informações a todos os envolvidos no projeto
Gerenciamento de riscos do projeto	Foco na identificação, na análise e na proposição de respostas aos riscos do projeto	Não há uma maneira padrão sugerida para o tratamento de riscos. Cada projeto deve adotar a sua própria abordagem
Gerenciamento das aquisições do projeto	Foco na aquisição de produtos ou serviços externamente à organização executora, para a realização do projeto	Segue os melhores princípios para aquisição de bens ou serviços dando maior ênfase à colaboração (estabelecimento de parcerias) do que à negociação de contratos

FONTE: ADAPTADO DE UDO; KOPPENSTEINER, 2003, p. 6.

# 2.6 Considerações Finais

Os projetos de desenvolvimento de software são normalmente caracterizados por um elevado grau de incertezas iniciais e, conseqüentemente, por uma grande dificuldade de detalhamento

do escopo e de elaboração de um planejamento completo. Novos processos de negócio, linguagens de programação, ferramentas, arquiteturas e aplicações inovadoras são constantemente introduzidos. Muitos autores afirmam que neste cenário, ciclos rápidos de especificação, teste, validação e implantação podem produzir resultados mais vantajosos que o desenho e o planejamento integral de todo o projeto (AUER; MILLER, 2002; BECK, 2000; BECK; FOWLER, 2001; JEFFRIES *et al*, 2001; CHIN, 2004; COCKBURN, 2001; SCHAWABER, 2002; SCHWABER; BEEDLE 2001; HIGHSMITH, 2002; 2004; POPPENDIECK; POPPENDIECK, 2003; AMBLER, 2002a; COHEN *et al*, 2003; FOWLER, 2003; UDO; KOPPENSTEINER, 2003). Várias iterações em curtos períodos de tempo tendem a maximizar o aprendizado da organização e a potencializar o desenvolvimento da equipe de projeto.

Cohen *et al* (2003, p. 46-47) escreve que "[...] indubitavelmente os Métodos Ágeis vieram para ficar". Entretanto, estes autores, assim como outros (PAULK, 2001; GLASS, 2001; THOMSETT, 2002; UDO; KOPPENSTEINER, 2003; HIGHSMITH, 2004) acreditam que este novo enfoque não ocupará totalmente o espaço dos modelos clássicos de desenvolvimento de software, mas sim que as duas abordagens deverão trabalhar em sintonia (simbiose) no futuro. Embora alguns praticantes enxerguem uma grande distância entre os métodos ágeis e os clássicos, outros acreditam que se pode construir uma ponte entre eles, possibilitando a troca constante de informações e experiências e, conseqüentemente, um aprimoramento do processo de desenvolvimento de software.

Glass (*op. cit.*) destaca que os defensores dos métodos clássicos têm muito a aprender com os adeptos dos métodos ágeis, argumentando que o processo clássico de desenvolvimento pode ser enriquecido com a utilização de algumas das práticas propostas pelos Métodos Ágeis. Em contrapartida, uma das razões para que os Métodos Ágeis não se sobreponham totalmente ao desenvolvimento clássico de software é que alguns processos "antigos" ainda se fazem necessários. Lindvall e Russ (2000) ilustram esta diferença ao mencionar que " [...] desenvolver um software para controlar um ônibus espacial não é o mesmo que desenvolver um software para um torradeira". Cohen *et al* (2003, p. 45) complementam que aplicações que possam vir a colocar em risco a vida humana devem passar por um rígido controle de qualidade, sendo o enfoque clássico preferido para seu desenvolvimento.

Com relação ao gerenciamento de projetos, as ponderações seguem a mesma linha. O fato dos Métodos Ágeis de Desenvolvimento de Software e do Gerenciamento Ágil de Projetos terem a mesma origem e serem construídos sobre os mesmos valores e princípios pode sugerir que, possivelmente, o Gerenciamento Ágil de Projetos seja mais indicado para o gerenciamento de software realizado com o uso de Métodos Ágeis. Contudo, cabe salientar que não foram encontradas evidências durante a revisão bibliográfica que comprovassem (ou negassem) tal afirmação.

Thomsett (2002), Highsmith (2004) e Chin (2004) indicam a utilização do Gerenciamento Ágil de Projetos em projetos que requerem inovação e criatividade ou que estejam sujeitos a alterações constantes dos requisitos do negócio, como por exemplo, os projetos de desenvolvimento de novos produtos (o desenvolvimento de software se enquadra nesta categoria). Mas apesar de mencionarem que o Gerenciamento Clássico de Projetos traz uma rigidez indesejada a projetos desta natureza, estes autores reconhecem a importância do arcabouço de conhecimentos aportado por este enfoque de gerenciamento de projetos, chegando a afirmar que, em determinadas situações, uma combinação do enfoque clássico com as novas práticas propostas pelo Gerenciamento Ágil de Projetos é mais apropriada para que se alcance resultados cada vez efetivos.

Expandindo esta visão, Thomsett (op. cit.) chega a mencionar que as práticas do Gerenciamento Ágil de Projetos podem ser utilizadas até mesmo no desenvolvimento de software conduzido segundo um método clássico. Num contraponto, Paulk (2001) cita que os Métodos Ágeis e o SW-CMM (que tem por base o Gerenciamento Clássico de Projetos) não são incompatíveis, sugerindo que o Gerenciamento Clássico de Projetos pode ser aplicado no desenvolvimento de software realizado com o uso de Métodos Ágeis. Enfim, não há uma opinião única sobre o assunto.

Todavia, é consenso entre os autores pesquisados que se deve fazer uma análise do ambiente interno (aspectos organizacionais e culturais) e do contexto externo no qual o projeto está inserido, para que se selecione o modelo de gerenciamento de projetos (clássico ou ágil) e se defina o método de desenvolvimento (clássico ou ágil) e o respectivo conjunto de processos, técnicas, ferramentas ou práticas a serem seguidos (AMBLER, 2002c; THOMSETT, 2002; COHEN *et al*, 2003; COHN; FORD, 2003; FOWLER, 2003; UDO; KOPPENSTEINER, 2003; HIGHSMITH 2004; CHIN, 2005; NERUR *et al*, 2005).

Por serem assuntos relativamente novos (os primeiros Métodos Ágeis, ainda incipientes, surgiram no final dos anos 90 e o embrião do Gerenciamento Ágil de Projetos surgiu em 2001), pouquíssimas são as pesquisas empíricas que versam sobre os benefícios de sua utilização, não sendo encontrado nenhum estudo, até o momento de encerramento deste trabalho, que investigasse diretamente a questão relativa ao enfoque de gerenciamento de projetos mais apropriado para o desenvolvimento de software com o uso de Métodos Ágeis, o que aumenta a motivação para o desenvolvimento deste estudo.

Finalmente, espera-se que a presente pesquisa traga um valor ao meio acadêmico, ao reunir uma parcela significativa do conhecimento sobre o tema e ao fornecer algumas evidências empíricas da realidade brasileira. Espera-se também que tenha sido traçada a base teórica para a resposta da pergunta-problema desta pesquisa: *Qual o enfoque de gerenciamento de projetos mais apropriado para o desenvolvimento de software conduzido com o uso de um Método Ágil?* 

### 3 METODOLOGIA DE PESQUISA

Este capítulo tem por objetivo apresentar a metodologia de pesquisa utilizada neste estudo. Tendo por base a pergunta-problema da pesquisa, foram avaliadas diferentes estratégias, até se chegar à escolha da mais adequada para este trabalho.

A seguir, será feita a apresentação da tipologia do estudo e das técnicas utilizadas para a coleta e análise de dados. Vale lembrar que o modelo conceitual a ser utilizado nesta investigação foi apresentado no tópico 1.5 – Modelo Conceitual – na introdução deste trabalho.

## 3.1 Tipologia da Pesquisa

O presente estudo é classificado em sua primeira etapa como exploratório e em uma segunda etapa, como quantitativo-descritivo. Esta concepção em dois estágios é recomendada por Ticehurst e Veal (1999) e Cooper e Schindler (2003, p. 135-136) quando não se têm uma idéia clara do problema a ser estudado, sendo possível, através da exploração inicial, o desenvolvimento de conceitos, o estabelecimento de prioridades e o planejamento mais aprimorado da pesquisa. Cooper e Schindler (2003, p. 131) ainda mencionam que, apesar de muitos pesquisadores e administradores darem menos atenção aos estudos exploratórios, talvez por vieses antigamente associados à pesquisa qualitativa, este tipo de estudo não deve ser menosprezado e enfatizam que sua utilização pode conferir uma economia de tempo e dinheiro ao processo de investigação.

Segundo Selltiz et al (1974) e Marconi e Lakatos (2005, p. 189-190), os estudos exploratórios ou formuladores têm por objetivos principais, desenvolver, esclarecer e modificar conceitos e idéias, visando a formulação de problemas mais precisos ou a criação de hipóteses. Também podem ser usados para aumentar o conhecimento do pesquisador acerca de um fenômeno a ser investigado em um estudo posterior, mais estruturado. Os estudos descritivos, por sua vez, agrupam um grande conjunto de interesses de pesquisa. Ao contrário do que ocorre com os estudos exploratórios, caracterizados acima, os estudos descritivos pressupõem um conhecimento anterior sobre o problema a ser estudado. O pesquisador precisa ser capaz de

definir claramente o que deseja medir, de encontrar métodos adequados para esta mensuração, além de especificar quem deve ser incluído na definição de "determinada comunidade" ou "determinada população" (SELLTIZ *et al*, 1974). Marconi e Lakatos (2005, p. 189) complementam que os estudos quantitativo-descritivos consistem em investigações de pesquisa empírica cuja finalidade principal é o delineamento ou a análise das características de fatos ou fenômenos, a avaliação de programas, ou o isolamento de variáveis principais ou chave.

Uma vez que o Gerenciamento Ágil de Projetos é relativamente recente (sua origem data de 2001) e pouco se conhece a seu respeito, o caráter exploratório, que marca o início desta pesquisa, é plenamente justificado para:

- A ampliação do conhecimento sobre o tema e a determinação de uma comparação, à luz do referencial teórico, entre o Gerenciamento Clássico de Projetos e o Gerenciamento Ágil de Projetos;
- A identificação das principais características de um projeto de desenvolvimento de software conduzido com o uso de um Método Ágil;
- A identificação de uma comunidade de pessoas que tenha experiência em projetos de desenvolvimento de software realizados com o uso de um Método Ágil;
- A estruturação detalhada de uma segunda etapa de pesquisa, de caráter quantitativodescritivo, a partir da clara definição do problema e da pergunta de pesquisa e da seleção dos métodos e procedimentos adequados para a condução do estudo.

Por meio do estudo quantitativo-descritivo, tendo por base evidências numéricas e utilizando análises estatísticas, pretende-se descobrir variáveis pertinentes a uma situação específica e determinar relações relevantes entre as variáveis de interesse, com vistas a responder a pergunta-problema proposta no Capítulo 1. Ou seja, deseja-se:

- Investigar a existência de uma associação entre os enfoques de gerenciamento de projeto (ágil ou clássico) e o desempenho dos projetos de desenvolvimento de software realizados com o emprego de um Método Ágil;
- Determinar as técnicas ou características específicas dos Métodos Ágeis que influenciam o desempenho dos projetos de desenvolvimento de software e que determinam a adoção de um ou outro enfoque de gerenciamento de projetos (ágil ou clássico);

- E, se possível, encontrar as variáveis de valor preditivo para a geração de modelos capazes de antecipar o resultado de desempenho de um projeto de desenvolvimento de software realizado com o uso de um Método Ágil e a escolha de um determinado enfoque de gerenciamento de projetos.

Ressalta-se que durante a etapa de exploração desta pesquisa, o estudo conduzido por Lazarevic (2003), cujos resultados foram sumarizados na Revisão Bibliográfica, despertou especial atenção, por seu enfoque diferenciado das demais publicações analisadas. A metodologia empregada na referida pesquisa permitiu ao autor a coleta, a análise de dados e posterior apresentação de resultados referentes a diferentes projetos de desenvolvimento de software, realizados por várias organizações. Ainda, de acordo com Lazarevic (2003), nenhum estudo similar foi realizado anteriormente e, após extensa revisão da literatura, esta colocação foi ratificada.

Desta forma, considerando-se: a) a congruência entre objetivos da referida pesquisa (LAZAREVIC, 2003) e desta aqui apresentada; b) a possibilidade de obtenção de dados de vários projetos distintos em um mesmo trabalho; c) o grau de ineditismo da referida pesquisa (LAZAREVIC, *Ibid.*); e, finalmente, d) a oportunidade de se comparar os resultados de ambos os estudos, decidiu-se pela adoção de uma estratégia de pesquisa similar à utilizada por Lazarevic (2003).

Chama-se a atenção para o fato da incorporação neste trabalho de pequenas modificações nas variáveis e no instrumento de pesquisa proposto por Lazarevic (2003), buscando a perfeita adaptação aos objetivos aqui propostos, além da proposição de técnicas estatísticas adicionais para a análise dos resultados, com o intuito de aumentar o rigor metodológico do trabalho.

#### 3.2 Variáveis da Pesquisa

De acordo com o modelo conceitual apresentado no tópico 1.5 deste trabalho, os enfoques de gerenciamento clássico e ágil de gerenciamento de projetos constituem as variáveis independentes da pesquisa. O gerenciamento de projetos, neste trabalho denominado Gerenciamento Clássico de Projetos, é definido pelo PMI (2004, p.8) como "[...] a aplicação

de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto, a fim de atender aos seus requisitos", sendo realizado através da aplicação e da integração dos seguintes processos de gerenciamento de projetos: iniciação, planejamento, execução, monitoramento e controle e encerramento. O Gerenciamento Ágil de Projetos é definido por Highsmith (2004, p.16) como "[...] um conjunto de valores, princípios e práticas que auxiliam a equipe de projeto a entregar produtos ou serviços de valor em um ambiente desafiador". Para efeito da pesquisa considera-se o grau de combinação entre esses enfoques de gerenciamento de projetos, definido em uma escala de cinco níveis entre "não significativo" a "muito significativo", conforme modelo proposto por Lazarevic (2003).

Como variável dependente, tem-se o desempenho dos projetos de desenvolvimento de software, representando o grau de sucesso obtido pelo projeto em análise. Esta variável é qualificada pelas seguintes opções: "sucesso", "sucesso parcial" ou "insucesso", que correspondem às modalidades utilizadas pelo STANDISH GROUP INTERNATIONAL (1999; 2001; 2003) em suas pesquisas sobre o desempenho dos projetos de tecnologia de informação. Esse critério de classificação também foi utilizado por Lazarevic (2003).

O modelo conceitual ainda leva em consideração as variáveis intervenientes, que têm uma contribuição importante na relação variável independente - variável dependente. Estas variáveis representam as técnicas e características comuns aos principais Métodos Ágeis de Desenvolvimento de Software, selecionadas a partir do estudo de Lazerevic (2003) e da análise da bibliografia correlata (ver tópico 2.4.3), sendo identificadas no instrumento de pesquisa para que se verifique a sua presença nos projetos em análise e se estabeleça (ou não) a relação com o desempenho dos projetos e com o enfoque de gerenciamento de projetos adotado. Lembrando que os Métodos Ágeis de Desenvolvimento de Software são definidos como os que têm por foco: a adaptação dos requisitos de software às necessidades do negócio e não a entrega de especificações pré-definidas; a ênfase nas pessoas e não nos processos; e a ênfase na inovação, na qualidade, na entrega de valor e não na documentação (HIGHSMITH, 2004; COCKBURN, 2001; BECK et al, 2001; UDO; KOPPENSTEINER, 2003; LAZAREVIC, 2003). Cada variável é representada por uma assertiva qualificada segundo uma escala de classificação composta por cinco níveis: "discordo totalmente", "discordo parcialmente", "neutro", "concordo parcialmente" e "concordo totalmente". Estas variáveis são também chamadas explicativas ou preditoras nesta pesquisa.

#### 3.3 Amostra da Pesquisa

Uma vez que esta pesquisa tem como propósito a investigação do enfoque de gerenciamento de projetos mais apropriado para o desenvolvimento de software realizado com o uso de um Método Ágil, decidiu-se pela seleção de uma amostra composta especificamente por pessoas com interesse e/ou experiência na execução ou gerenciamento de projetos de tal natureza no Brasil.

Esse tipo de amostragem, denominado *amostragem intencional por julgamento*, é definido como uma amostragem não-probabilística que atende a critérios pré-estabelecidos (COOPER; SCHINDLER, 2003, p. 169). Cooper e Schindler (*Ibid.*) explicam que, apesar da amostragem probabilística oferecer o verdadeiro corte transversal da população, uma amostragem não-probabilística cuidadosamente controlada pode oferecer resultados aceitáveis, além de trazer benefícios quanto ao custo e ao tempo despendidos no processo. Os autores ainda destacam que, quando se deseja selecionar um grupo viesado para fins de filtragem ou quando a população-alvo não está disponível, esse método de amostragem é uma boa escolha. Estas duas colocações se aplicam à presente pesquisa, justificando assim a aplicação da amostragem intencional por julgamento.

De forma similar ao estudo conduzido por Lazarevic (2003), a amostra foi selecionada entre grupos da Internet especializados na troca de experiência nos diferentes Métodos Ágeis de Desenvolvimento de Software. Ao todo foram identificados dez grupos distintos, distribuídos nas várias regiões do Brasil (Sul, Sudeste, Norte, Nordeste e Centro-Oeste), todos com algum nível de atividade no primeiro semestre de 2005. Esses grupos foram indicados por especialista em Métodos Ágeis contatados durante a fase de exploração desta pesquisa. Para que se conheça o tamanho da população-disponível quando da realização da pesquisa de campo, havia cerca de 1.500 indivíduos pertencentes a esses grupos. Ressalta-se, entretanto, que como a inclusão nos grupos não é excludente, ou seja, um mesmo indivíduo pode participar de mais de um grupo, o número real de pessoas consideradas na amostra tende a ser um pouco menor que este total apresentado. Outro ponto a ser destacado é que não foi encontrado um registro formal da população total ou população-alvo envolvida com o desenvolvimento de software realizado com o uso de Métodos Ágeis no Brasil.

#### 3.4 Coleta de Dados da Pesquisa

Para a coleta de dados desta pesquisa, foram utilizadas fundamentalmente, como fonte primária, a *aplicação de questionários* e, como fonte secundária, a *pesquisa bibliográfica*, seguindo as definições de Marconi e Lakatos (2005). *Entrevistas não-estruturadas* também foram realizadas, em pequena escala, na etapa exploratória deste estudo.

A pesquisa bibliográfica deve abranger, tanto quanto possível, toda a bibliografia já tornada pública em relação ao tema em estudo, desde publicações avulsas, boletins, jornais, revistas, livros, monografias, teses, material cartográfico, etc, até meios de comunicação oral. Sua finalidade é colocar o pesquisador em contato com tudo o que foi dito, escrito ou filmado sobre determinado assunto (MARCONI; LAKATOS, 2005, p. 185). A pesquisa bibliográfica oferece meios para definir e resolver, não somente problemas já conhecidos, como também explorar novas áreas onde os problemas não se cristalizaram bem, ou seja, não se trata de mera repetição do que já foi registrado sobre determinado assunto, mas propicia o exame de um tema sob novo enfoque ou abordagem, chegando a conclusões inovadoras (MARCONI; LAKATOS, *Ibid.*). O material coletado e analisado por meio desta técnica encontra-se consolidado no Capítulo 2 – Revisão Bibliográfica – deste trabalho.

Entrevistas não-estruturadas, definidas como aquelas em que o entrevistador tem liberdade para desenvolver cada situação em qualquer direção que considere adequada (MARCONI; LAKATOS, 2005, P. 199), foram utilizadas na etapa exploratória deste trabalho. Ao todo foram realizadas duas entrevistas informais, por telefone, com especialistas em desenvolvimento de software realizado com o uso de Métodos Ágeis, para a ampliação do conhecimento do pesquisador sobre o tema e para a obtenção de dados sobre a aplicação desses métodos no Brasil. Além disso, informações complementares foram obtidas através da troca de correio eletrônico, com estes especialistas e com um professor da Universidade do Colorado, nos Estados Unidos. Estas informações, aliadas à revisão bibliográfica serviram de base para a estruturação do instrumento de pesquisa.

De acordo com Marconi e Lakatos (2005, p. 203), o questionário é um instrumento de coleta de dados, constituído por uma série ordenada de perguntas, que devem ser respondidas sem a

presença do entrevistador. A elaboração de um questionário requer a observância de normas precisas a fim de aumentar a sua eficácia e exige cuidado na seleção das questões, de modo a oferecer condições para a obtenção de informações válidas. Os aspectos gráficos e de estética também devem ser observados (MARCONI; LAKATOS, 2005, p. 203-206). Uma vez redigido, o questionário precisa ser testado, uma ou mais vezes, antes de sua utilização definitiva, aplicando-se alguns exemplares em uma pequena população escolhida. A tabulação e análise dos dados obtidos em um pré-teste podem identificar problemas como inconsistência ou complexidade de questões, ambigüidade ou linguagem inacessível, existência de perguntas supérfluas, ou mesmo, problemas quanto à forma de entrega / aplicação (MARCONI; LAKATOS, *Ibid.*; COOPER; SCHINDLER, 2003, p. 295-296). Verificadas as falhas, o questionário deve ser reformulado. Marconi e Lakatos (*op. cit.*) mencionam que o pré-teste serve para refinar o instrumento de pesquisa, garantindo que o questionário apresente três importantes elementos:

- *Fidedignidade*: qualquer pessoa que o aplique ao mesmo grupo de respondentes obterá sempre os mesmos resultados;
- *Validade*: os resultados recolhidos são necessários e relevantes à pesquisa.
- *Operatividade*: vocabulário acessível e significado claro.

Com relação às formas de entrega e aplicação de um questionário, Cooper e Schindler (2003, p. 260) mencionam que o questionário auto-administrado tornou-se bastante comum na vida moderna. Os autores comentam que o envio de questionários pelo computador tem se tornado uma prática cada vez mais freqüente, dados os custos menores e a facilidade de aplicação. De forma geral, as técnicas tradicionais de pesquisa por correio podem ser facilmente transpostas para as pesquisas por computador. Além disso, os autores ressaltam que "a Internet disponibiliza software avançado e de fácil uso para a criação de pesquisas na Web", sem que haja a necessidade de domínio de técnicas de programação por parte do pesquisador (COOPER; SCHINDLER, 2003, p. 264). Cooper e Schindler (2003, p. 161) ainda chamam a atenção para outras vantagens do questionário auto-administrado como a possibilidade de contato com respondentes inacessíveis de outra forma, a maior cobertura geográfica a um menor custo, a necessidade de poucos recursos humanos para a aplicação da pesquisa, a sensação de anonimato do respondente, o maior tempo que o respondente tem para pensar sobre a pergunta, o acesso rápido às pessoas que lidam com computador e a coleta mais rápida

de dados. Os mesmos autores também pontuam algumas desvantagens desta forma de entrega e aplicação do questionário, que são apresentadas no tópico 3.6 – Limitações do Método.

Como mencionado anteriormente, o instrumento de pesquisa empregado neste estudo teve por base o questionário utilizado por Lazarevic (2003), traduzido e com pequenas alterações, realizadas para aprimorar o instrumento e adaptá-lo aos objetivos aqui propostos. Para este aprimoramento foram observadas as recomendações feitas por Marconi e Lakatos (2005, p. 204-205) e por Cooper e Schindler (2003, p. 280-296). O questionário, disponível no Anexo A, é composto por 34 questões, divididas em 6 seções, a saber:

- 1. Introdução: para a apresentação do instrumento e dos objetivos da pesquisa;
- 2. Qualificação do Respondente: para a validação da experiência do respondente em projetos de desenvolvimento de software realizado com o uso de Métodos Ágeis, representada pela questão de abertura. Em caso negativo, o respondente é direcionado à seção 6;
- 3. *Caracterização do Respondente*: seção em que são coletados os dados de qualificação propriamente ditos do respondente, compreendendo as questões 1 a 5 (Q1 a Q5<sup>25</sup>);
- 4. Qualificação do projeto: destinada à coleta de informações sobre as características do projeto de desenvolvimento de software, compreendendo as questões 6 a 12 (Q6 a Q12). Nesta seção estão incluídas as questões de interesse da pesquisa, específicas sobre o desempenho do projeto (Q7) e do grau de combinação do enfoque de gerenciamento de projetos adotado (Q8);
- 5. *Técnicas e Características dos Métodos Ágeis*: para verificar se determinadas técnicas ou características comuns aos Métodos Ágeis estão presentes no projeto em análise, sendo representadas pelas questões 13 a 34 (Q13 a Q34);
- 6. Comentários: espaço aberto para que o respondente registre suas considerações.

Quanto às estratégias de resposta, a seção 2 é composta por uma questão de seleção dicotômica (sim ou não), sendo esta a única de resposta obrigatória em todo o instrumento de pesquisa. Nas seções 3 e 4, foram empregadas questões de múltipla escolha e de classificação. A seção 5 é formada basicamente por questões de classificação.

\_

<sup>&</sup>lt;sup>25</sup> Doravante "Qn" é o padrão de notação adotado para designar a questão de número "n".

Inicialmente pretendia-se divulgar o questionário, elaborado em editor de texto, através de correio eletrônico aos grupos selecionados como amostra da pesquisa. Entretanto, após o primeiro pré-teste, realizado com especialistas em Métodos Ágeis, foi identificada a necessidade de publicação do questionário na Internet, dado o perfil técnico do público-alvo, além da necessidade de realização de pequenos ajustes no texto das questões. Foi selecionada, então, uma ferramenta padrão de mercado, de fácil utilização, para a confecção do instrumento de pesquisa e colocação na Web. Esta nova versão do questionário passou pelo pré-teste do próprio pesquisador e dos mesmos especialistas, sendo aprovado para divulgação.

O endereço específico para acesso ao questionário na Web foi enviado, por meio de correio eletrônico, aos associados dos dez grupos especializados selecionados como amostra de pesquisa. No texto da mensagem havia uma carta de apresentação, com a explicação sucinta do estudo, de seus objetivos e do prazo para resposta e um convite à participação. A pesquisa ficou disponível por um período de 30 dias, sendo enviados dois lembretes intermediários (o primeiro após dez dias do início e outro, a uma semana do encerramento da pesquisa), visando a melhorar o índice de retorno. Esta prática é indicada por Cooper e Schindler (2003, p. 260) para pesquisas por correspondência, tendo sido adaptada para esta pesquisa. Cada respondente acessou diretamente o endereço eletrônico informado e respondeu a pesquisa de forma *online*. Os dados foram consolidados na própria ferramenta de mercado e extraídos para futura análise estatística.

#### 3.5 Tratamento de Dados da Pesquisa

O tratamento dos dados desta pesquisa foi feito com a utilização de métodos estatísticos – Análise Descritiva, Análise Discriminante e Regressão Logística – aplicados nesta sequência e descritos a seguir.

#### 3.5.1 Análise Descritiva

Segundo Cooper e Schindler (2003, p. 359), o objetivo da análise descritiva é desenvolver conhecimento suficiente para descrever um conjunto de dados. Isto é feito por meio do

entendimento dos dados coletados, do resumo das informações neles contidas e da disponibilização de tais informações em um formato mais interessante e inteligível. Com a estatística descritiva pode-se descobrir valores mal codificados, dados faltantes e outros problemas no conjunto de dados.

Inicialmente é feita uma análise com as distribuições de freqüências de todas as questões referentes à qualificação dos respondentes e dos projetos de desenvolvimento de software (ver Q1 a Q12 do questionário no Anexo A). Em seguida, procede-se ao estudo de associação por Tabelas de Contingência, etapa descritiva da comparação entre as questões em estudo, com o intuito de identificar as associações marginais de cada questão com as questões de interesse. A quantificação de diferenças na distribuição de freqüências das questões é feita através do nível descritivo (valor "P"), calculado no teste Qui-Quadrado (CONOVER, 1999). Adotando um nível de significância de 5%, considera-se como diferença significativa o resultado P < 0,05. Vale salientar que as conclusões extraídas com base nos níveis descritivos são consideradas, neste trabalho, sob um contexto da análise descritiva e não inferencial dos dados, por uma postura mais conservadora.

Objetivando responder a pergunta-problema, são considerados todos os pareamentos de questões contra Q7 e Q8 do instrumento de pesquisa (ver Anexo A), relativas respectivamente ao desempenho do projeto e ao enfoque de gerenciamento de projetos utilizado.

#### 3.5.2 Análise Discriminante

A análise discriminante faz parte de um conjunto de técnicas estatísticas englobadas pela análise multivariada de dados. Métodos pertencentes a esta classe buscam descrever e resumir a estrutura de diversas variáveis conjuntamente. A análise discriminante, em particular, busca uma regra ótima para separar objetos ou indivíduos em um número pré-estabelecido de grupos, a partir de medições de diversas variáveis (JOHNSON; WICHERN, 1982; MARDIA et al, 1979).

No caso desta pesquisa, a análise é repetida duas vezes, uma considerando o desempenho do projeto e outra, considerando o enfoque de gerenciamento de projetos adotado. No primeiro

caso, a alocação entre os grupos é dada pelas respostas à Q7 do instrumento de pesquisa. No segundo caso, a alocação entre os grupos é dada pelas respostas à Q8. Nas duas modelagens, as variáveis explicativas, correspondentes à Q13 até Q34 (relativas às técnicas e características comuns aos Métodos Ágeis), são usadas para discriminar os indivíduos que qualificam a característica estudada de forma diferente.

A formulação usual desta técnica requer que as variáveis explicativas (preditoras) sejam contínuas (MARDIA *et al*, 1979). Entretanto, nesta pesquisa, as respostas à Q13 até Q34, são a rigor, variáveis qualitativas ordinais, o que poderia gerar um questionamento sobre sua adequação. Mas, de acordo com Conover (1999), o tipo de escala ordinal empregado na presente pesquisa (que prevê as respostas 1, 2, 3, 4 e 5) pode ser considerado robusto o suficiente para ser tratado como uma variável contínua.

Dadas as potenciais 22 variáveis preditoras (Q13 a Q34) que podem fazer parte do modelo de discriminação, é necessário estabelecer um procedimento para selecionar as que têm o maior poder de discriminação entre os dois grupos de interesse. Os procedimentos de seleção são amplamente discutidos na literatura estatística, sendo que cada um tem suas vantagens e desvantagens (MILLER, 1984). Draper e Smith (1981) apontam que, em particular, é comum o uso de procedimentos de seleção *stepwise* por sua simplicidade computacional. Entretanto, apesar de ter propriedades estatísticas bem estudadas, este procedimento é sujeito a críticas no contexto da análise discriminante, como as mencionadas por Miller (1984) e Copas e Long (1991).

Uma vez que o número potencial de variáveis preditoras na presente pesquisa não é muito grande, restringindo a seleção a no máximo quatro questões com o melhor poder de discriminação entre os dois grupos, decidiu-se adotar uma estratégia de seleção de busca exaustiva, computacionalmente intensiva, análoga à descrita em Kudo e Tarumi (1975). Este procedimento prevê o ajuste dos possíveis modelos de análise discriminante com uma até quatro variáveis explicativas, considerando como potenciais variáveis explicativas todas as questões relativas às técnicas e características dos Métodos Ágeis (Q13 a Q34). Uma vez encontrada a equação discriminante, ela pode ser usada para predizer a classificação de uma nova observação (COOPER; SCHINDLER, 2003, p. 458). Além disso, é possível determinar, pela análise dos pesos relativos a cada variável discriminatória, quais têm maior ou menor importância.

#### 3.5.3 Regressão Logística

Apesar do procedimento de busca exaustiva ser bem robusto, ele tem a desvantagem de que suas propriedades estatísticas dependem dos dados amostrais efetivamente observados, não sendo possível utilizar um critério de significância estatística para a escolha dos modelos. O que se faz é selecionar as variáveis que mais aparecem nos melhores modelos classificados de acordo com a proporção de acerto.

Como forma de validação desta estratégia de seleção de modelos, propõe-se uma abordagem alternativa, através do uso de um modelo de regressão logística (AGRESTI, 2002). Nesta abordagem, modela-se a taxa de riscos de um evento binário através de uma função de probabilidade definida, ligada a um conjunto de variáveis preditoras por um termo linear (AGRESTI, *Ibid.*). Para tanto, deve-se ajustar um modelo saturado, formado por um subconjunto das 22 questões preditoras (Q13 a Q34), dada a limitação do grau de liberdade para ajustar o modelo com todas as variáveis preditoras simultaneamente. Para esta préseleção são utilizados, como critério de entrada, os níveis de significância obtidos nos perfis marginais na análise descritiva.

Em seguida, é conduzido um procedimento de seleção *stepwise* pelo Critério de Informação de Akaike (AKAIKE, 1974). Após este procedimento, chega-se a um modelo final com determinadas variáveis, cujo resultado deve ser comparado ao obtido na análise por busca exaustiva, por meio de modelos de análise discriminante. Desta forma, esta modelagem alternativa complementa a estratégia anterior, validando ou não os resultados obtidos.

#### 3.6 Limitações do Método

Apesar da adequação das especificações metodológicas ao tipo de pesquisa em questão e da estruturação do estudo em dois estágios (exploratório e quantitativo-descritivo), há que se expor as limitações existentes. A primeira delas diz respeito ao tipo de amostragem selecionado. Como as generalizações sobre os resultados de um estudo só podem ser feitas

levando-se em consideração a representatividade da amostra, ao se optar por uma amostragem intencional não-probabilística (pelos motivos já justificados), através da qual a probabilidade de selecionar elementos dentro de uma população é desconhecida, havendo maior chance de viés nos resultados do estudo, deve-se deixar claro que as conclusões aqui apresentadas ficam restritas ao âmbito da população amostrada neste trabalho, não podendo ser automaticamente inferidas para situações distintas das aqui abordadas.

Outras limitações estão relacionadas à utilização do questionário auto-administrado. Cooper e Schindler (2003, p. 260) afirmam que a principal limitação neste caso é o erro de não-resposta. Muitos estudos mostram que respondentes com nível educacional mais alto e aqueles interessados no assunto respondem mais a este tipo de pesquisa. Um alto percentual daqueles que respondem normalmente já participou de pesquisas anteriores, enquanto uma grande parcela daqueles que não respondem é composta por "não-respondentes habituais" (COOPER, SCHINDLER, *Ibid.*). Outra limitação diz respeito ao tipo e à quantidade de informações que podem ser obtidas. Geralmente os respondentes se recusam a cooperar quando os questionários são longos e/ou complexos, a não ser que constatem benefícios pessoais. Desta forma não se consegue obter grandes volumes de informação e não se pode aprofundar nas questões (COOPER; SCHINDLER, *Ibid.*). Para minimizar esses efeitos indesejados, aumentando o índice de retorno, houve uma grande preocupação com a elaboração do instrumento de pesquisa e com a forma de envio, sendo realizado um acompanhamento durante o período de resposta. Estas orientações estão em conformidade com a proposição dos referidos autores.

Com relação aos métodos estatísticos empregados, faz-se a ressalva da utilização de variáveis explicativas (preditoras) com respostas em escala ordinal, quando da análise discriminante, conforme defendido por Conover (1999). Entretanto, apesar desta prática ser comum a várias áreas de pesquisa, pode haver uma perda de poder no ajuste dos modelos de predição.

Para atenuar os problemas da confiabilidade e da validação dos resultados deste estudo, esta pesquisa incorpora os quatro critérios propostos por Bradley (1993, p. 436):

- 1. Conferir credibilidade ao material investigado;
- 2. Zelar pela fidelidade no processo de transcrição que antecede a análise;
- 3. Considerar os elementos que compõem o contexto;

4. Assegurar a possibilidade de confirmar posteriormente os dados pesquisados.

### 3.7 Considerações Finais

Neste capítulo foram apresentados a tipologia da pesquisa, a definição das variáveis envolvidas, o tipo de amostragem empregado e as técnicas utilizadas para a coleta e o tratamento dos dados. Foram também expostas as limitações do método intrínsecas a este trabalho. As principais informações que caracterizam metodologia da presente pesquisa são resumidas na Tabela 25 abaixo.

Tabela 25 - Resumo da metodologia de pesquisa empregada

Característica	Classificação
Tipologia da pesquisa	1ª etapa: estudo exploratório
	2ª etapa: estudo quantitativo-descritivo
Amostra de pesquisa	Amostragem intencional por julgamento (não-probabilística)
Técnicas de coleta de dados	Pesquisa bibliográfica, entrevistas não- estruturadas e aplicação de questionários
Métodos para tratamento dos dados	Métodos estatísticos: análise descritiva, análise discriminante e regressão logística.

Por fim, entende-se que, apesar das limitações apontadas, a metodologia de pesquisa definida atende às necessidades desta pesquisa, propiciando o alcance dos objetivos primários e secundários do estudo. Nos próximos capítulos são apresentadas a análise dos resultados e as conclusões da pesquisa e, em seguida, são tecidas as considerações finais deste trabalho.

## 4 RESULTADOS E ANÁLISE

Neste capítulo é apresentada a análise dos dados desta pesquisa. Conforme mencionado anteriormente, o tratamento dos dados é feito através de métodos estatísticos, sendo conduzida primeiramente a análise descritiva e em seguida, a análise discriminante e de regressão logística.

É importante ressaltar que todo tratamento estatístico de dados desta pesquisa, incluindo os procedimentos e os ajustes de modelos necessários à estatística descritiva, à análise discriminante e à regressão logística, foi realizado com o uso da linguagem estatística de código aberto "R" (R PROJECT FOR STATISTICAL COMPUTING, 2005).

#### 4.1 Análise Descritiva dos Dados da Pesquisa

## 4.1.1 Análise Descritiva dos Respondentes e Informações Complementares

A presente pesquisa contou com a participação de 235 respondentes, o que corresponde a um índice de retorno de aproximadamente 16%. Apesar de Cooper e Schindler (2003, p. 260) mencionarem que uma pesquisa por correspondência com retorno aproximado de 30% pode ser considerada satisfatória, acredita-se que o índice alcançado seja adequado a este estudo, provendo um volume de dados robusto o suficiente para a realização das análises estatísticas, além de ser bastante superior aos 4% de retorno obtidos por Lazarevic (2003). Há a possibilidade de que este índice seja ainda maior, uma vez que não se sabe ao certo quantos indivíduos estão vinculados a mais de um grupo, sendo considerado para o cálculo do retorno, o valor aproximado de 1.500 associados de todos os dez grupos, no momento da aplicação da pesquisa. Dos 235 respondentes, 55,3% possuíam experiência prévia no desenvolvimento de software realizado com o uso de Métodos Ágeis, sendo esta a parcela de respondentes qualificados para a continuidade da pesquisa (ver Tabela 26 do Anexo B).

Cabe salientar que, como nem todos os respondentes qualificados preencheram de forma integral o questionário, a partir de agora, ao se mencionar o termo "respondentes" subentendese os indivíduos que efetivamente responderam a questão em análise.

O grupo de respondentes qualificados, em sua maioria (79,4%) composto por gerentes ou diretores de projeto, programadores, analista ou consultores, relatou de forma unânime ter no máximo quatro anos de experiência no desenvolvimento de software realizado com o uso de Métodos Ágeis (ver Tabelas 27 e 28 do Anexo B). Uma vez que o *Manifesto para o Desenvolvimento Ágil de Software* (BECK *et al*, 2001) foi publicado em 2001 e, que no Brasil, os primeiros relatos de aplicação são reportados a partir de 2002 (EXTREME PROGRAMMING BRASIL, 2002), pode-se considerar esta resposta como um indicador de que todos os que participaram da pesquisa, o fizeram com seriedade, provendo informações reais, o que aporta maior confiabilidade aos resultados.

De forma similar ao estudo conduzido por Lazarevic (2003) e ratificando a colocação de COHEN et al (2003, p.12) e de Turk et al (2005, p. 3) de que o Extreme Programming (XP) é o Método Ágil de maior expressão criado nos últimos anos, o resultado da presente pesquisa apontou que 81,2% dos respondentes utilizaram o XP em seus projetos de desenvolvimento de software. Outros métodos como o Scrum, o Feature Driven Developement (FDD), o Adaptative Software Developement (ASD) e o Lean Developement (LD) também foram citados pelos respondentes, porém com menor expressão (ver Tabela 29 do Anexo B). Ainda com relação à experiência, cerca de 45% dos respondentes afirmaram que suas empresas adotavam algum Método Ágil na execução de mais da metade de seus projetos de desenvolvimento software. Isto parece indicar que, uma vez vencida a barreira inicial e havendo condições propícias à adoção destes métodos, conforme discutido por Ambler (2002c), Cohen et al (2003), Cohn e Ford (2003) e Nerur et al (2005), as organizações tendem a adotá-los em todos os seus projetos de desenvolvimento de software, mesmo que de uma forma gradativa (ver Tabela 30 do Anexo B).

Cabe aqui ressaltar que a realização da pesquisa por meio de um questionário eletrônico, acessado na Web, foi elogiada por vários dos respondentes, seja através do próprio formulário (no campo destinado aos comentários) ou por correio eletrônico enviado ao pesquisador, o que indica a escolha acertada da forma de elaboração e envio do instrumento de pesquisa. Por outro lado, deve-se mencionar que em função da lógica imposta pelo questionário eletrônico (que direcionava aqueles que respondiam de forma negativa à questão referente à experiência prévia em Métodos Ágeis diretamente ao campo de comentários), não foi possível traçar um perfil mais detalhado do grupo que, apesar de ter interesse na pesquisa, não foi qualificado em

virtude da resposta negativa à questão de abertura. Com toda a certeza este é um ponto que deve ser melhorado quando da realização de pesquisas futuras.

### 4.1.2 Análise Descritiva dos Projetos de Desenvolvimento de Software

Tendo por foco os projetos de desenvolvimento de software realizados com o uso de Métodos Ágeis considerados pelos respondentes quando da participação na pesquisa, pode-se dizer que, de forma geral, são projetos de curta ou média durações (91,7% têm duração inferior ou igual a 12 meses, sendo que 66,7% têm duração inferior ou igual a 6 meses), formados por equipes pequenas (90,2% dos projetos possuíam até dez integrantes em sua equipe de desenvolvimento), conforme apontam as Tabelas 35 e 36 (ver Anexo B). Estas características estão perfeitamente alinhadas à descrição feita por vários autores, que mencionam que os Métodos Ágeis são usualmente aplicados a projetos de curta ou média durações, realizados com equipes pequenas ou médias (SCHWABER, 2002; HIGHSMITH, 2002; COHEN et al, 2003; UDO, KOPPENSTEINER, 2003; BECK, 2004). Com relação a esse tópico ainda é interessante ressaltar o relato feito por um respondente, de um projeto de maior porte, cuja equipe possuía de 50 a 100 integrantes, como um indício de que é possível utilizar Métodos Ágeis para projetos maiores, como defendem Highsmith (op. cit.), Cohen et al (op. cit.) e Cockburn (2004) e exemplifica Fowler (2003).

Outro ponto importante, que mostra a aderência dos resultados à teoria de que os Métodos Ágeis são em sua essência incrementais, diz respeito ao fato de que 76% dos respondentes afirmaram que a primeira versão do software contemplava no máximo 60% das funcionalidades finais do software, sendo que 36% reportaram que no máximo 20% das funcionalidades foram disponibilizadas na primeira versão (ver Tabela 37 do Anexo B). Este dado não só sugere a realização do desenvolvimento do software em etapas, ou seja, de forma iterativa (por incrementos de funcionalidade), mas também indica uma preocupação em agregar valor, tão logo quanto possível. Este resultado encontra-se totalmente alinhado aos valores e princípios propostos para os Métodos Ágeis e para o Gerenciamento Ágil de Projetos. De acordo com a AGILE ALLIANCE (2005), o primeiro e quarto princípios dos Métodos Ágeis se referem à satisfação do cliente mediante a entrega rápida e freqüente de software de valor e o quinto princípio menciona que a entrega do software em funcionamento

é a medida primária do progresso do projeto. Com relação ao Gerenciamento Ágil de Projetos, Highsmith (2004, p. 27) também ressalta a importância de se agregar valor ao cliente, através de entregas iterativas, baseadas em funcionalidades. Vale mencionar também que apenas 10% dos respondentes indicaram a entrega de mais de 80% das funcionalidades na primeira versão do software.

A presença de um *champion* do projeto, defendida por Ambler (2002), também foi verificada nos projetos analisados. Aproximadamente 66% dos respondentes declaram a existência de um *champion* do projeto trabalhando diretamente com a equipe de desenvolvimento, sendo que cerca de 58% desse grupo mencionou que se tratava de um profissional de uma área de negócio (ver Tabela 34 do Anexo B). Este resultado corrobora a característica de colaboração entre diferentes áreas ou equipes, no desenvolvimento de um software realizado com o uso de Métodos Ágeis.

Considerando o desempenho do projeto, variável dependente desta pesquisa retratada pela Q7, as respostas apontaram, de forma surpreendente, para uma situação inversa à reportada pelas diferentes pesquisas destinadas à avaliação de desempenho de projetos de desenvolvimento de software (STANDISH GROUP INTERNATIONAL, 1999; 2001; 2003; 2004). Se por um lado, o relatório da série *The Chaos Report* elaborado pelo STANDISH GROUP INTERNATIONAL (2003) e sua versão mais recente (ainda que parcial) publicada em 2004 (STANDISH GROUP INTERNATIONAL, 2004), apontavam para cerca de 30% de projetos de sucesso frente a 70% de projetos entregues com algum problema de desempenho, a presente pesquisa relata que 68,9% dos projetos de desenvolvimento de software foram classificados pelos respondentes como bem-sucedidos, sendo que os 31,1% restantes foram qualificados como de sucesso parcial. Nenhum respondente classificou seu projeto como malsucedido (ver Tabela 32 do Anexo B).

Este resultado parece sugerir que os projetos de desenvolvimento de software realizados com o uso de Métodos Ágeis têm mais chance de sucesso que os projetos realizados segundo os métodos clássicos de desenvolvimento de software, ou seja, que os Métodos Ágeis constituem uma solução promissora para o desenvolvimento de software. No entanto, não se deve menosprezar o que Turk *et al* (2003; 2005) relatam como a "empolgação do início", fenômeno em que ao se adotar um novo processo ou método, as pessoas favoráveis a ele enxergam apenas os seus benefícios e valorizam seus resultados de forma não natural. Estes

autores mencionam que dificilmente durante esse momento inicial, indivíduos que se sintam motivados com os Métodos Ágeis qualificariam seus projetos como mal-sucedidos. Sendo assim, não se pode concluir, com base nestes dados, que realmente desenvolvimentos de software realizados com o uso de Métodos Ágeis tenham maior chance de sucesso que aqueles conduzidos por métodos clássicos. Para que se possa fazer uma avaliação mais abrangente é fundamental considerar também os critérios utilizados para mensuração do desempenho e o enfoque de gerenciamento de projetos empregado.

Abordando os critérios utilizados para mensurar o sucesso dos projetos analisados, 45,9% dos respondentes identificaram a satisfação das expectativas dos principais interessados no projeto como o indicador mais importante do sucesso de seus projetos de desenvolvimento de software (ver Tabela 31 do Anexo B). Outros critérios, como o atendimento aos requisitos do projeto e o cumprimento do prazo do projeto, receberam 19,7% e 14,8% das respostas, respectivamente. A satisfação da equipe do projeto apareceu na quarta posição com 6,6% das respostas e a entrega segundo os padrões de qualidade acordados, em quinto lugar, com 4,9% das respostas. O cumprimento dos objetivos de custo e o de indicadores financeiros receberam 1,6% e 3,3% das respostas, respectivamente. Novamente este é um dado interessante, que parece indicar que os projetos de desenvolvimento de software analisados não tinham como critério básico para a mensuração de sucesso, o tradicional cumprimento do objetivo triplo do projeto, defendido por Meredith e Mantel (2000, p. 4) e utilizado como indicador de desempenho para as pesquisas conduzidas pelo STANDISH GROUP INTERNATIONAL (1999; 2001; 2003). Talvez a abordagem proposta por Thomsett (2002, p. 75) através de seu conjunto de indicadores de sucesso de um projeto seja mais condizente com os resultados observados.

Finalizando a etapa de descrição preliminar dos dados relativos aos projetos, deve-se abordar a Q8 (ver Anexo A), referente ao enfoque de gerenciamento de projetos aplicado às iniciativas de desenvolvimento de software analisadas na pesquisa. As possíveis respostas à Q8 são distribuídas em cinco classificações: (1) "Não significativo", (2) "Pouco significativo", (3) "Moderado", (4) "Significativo" e (5) "Muito significativo". As respostas (1) e (2) indicam o uso do Gerenciamento Ágil de Projetos, a resposta (3), uma combinação dos enfoques ágil e clássico e as respostas (4) e (5) apontam para a adoção do Gerenciamento Clássico de Projetos. Cerca de 55% dos respondentes afirmaram que houve uma combinação entre os enfoques ágil e clássico no gerenciamento de seu projeto de desenvolvimento de software;

23,4% dos respondentes selecionaram opções relativas à adoção do Gerenciamento Ágil de Projetos e 21,3% informaram o uso do Gerenciamento Clássico de Projetos (ver Tabela 33 do Anexo B). Esta distribuição sugere que, apesar de se tratar de desenvolvimento de software realizado com o uso de Métodos Ágeis, é possível adotar qualquer um dos enfoques de gerenciamento de projetos – clássico ou ágil – o que, mais uma vez, se mostra aderente às discussões apresentadas pela literatura específica. Mas observa-se que o uso de uma combinação entre os enfoques ágil e clássico mostra-se mais freqüente. Vale lembrar que autores como Thomsett (2002), Highsmith (2004) e Chin (2004) defendem a adoção do Gerenciamento Ágil de Projetos para o desenvolvimento de software (assim como para quaisquer outras iniciativas que envolvam inovação), mas aceitam a possibilidade de combinação entre os enfoques ágil e clássico em determinadas situações. Paulk (2001), por sua vez, é partidário de que o SW-CMM (calcado nos princípios do Gerenciamento Clássico de Projetos) seja aplicado nos desenvolvimentos de software realizados com o uso de Métodos Ágeis.

O fato é que o resultado encontrado para a Q8, diferentemente do se poderia esperar, indica que para o desenvolvimento de software realizado com o uso de Métodos Ágeis, o Gerenciamento Ágil de Projetos ainda não se mostra como a opção prevalecente. Por outro lado, o mesmo resultado aponta que certas organizações já iniciaram a sua aplicação, optando em sua maioria por uma mescla entre os enfoques ágil e tradicional de gerenciamento de projetos.

# 4.1.3 Análise Descritiva das Práticas Relativas aos Métodos Ágeis

As questões Q13 a Q34 do instrumento de pesquisa (ver Anexo A) se referem às principais práticas e características relativas aos Métodos Ágeis de Desenvolvimento de Software. Nesta seção da pesquisa, para cada uma das assertivas propostas, o respondente foi convidado a escolher entre as respostas "Discordo Totalmente", "Discordo Parcialmente", "Neutro", "Concordo Parcialmente" e "Concordo Totalmente", que na codificação de dados correspondem aos escores 1, 2, 3, 4 e 5, respectivamente. A análise exposta a seguir leva em consideração o cálculo do escore médio de resposta e a distribuição de freqüências das respostas de cada uma destas questões, conforme mostra a Tabela 38 (ver Anexo B).

À exceção da Q21, que menciona que "a equipe de desenvolvimento recebeu treinamento formal no enfoque ágil", todas as demais questões tiveram um escore médio de resposta superior a 3 e algumas vezes superior a 4 (ver Tabela 38 do Anexo B). Isto significa que, de forma geral, a maioria dos respondentes concordou parcial ou totalmente que as técnicas e características dos Métodos Ágeis apresentadas estavam presentes em seus projetos de desenvolvimento de software.

Cohen et al (2003) mencionam que os Métodos Ágeis requerem menos treinamento formal que os métodos clássicos de desenvolvimento de software. Isto porque algumas técnicas específicas da abordagem ágil como, por exemplo, a programação em pares, minimizam a necessidade de treinamento, uma vez que as pessoas aprendem uma com as outras. Analisando as respostas da Q21 e da Q22 percebe-se que apenas 26% dos respondentes afirmaram ter havido um treinamento formal no Método Ágil durante o desenvolvimento do software; por outro lado, 70% dos respondentes mencionaram ter recebido algum tipo de treinamento informal. Apesar de autores como Bohem (2002), Cohn e Ford (2003), Fowler (2003) e Nerur et al (2005) ressaltarem a importância de a equipe de desenvolvimento do projeto estar familiarizada com os Métodos Ágeis, percebe-se que o processo de treinamento ou capacitação nas práticas dos Métodos Ágeis pode ser feito informalmente, conforme proposto por Cohen et al (2003). Esta abordagem está refletida no contexto dos projetos analisados nesta pesquisa, sendo constatada uma pequena incidência de capacitação formal da equipe de desenvolvimento nos Métodos Ágeis e uma parcela significativa de treinamento informal, provavelmente realizado através de práticas vivenciadas no dia-a-dia do projeto, como a propriedade coletiva do código. Mesmo assim, 68,9% dos projetos foram considerados bem-sucedidos pelos respondentes (ver Tabela 32 do Anexo B), sugerindo que a ausência de treinamento formal não comprometeu os resultados dos projetos.

Outras questões também chamam a atenção nesta primeira análise, pelas altas médias alcançadas por suas respostas (acima de 4) e por seus resultados se mostrarem bastante aderentes à teoria apresentada no Capítulo 2.

Cerca de 80% dos respondentes afirmaram que houve boa coordenação e colaboração entre os diferentes grupos envolvidos nos projetos analisados (ver Q13 e Q15 – Tabela 38 do Anexo B). O espírito de colaboração e a coordenação eficaz entre as várias equipes participantes do

projeto são considerados extremamente críticos para o sucesso de projetos de desenvolvimento de software realizados com o uso de Métodos Ágeis por vários autores, como por exemplo, Ambler (2002), Cohen *et al* (2003), Cohn e Ford (2003), Nerur *et al* (2005) e Turk *et al* (2005). Vale lembrar também que um dos valores-chave do Métodos Ágeis é enunciado como a priorização da colaboração sobre a negociação de contratos (BECK *et al*, 2001).

Ao se avaliar a capacidade e habilidade dos integrantes da equipe de desenvolvimento, verifica-se que 86% dos respondentes afirmaram que a equipe de desenvolvimento foi composta por profissionais bem qualificados tecnicamente (ver Q20 – Tabela 38 do Anexo B). Bohem (2002) enfatiza a importância de se ter uma equipe bem treinada e constituída por especialistas, uma vez que os Métodos Ágeis depositam elevado grau de confiança no conhecimento tácito e na capacidade de tomada de decisões de cada indivíduo. Novamente, o resultado encontrado valida outro valor-chave dos Métodos Ágeis, proposto por Beck *et al* (2001), referente à valorização das pessoas.

Ainda com relação à equipe de desenvolvimento, 90% dos respondentes concordaram parcial ou integralmente que os programadores e analistas estavam bastante motivados com o projeto; 84% afirmaram ter havido um clima de descontração e alegria entre a equipe durante o desenvolvimento (ver Q25 e Q32 – Tabela 38 do Anexo B). A valorização do indivíduo, a equipe composta por especialistas, o espírito de colaboração e o próprio fato de se trabalhar com algo relativamente novo podem explicar esta motivação e o clima de alegria predominante nos projetos analisados. Este resultado está alinhado à colocação de Juric (2002) que menciona que os Métodos Ágeis são uma forma divertida de se desenvolver software.

A necessidade de mudanças nos requisitos do software também se mostrou presente nos projetos analisados, sendo esta uma das características marcantes dos projetos inseridos em ambientes de negócio dinâmicos. Ao todo, 90% dos respondentes mencionaram a ocorrência de mudanças (adaptações) ao longo do projeto e 86% deles concordaram que as solicitações de mudanças foram incorporadas sem problemas ao longo das várias versões do software (ver Q23 e Q24 – Tabela 38 do Anexo B). Mais uma vez, os resultados se mostraram aderentes à teoria, haja vista que os Métodos Ágeis surgiram como uma resposta às necessidades de constantes mudanças nos requisitos do software para atender os clientes ou o negócio,

incentivando as adaptações e tendo como outro valor-chave a priorização das respostas às mudanças sobre o seguimento de um plano (BECK *et al*, 2001).

Considerando a questão relativa ao apoio da alta administração ao projeto de desenvolvimento de software, tem-se que 70% dos respondentes indicaram que houve tal apoio no decorrer do projeto (ver Q14 – Tabela 38 do Anexo B). De forma geral, este é um dos pontos citados como um dos fatores críticos de sucesso dos projetos de tecnologia de informação (Jiang *et al*, 1996; STANDISH GROUP INTERNATIONAL, 1999; 2001). Ambler (2002), Cohn e Ford (2003) e Nerur *et al* (2005) compartilham a visão de que é imprescindível o envolvimento e o apoio da alta administração nos projetos de desenvolvimento de software, em especial quando do uso de Métodos Ágeis, uma vez que a adoção destes métodos pressupõe mudanças de âmbito técnico, gerencial e cultural.

Caso seja de interesse do leitor, as respostas às demais questões relativas às técnicas e características dos Métodos Ágeis podem ser visualizadas na Tabela 38 (ver Anexo B). Vale ressaltar que as colocações até aqui apresentadas (tópicos 4.1.1 a 4.1.3) correspondem a uma análise descritiva preliminar, que por si só não forneceu os subsídios necessários para responder a pergunta-problema desta pesquisa, garantiu o melhor entendimento dos dados e do contexto da pesquisa.

### 4.1.4 Análise de Associação por Tabelas de Contingência

A seguir, tem-se a segunda etapa da estatística descritiva voltada à identificação das associações marginais entre as variáveis de interesse (relativas ao desempenho do projeto e ao enfoque de gerenciamento de projetos utilizado e retratadas, respectivamente, pela Q7 e pela Q8) e todas as demais variáveis da pesquisa. Esta análise é feita através da construção de Tabelas de Contingência de dupla entrada e da verificação do nível descritivo "P", calculado pelo teste Qui-Quadrado (CONOVER, 1999), considerando um nível de significância de 5%. Em termos práticos, ao se analisar todos os possíveis pareamentos relativos à Q7, busca-se identificar as variáveis que têm alguma relação ou que podem influenciar o desempenho de um projeto, ou seja, pretende-se determinar, de forma preliminar, as variáveis que correspondem potencialmente aos fatores críticos de sucesso de um projeto de

desenvolvimento de software realizado com o uso de Métodos Ágeis. Avaliando os possíveis pareamentos relativos à Q8, é possível determinar fatores que podem influenciar a escolha de um determinado enfoque de gerenciamento de projetos. Contudo, antes de apresentar a análise propriamente dita, é importante destacar que:

- Uma vez que os resultados da Q7, referente ao desempenho do projeto, recaíram sobre duas possibilidades de resposta "Sucesso" e "Sucesso Parcial" (ver Tabela 32 do Anexo B)
   para a estruturação das Tabelas de Contingência e o cálculo do nível descritivo serão consideradas apenas estas duas opções.
- Com relação à Q8, relativa ao grau de combinação entre os enfoques ágil e clássico de gerenciamento de projetos, qualificada por cinco níveis (1) "Não Significativo", (2) "Pouco Significativo", (3) "Moderado", (4) "Significativo" e (5) "Muito Significativo", procedeu-se ao seguinte agrupamento: as respostas (1) e (2) formam o grupo "ágil" relativo à aplicação exclusiva do Gerenciamento Ágil de Projetos; as respostas (3), (4) e (5) correspondem a um grupo denominado "moderado clássico", que engloba desde a combinação dos enfoques ágil e clássico à aplicação exclusiva do Gerenciamento Clássico de Projetos.

Vale salientar que os níveis descritivos obtidos a partir dos testes de associação Qui-Quadrado devem ser considerados com cautela. Devido à ocorrência de caselas com freqüências nulas, identificadas durante a etapa inicial de análise dos dados, a aplicação apropriada dos testes estatísticos fica inviabilizada. Este cuidado é tomado durante a análise dos resultados, ao se conferir a eles um caráter descritivo mais do que inferencial.

Considerando todos os pareamentos possíveis relacionados à Q7 (ver Tabelas 39 a 70 do Anexo B), percebe-se que apenas as Q23, Q25, Q30, Q31 e Q33, cujos enunciados podem ser vistos abaixo, apresentam um nível de associação considerado estatisticamente significativo, com o nível descritivo "P" inferior a 5%. A maioria destas questões está relacionada às técnicas ou características dos Métodos Ágeis de Desenvolvimento, à exceção da Q33, relativa ao enfoque de gerenciamento de projetos.

- Q23: Os desenvolvimentos sofreram modificações (adaptações) ao longo do projeto;
- Q25: Os programadores e analistas estavam muito motivados com o projeto;

- Q30: Os desenvolvimentos diários de sistemas eram entregues com a utilização de técnicas dos Métodos Ágeis;
- Q31: Os membros das equipes de desenvolvimento reviam a codificação desenvolvida por outra pessoa, antes da liberação final;
- Q33: O enfoque ágil é mais indicado para o gerenciamento de projetos de desenvolvimento de sistemas de TI do que o enfoque clássico de gerenciamento de projetos (como por exemplo, o enfoque baseado em processos conforme proposto pelo PMBoK PMI).

Dentre as cinco questões selecionadas, a Q25, que versa sobre a motivação dos programadores e analistas, é a que apresentou o maior nível de associação com o desempenho do projeto (ver Tabela 61 do Anexo B). Nos projetos bem-sucedidos, 97% dos respondentes concordaram que a equipe de desenvolvimento estava bastante motivada (79% concordaram totalmente e 18% concordaram parcialmente). Já nos projetos classificados como de "Sucesso Parcial", apesar do percentual de respondentes que concordaram com a afirmação ser alto (cerca de 86%), era inferior ao do primeiro grupo e a distribuição se mostrou diferente (29% concordaram totalmente e 47% concordaram parcialmente). Percebe-se então uma relação bastante forte entre a motivação da equipe e o desempenho do projeto.

Este resultado vem ampliar a lista de estudos já realizados que identificaram o elevado grau de motivação das equipes como um dos pontos de maior destaque dos projetos de desenvolvimento de software realizados com o uso de Métodos Ágeis. Como exemplos destes estudos, podem ser citados: Cockburn e Highsmith (2001b), Reifer (2002), Lazarevic (2003), Gawlas (2004) e Bonato (2004). Cockburn e Highsmith (2001b) e Highsmith (2004) vão ao extremo ao mencionar, que uma equipe motivada é capaz de fazer qualquer projeto ser um sucesso, independente do método de desenvolvimento e do enfoque de gerenciamento empregado.

Considerando os dados da Tabela 59 (ver Anexo B), referentes à associação Q23 *versus* Q7, percebe-se pela análise da distribuição das freqüências e do escore médio das respostas dadas, tanto por aqueles que qualificaram seu projeto como de "Sucesso", quanto por aqueles que qualificaram-no como de "Sucesso Parcial", que houve várias modificações nos requisitos do software no decorrer do projeto (cerca de 90% dos respondentes dos dois grupos concordaram com esta afirmação). Expandindo esta análise, dois pontos chamam a atenção: aparentemente

houve mais mudanças ao longo dos projetos classificados como bem-sucedidos (ver Tabela 59 do Anexo B); e ainda, 91% daqueles que classificaram o projeto como de "Sucesso", contra 76% dos que qualificaram seu projeto como de "Sucesso Parcial", concordaram que as mudanças foram absorvidas sem problemas ao longo das várias versões do software desenvolvidas (ver Tabela 60 do Anexo B).

Estas observações sugerem que a absorção das solicitações de alteração no decorrer do projeto e, principalmente, a adaptação tranqüila dos requisitos do software às necessidades do cliente ou do negócio ao longo das várias versões desenvolvidas, levaram a um melhor atendimento às expectativas dos clientes e, conseqüentemente, a um projeto bem-sucedido. Deve-se lembrar que esta necessidade de adaptação constante dos requisitos do software ao longo do processo de desenvolvimento, dadas as mudanças no cenário de negócio, foi um dos principais determinantes da criação dos Métodos Ágeis de Desenvolvimento de Software e do Gerenciamento Ágil de Projetos (BECK *et al*, 2001; COHEN *et al*, 2003; HIGHSMITH, 2004; CHIN, 2004). Pode-se dizer, então, que os resultados da presente pesquisa reafirmam a existência de uma associação direta entre a adaptação tranqüila dos requisitos do software e o sucesso de um projeto de desenvolvimento de software realizado com o uso de Métodos Ágeis.

Antes da análise das associações da Q30 e da Q31 com a Q7, cabe comentar o resultado do pareamento Q26 *versus* Q7, apesar desta associação não se mostrar estatisticamente significativa. Pelos escores médios das respostas à Q26, percebe-se que os respondentes que qualificaram seu projeto como de "Sucesso" estavam mais familiarizados com as práticas dos Métodos Ágeis de Desenvolvimento de Software do que aqueles que classificaram-no como de "Sucesso Parcial" (ver Tabela 62 do Anexo B).

Os resultados das associações Q30 *versus* Q7 e Q31 *versus* Q7 indicam que, diferentemente do ocorrido nos projetos de "Sucesso Parcial", as práticas dos Métodos Ágeis eram aplicadas no dia-a-dia dos projetos bem-sucedidos e que havia uma grande preocupação com a revisão do código por outro integrante da equipe de desenvolvimento antes de sua liberação final, prática esta que assegura um melhor padrão de qualidade no software entregue (ver Tabelas 66 e 67 do Anexo B). Esta diferença de comportamento pode ser explicada pelo fato de o grupo de respondentes dos projetos de "Sucesso" ter indicado que a equipe de

desenvolvimento recebeu um treinamento (mesmo que de maneira informal) nos Métodos Ágeis, o que não se percebe com o outro grupo (ver Tabela 58 do Anexo B).

Conforme já mencionado, a relação entre a familiaridade ou o uso das práticas dos Métodos Ágeis e o desempenho de um projeto de desenvolvimento de software realizado segundo enfoque ágil já foi discutida por Bohem (2002), Cohen *et al* (2003), Cohn e Ford (2003), Fowler (2003) e Nerur *et al* (2005). Estes autores destacam que tanto as equipes de projeto, como os gerentes, devem estar preparados e familiarizados com esta nova abordagem de desenvolvimento de software, uma vez que ela significa não apenas uma mudança de processo, mas sim uma mudança de cultura da organização e de postura frente ao projeto.

Pela análise do pareamento Q33 *versus* Q7 (ver Tabela 69 do Anexo B), percebe-se que cerca de 65% dos respondentes (independente da classificação do desempenho do projeto) concordaram com a afirmativa de que o enfoque ágil de gerenciamento de projetos é mais apropriado para o desenvolvimento de software conduzido com o uso de Métodos Ágeis. Contudo, aqueles que qualificaram seu projeto como de "Sucesso" foram mais enfáticos nesta afirmação. Este resultado está bastante alinhado às colocações de Thomsett (2002) e, principalmente, às de Highsmith (2004) e de Chin (2004), que indicam o uso do Gerenciamento Ágil de Projetos no desenvolvimento de software. Este seria o resultado esperado, uma vez que o Gerenciamento Ágil de Projetos e os Métodos Ágeis são construídos sobre valores e princípios similares.

O que chama ainda mais a atenção neste ponto, é o fato de que ao se buscar a associação direta entre as variáveis relativas ao desempenho do projeto (Q7) e o grau de combinação entre os enfoques ágil e clássico de gerenciamento de projetos (Q8), procurando identificar se o uso de um determinado enfoque de gerenciamento de projetos exerce influência sobre o desempenho de um projeto de desenvolvimento de software realizado com o uso de Métodos Ágeis, os resultados observados apresentaram um nível descritivo bem superior a 5% (ver Tabelas 44 e 76 do Anexo B). Deve-se lembrar que, de fato, a Q8 não aparece na lista apresentada de questões com associação estatisticamente significativa com a Q7. Face a este resultado, há que se colocar que com os dados disponíveis não houve evidência amostral para encontrar tal associação.

Entretanto, não se deve desprezar os resultados do pareamento Q7 *versus* Q8 (ver Tabela 76 do Anexo B). Fazendo uma análise puramente descritiva dos dados desta tabela, percebe-se que 91% daqueles que optaram pelo Gerenciamento Ágil de Projetos, classificaram o desempenho final de seu projeto como de "Sucesso", enquanto 64% dos que optaram por um enfoque misto ou pelo Gerenciamento Clássico de Projetos classificaram seu projeto como bem-sucedido. Esta observação parece indicar que ao se optar pelo Gerenciamento Ágil de Projetos no desenvolvimento de software realizado com o uso de Métodos Ágeis há uma maior chance de se alcançar o sucesso do projeto, sugerindo que este seria o enfoque mais apropriado para projetos desta natureza. Mas dada a ausência de evidências estatísticas, tal colocação não pode ter um efeito conclusivo.

Continuando a análise referente ao enfoque de gerenciamento de projetos, não se pode esquecer que 55% dos respondentes selecionaram uma combinação entre os enfoques ágil e clássico de gerenciamento de projetos e os demais 45% se distribuíram quase que de maneira uniforme entre o Gerenciamento Ágil de Projetos e o Gerenciamento Clássico de Projetos (ver Tabela 33 do Anexo B). Este ponto já foi discutido no tópico 4.1.2, mas vale complementar que, com os dados disponíveis na pesquisa, não se consegue concluir se esta grande concentração em um modelo intermediário de gerenciamento de projetos, que traz em si características ágeis e clássicas, está relacionada a um período de transição, ou se esta é uma opção consciente e definitiva daqueles que visualizaram nesta combinação, a saída para alcançarem um bom desempenho de seus projetos de desenvolvimento de software realizados com o uso de Métodos Ágeis. Cabe lembrar que até mesmo os defensores do Gerenciamento Ágil de Projetos, como Highsmith (2004) e Chin (2004), mencionam que o uso desta combinação entre os enfoques ágil e clássico pode ser uma opção interessante em determinadas situações.

Outra associação que merece destaque, apesar de também não ter apresentado um resultado estatisticamente significativo, é a da Q34 *versus* Q7 (ver Tabela 70). Ao se analisar a distribuição de respostas, percebe-se que 82% daqueles que classificaram o projeto como bem-sucedido concordaram com a afirmação de que o uso de um Método Ágil foi um fator fundamental para o alcance desse desempenho. Por outro lado, 53% dos que classificaram o projeto como de "Sucesso Parcial" apontaram o uso de um Método Ágil como fator importante para o alcance do sucesso. Estes dados podem ser explicados ao se retomar a discussão sobre a familiaridade com as práticas dos Métodos Ágeis e sua aplicação no dia-a-

dia do projeto, apresentada quando das análises das associações da Q7 com a Q26, Q30 e Q31 feitas anteriormente.

Passando para a análise dos pareamentos possíveis entre a Q8 (relativa ao grau de combinação entre os enfoques de gerenciamento de projetos) e as demais questões de pesquisa, retratados nas Tabelas 71 a 102 (ver Anexo B), percebe-se que outras questões se destacam por apresentarem uma associação significativa do ponto de vista estatístico. São elas: Q2, Q14, Q27, Q30, Q33 e Q34, cujos enunciados são apresentados a seguir. À exceção da Q2 que corresponde a uma questão de caracterização do respondente, as demais, se relacionam às técnicas e características dos Métodos Ágeis de Desenvolvimento de Software:

- Q2: Há quanto tempo você trabalha com Métodos Ágeis?
- Q14: Houve um grande apoio da alta administração no projeto;
- Q27: O código foi desenvolvido com um sentimento de propriedade coletiva (os programadores se sentiam à vontade e capazes de alterar o código desenvolvido por outro profissional);
- Q30: Os desenvolvimentos diários de sistemas eram entregues com a utilização de técnicas dos Métodos Ágeis;
- Q33: O enfoque ágil é mais indicado para o gerenciamento de projetos de desenvolvimento de sistemas de TI, do que o enfoque clássico de gerenciamento de projetos (como por exemplo, o enfoque baseado em processos conforme proposto pelo PMBoK PMI);
- Q34: O uso de um Método Ágil foi fundamental para o sucesso do projeto de desenvolvimento do Sistema X.

Analisando a relação de questões apresentada acima, o primeiro comentário a ser feito é que há pouca concordância com a lista encontrada na análise dos pareamentos com a Q7. Somente as questões 30 e 33 aparecem nas duas relações. Esta constatação corrobora o fato de não ter sido encontrada uma associação significativa do ponto de vista estatístico, entre Q7 e Q8, conforme já explicado. Caso esta associação fosse encontrada, seria de se esperar que as duas listas apresentassem mais pontos em comum.

Tendo por foco a associação Q2 *versus* Q8 (ver Tabela 72 do Anexo B), percebe-se que 91% dos respondentes que optaram pelo Gerenciamento Ágil de Projetos têm de um a quatro anos

de experiência no desenvolvimento de software realizado com o uso de Métodos Ágeis, sugerindo eles integram um público com um pouco mais de vivência em projetos que usam este novo modelo de desenvolvimento de software. Aparentemente, resultado se mostra compatível com Thomsett (2002), que afirma que os projetos de desenvolvimento de software têm sempre duas vertentes (uma técnica e outra gerencial) e que as organizações, usualmente, aprimoram primeiramente os modelos de desenvolvimento (aspecto técnico), para depois introduzir melhorias de âmbito gerencial. Este fato também parece estar alinhado à colocação de Highsmith (2004) de que o Gerenciamento Ágil de Projetos surge como uma evolução natural, porém com ênfase gerencial, do modelo utilizado para o desenvolvimento ágil de software.

Fazendo uma análise descritiva geral de todas as Tabelas de Contingência geradas para as associações com a Q8 (ver Tabelas 71 a 102 do Anexo B), um comportamento chama a atenção: de forma geral, o grupo de respondentes que optou pelo Gerenciamento Ágil de Projetos teve um escore médio de respostas acima de 4,5, ou seja, suas respostas se concentraram entre "Concordo Parcialmente" e "Concordo Totalmente", em praticamente todas as questões relativas às técnicas e às características dos Métodos Ágeis, salvo pequenas exceções (referentes ao treinamento formal e informal nos Métodos Ágeis, já discutidos anteriormente). Mesmo nestes casos de exceção, as médias foram superiores a 2,5. Isto mostra não só uma coerência de respostas, mas também indica que a opção pelo Gerenciamento Ágil de Projetos é feita em um ambiente em que as práticas dos Métodos Ágeis já estão consolidadas, reforçando a idéia de que se trata de uma evolução do conceito do desenvolvimento ágil (HIGHSMITH, 2004), conforme discutido no parágrafo anterior.

Considerando a questão relativa ao apoio da alta administração no projeto (ver Tabela 82 do Anexo B), percebe-se que a totalidade dos respondentes que optaram pelo Gerenciamento Ágil de Projetos mencionou que houve um grande apoio da alta administração, sendo que 90% concordaram totalmente e 10% concordaram parcialmente. Este resultado mostra de forma clara que a adoção do Gerenciamento Ágil de Projetos pressupõe um alto comprometimento e apoio do corpo diretivo da organização, uma vez que se trata não apenas da simples substituição de processos e ferramentas, mas sim de uma mudança de cultura e da forma de gerenciamento e controle do projeto, do estabelecimento de um ambiente de respeito e de confiança mútua entre "clientes e fornecedores", onde haja a valorização das pessoas e o incentivo à tomada de decisões (COCKBURN; HIGHSMITH, 2001a; HIGHSMITH, 2004;

NERUR *et al*, 2005). Levando em consideração o grupo que optou por enfoques gerenciamento de projetos variando de moderado a clássico, percebe-se que 62% confirmam que houve apoio da alta administração em seus projetos.

Com relação à associação Q30 *versus* Q8 (ver Tabela 98 do Anexo B), percebe-se que 70% dos respondentes que indicaram o emprego do Gerenciamento Ágil de Projetos mencionaram diretamente a utilização das práticas dos Métodos Ágeis no dia-a-dia de seus projetos de desenvolvimento de software. Dentre aqueles que optaram pelo enfoque moderado ou pelo Gerenciamento Clássico de Projetos, 59% indicaram o uso diário destas práticas.

Já a análise do pareamento Q27 *versus* Q8 (ver Tabela 95 do Anexo B) indica claramente que nos projetos de desenvolvimento de software gerenciados segundo o enfoque ágil, houve a adoção da prática de "propriedade coletiva do código", através da qual os programadores se sentiam à vontade e capazes de alterar o código desenvolvido por outro profissional (70% dos respondentes concordaram totalmente e 30% concordaram parcialmente). Para que se consiga isto é imprescindível o estabelecimento de um clima de confiança e respeito mútuos no projeto e que haja uma predisposição e uma maturidade da equipe para tal (Nerur *et al*, 2005). O que se pode perceber nesta pesquisa, é que tal prontidão e experiência prévia estavam presentes com maior força nos projetos dos respondentes que indicaram o uso do Gerenciamento Ágil de Projetos, uma vez que o mesmo comportamento de resposta se repetiu em várias outras questões relativas às técnicas e características dos Métodos Ágeis, como anteriormente comentado. Para ilustrar o comportamento do grupo de respondentes que selecionou um enfoque moderado ou clássico de gerenciamento de projetos, tem-se que 59% indicaram a adoção da prática de "propriedade coletiva do código", padrão esse que se repetiu nas respostas de várias outras questões.

Ainda, de acordo com Lazarevic (2003, p.28), a propriedade coletiva do código encoraja a sinergia, a colaboração e o trabalho em equipe. Atingir um estágio em que os resultados do grupo sejam melhores que a contribuição de cada indivíduo é o objetivo do Gerenciamento Ágil do Projeto (HIGHSMITH, 2004). De acordo com Sutherland (2001), quando os indivíduos se ajudam mutuamente e contribuem para o todo, a equipe de desenvolvimento atinge o estado de hiper-produtividade.

Seguindo esta linha de raciocínio, o resultado da associação Q34 *versus* Q8 (ver Tabela 102 do Anexo B) mostra que todos os respondentes que selecionaram o Gerenciamento Ágil de Projetos concordam quase que integralmente que o uso dos Métodos Ágeis foi fundamental para o sucesso do desenvolvimento de software em questão. Em contrapartida, 62% dos que selecionaram os enfoques moderado ou clássico de gerenciamento de projetos compartilham tal posição. Novamente cabe salientar que a questão de prontidão para o uso dos Métodos Ágeis pode ser determinante para este comportamento.

A associação Q33 *versus* Q8 (ver Tabela 101 do Anexo B), por sua vez, permite avaliar se o enfoque de gerenciamento de projetos adotado durante o desenvolvimento do software foi considerado de fato o mais apropriado para o projeto em questão. O resultado encontrado se mostrou bastante interessante: apenas 10% dos que indicaram o uso do Gerenciamento Ágil de Projetos não concordam que este seja o enfoque mais apropriado para o gerenciamento de projetos de desenvolvimento de software realizados com o uso de Métodos Ágeis. Por outro lado, 52% dos que selecionaram o enfoque moderado ou clássico de gerenciamento de projetos indicaram que o Gerenciamento Ágil de Projetos é mais adequado para projetos desta natureza. Ou seja, este resultado sugere que a maioria dos respondentes destas questões, independente do grupo a que pertença, tende a concordar que o Gerenciamento Ágil de Projetos é mais indicado para o desenvolvimento de software realizado com o uso de Métodos Ágeis, dando maior sustentação às ponderações feitas quando da análise descritiva do pareamento Q7 *versus* Q8.

Um ponto que merece destaque, ainda que a associação Q6 *versus* Q8 (ver Tabela 75 do Anexo B) não se mostre estatisticamente significativa, é apresentado a seguir. Cerca de 91% dos respondentes que optaram pelo Gerenciamento Ágil de Projetos indicaram como critérios predominantes para a avaliação do sucesso do desenvolvimento de software, a satisfação das expectativas dos principais interessados no projeto (73%) e o atendimento aos requisitos do projeto (18%). Praticamente não houve menção aos critérios relacionados ao cumprimento dos três objetivos principais do projeto, ou seja, ao cumprimento do trinômio "prazo-custo-qualidade", base da definição tradicional do sucesso (MEREDITH; MANTEL, 2000; KERZNER, 2002). Esse resultado ratifica a colocação feita durante a revisão bibliográfica, de que está ocorrendo uma mudança na maneira de se mensurar o sucesso dos projetos e que, de forma geral, o Gerenciamento Ágil de Projetos leva em consideração esta nova abordagem (COHEN; GRAHAM, 2002; THOMSETT, 2002). Vale mencionar que esta mudança também

pode ser vista entre aqueles que selecionaram os enfoques moderado ou clássico de gerenciamento de projetos, porém com menor ênfase. Cerca de 63% destes respondentes selecionaram o atendimento às expectativas dos principais interessados e o atendimento aos requisitos do projeto como principais critérios para a avaliação do desempenho do desenvolvimento de software. Para este grupo, foi verificada ainda a presença dos indicadores tradicionais de cumprimento dos objetivos de prazo e de qualidade.

Na mesma situação, cabe analisar o pareamento Q15 *versus* Q7 (ver Tabela 51 do Anexo B), relativa à influência da colaboração entre as diferentes equipes e o desempenho do projeto de desenvolvimento de software com o uso de Métodos Ágeis. Apesar desta associação não se mostrar estatisticamente significativa, ela é analisada por ser um dos pontos mais críticos para a adoção dos Métodos Ágeis e do Gerenciamento Ágil de Projetos, segundo afirmam Highsmith (2004), Turk *et al* (2005) e Nerur *et al* (2005). A análise puramente descritiva dos dados indicou que 88% dos respondentes que qualificaram seu projeto como bem-sucedido concordaram que houve uma boa colaboração entre as equipes envolvidas no projeto. Quanto aos que classificaram seu projeto como de "Sucesso Parcial", tem-se que 76% também o fizeram. Estes resultados sugerem que realmente a colaboração entre as equipes é um fator primordial para a operacionalização de um projeto de desenvolvimento de software com o uso de um Método Ágil.

Este fato se torna mais nítido entre aqueles que optaram pelo Gerenciamento Ágil de Projetos, como mostra a Tabela 83 (ver Anexo B). Este grupo concordou com uninimidade que houve uma boa colaboração entre as equipes envolvidas no projeto, o que valida a colocação de Highsmith (2004) e de Turk *et al* (2005) de que este é um dos pressupostos básicos para a adoção de um Método Ágil e, conseqüentemente, para o Gerenciamento Ágil de Projetos.

Com estas considerações, encerra-se a etapa de análise descritiva desta pesquisa sob um enfoque uni e bivariado e parte-se para a análise multivariada dos dados, visando à redução de dimensionalidade do problema. Esta etapa é realizada por meio da análise discriminante, sendo os resultados obtidos validados pela aplicação da regressão logística.

# 4.2 Análise das Variáveis Relacionadas ao Desempenho dos Projetos de Desenvolvimento de Software

Neste tópico são apresentados os resultados dos modelos de análise discriminante e de regressão logística ajustados aos dados, relacionados ao desempenho dos projetos de desenvolvimento de software conduzidos com o uso de Métodos Ágeis. Conforme exposto anteriormente, a análise discriminante é utilizada para classificar objetos ou indivíduos em um número pré-estabelecido de grupos, a partir de medições de diversas variáveis. É utilizada também para se determinar uma equação discriminante capaz de predizer o resultado de uma nova observação, com uma chance de acerto específica. O modelo de regressão logística, por sua vez, é utilizado nesta pesquisa exclusivamente como forma de validar a estratégia de modelagem e os resultados encontrados na análise discriminante.

#### 4.2.1 Resultados da Análise Discriminante

Nesta etapa da pesquisa, a análise discriminante é conduzida de forma a promover uma separação entre as duas categorias de desempenho do projeto de desenvolvimento de software – "Sucesso" e "Sucesso Parcial". Esta discriminação em apenas dois grupos, apesar da variável dependente relativa ao desempenho do projeto prever três classificações, deve-se ao fato de nenhum respondente ter qualificado seu projeto como de "insucesso", como mostra a Tabela 32 (ver Anexo B) .

A alocação entre os grupos é determinada, então, pela resposta dada à Q7 do instrumento de pesquisa (ver Anexo A), relativa ao desempenho do projeto. As variáveis explicativas (preditoras) usadas para discriminar os indivíduos correspondem às questões Q13 a Q34 do instrumento de pesquisa, referentes às técnicas e características dos Métodos Ágeis.

A possibilidade de se ter 22 variáveis preditoras, restringindo a seleção a no máximo quatro questões com o melhor poder de discriminação entre os dois grupos, levou à adoção da estratégia de seleção de busca exaustiva. Sendo assim, foram ajustados todos os possíveis modelos de análise discriminante com uma até quatro variáveis explicativas, chegando a um total de 9.108 modelos, resultantes das várias combinações possíveis, a saber:

Para cada modelo foram calculados a proporção de acertos e o número de "falsos positivos". Por "falso positivo" entende-se quantas vezes o desempenho do projeto foi classificado pelo modelo como "Sucesso", mas na verdade era "Sucesso Parcial".

Dentre os 9.108 modelos de análise discriminante gerados, foram escolhidos os 100 melhores, considerando-se a proporção de acertos. Os dez melhores modelos de predição para a Q7, com as respectivas proporções de acerto, número de falsos positivos e tamanho amostral, podem ser visualizados na Tabela 103 (ver Anexo C).

A partir da análise da lista reduzida dos 100 melhores modelos, foram selecionadas três questões – Q25, Q30 e Q32 – presentes em mais de 60% dos modelos ajustados. É importante explicar que o critério de seleção das questões adotado foi a presença em vários modelos e não a simples presença no melhor modelo, uma vez que vários modelos ajustados apresentaram o mesmo valor preditivo.

Ao se avaliar a distribuição da proporção de acertos para os 9.108 modelos gerados, por meio de gráficos de *boxplot*, percebe-se um sensível aumento nesta proporção de acerto entre os modelos que incorporam as três questões indicadas acima. Os modelos que não incluem as questões Q25, Q30 e Q32 têm uma chance de acerto média em torno de 72%. Já os modelos que incluem estas questões têm uma chance de acerto média de cerca de 79%, conforme mostra o Gráfico 2 (ver Anexo C). Pode-se dizer então que, considerando a amostra disponível, a presença das referidas questões tende a melhorar a proporção de acerto dos modelos de análise discriminante gerados, oferecendo sustentação à escolha realizada. A chance de acerto do melhor modelo ajustado é de 84% (ver Tabela 103 – Anexo C).

A equação discriminante final para o desempenho dos projetos de desenvolvimento de software realizado com o uso de Métodos Ágeis, considerando-se a amostra desta pesquisa, é:

$$D = 0.572*Q30 - 0.616*Q32 + 1.065*Q25.$$

A distribuição de acertos para o modelo proposto acima pode ser visualizada na Tabela 104 do Anexo C. Analisando os dados desta tabela percebe-se que o modelo de análise discriminante final apresenta uma tendência otimista: quando se trata de "sucesso" há um erro de 5%, ou seja, apenas 5% dos casos originais de sucesso são classificados pelo modelo como "sucesso parcial", entretanto, 53% dos casos que originalmente são considerados como "sucesso parcial", são classificados como "sucesso".

Enfocando as variáveis que compõem o modelo de análise discriminante selecionado, percebe-se que as questões 25 e 30 já haviam sido destacadas durante a etapa de estatística descritiva. Já a questão 32, cujo enunciado é apresentado abaixo, aparece pela primeira vez:

 Q32: Houve um clima de descontração, alegria e diversão durante o desenvolvimento.

A Q25 é a questão de maior importância no modelo de análise discriminante para o desempenho do projeto, o que pode ser percebido pelo seu maior coeficiente ou peso na equação (1,065). Além disso, a Q25 está presente nos 100 principais modelos preditivos gerados relacionados ao desempenho dos projetos de desenvolvimento de software realizados com o uso de Métodos Ágeis. Seu forte poder de influência no desempenho destes projetos já havia sido percebido e discutido no tópico 4.1.4, uma vez que o pareamento Q25 versus Q7 apresentou o nível de associação mais significativo, dentre todos os pareamentos possíveis com a Q7. Deve-se ressaltar ainda, que o sinal positivo do coeficiente vinculado à Q25 indica que quanto maior a motivação da equipe, melhor o desempenho do projeto. Sendo assim, a motivação da equipe de desenvolvimento pode ser considerada o principal fator crítico de sucesso dos projetos de desenvolvimento de software conduzido com o uso de Métodos Ágeis.

A Q30, por sua vez, presente em 60% dos melhores modelos de análise discriminante, também já foi discutida anteriormente, sendo que seu nível de associação com a Q7 foi o segundo mais significativo, perdendo somente para a Q25. Dado o sinal positivo do coeficiente relacionado a esta variável (0,572), pode-se dizer que a familiaridade e o emprego das práticas relativas os Métodos Ágeis no desenvolvimento diário de software só vem a

aumentar a chance de sucesso do projeto em questão. Este é considerado, então, o segundo fator crítico de sucesso para os projetos de desenvolvimento de software realizado com o uso de Métodos Ágeis.

Com relação à Q32, há um ponto interessante a ser comentado. Apesar da análise descritiva da associação Q32 *versus* Q7 (ver Tabela 68 – Anexo B) indicar que 85% dos respondentes que qualificaram seu projeto como bem-sucedido concordaram que houve um clima de descontração, alegria e diversão durante o desenvolvimento do software, esta questão, ao ser inserida na equação de análise discriminante, apresenta um coeficiente com sinal negativo (-0,616). Isto parece indicar que na presença das demais variáveis Q25 e Q30, a Q32 funcione como um moderador para determinar o desempenho final do projeto. Destaca-se também, que a associação Q32 *versus* Q7 não se mostrou estatisticamente significativa.

Aparentemente, os resultados encontrados através da análise discriminante se mostraram bastante alinhados aos resultados da estatística descritiva e à teoria apresentada, uma vez que as duas principais variáveis – motivação da equipe e emprego das técnicas relativas aos Métodos Ágeis (representadas pela Q25 e Q30) – foram destacadas em ambas as análises, além de serem comentadas nos trabalhos de vários autores, como Cockburn e Highsmith (2001b), Reifer (2002), Lazarevic (2003), Cohn e Ford (2003), Cohen *et al* (2003), Gawlas (2004), Bonato (2004), Highsmith (2004) e Nerur *et al* (2005).

É importante mencionar ainda que outros modelos, integrantes da lista dos dez melhores modelos de análise discriminante gerados, apresentados na Tabela 103 (ver Anexo C), utilizam além das questões Q25 e Q30, outras questões que também foram discutidas na etapa de análise descritiva como, por exemplo, as questões Q31 e Q32, ou mesmo as questões Q14 e Q34, que apresentaram associação com o enfoque de gerenciamento de projetos.

# 4.2.2 Resultados da Regressão Logística

Como forma de validar os resultados encontrados, procedeu-se à análise por regressão logística. Dada a limitação do grau de liberdade que impossibilitou o ajuste do modelo de regressão logística com as 22 potenciais variáveis preditoras simultaneamente, foi selecionado

um subconjunto destas variáveis. Como critério para esta pré-seleção foram adotados os níveis de significância obtidos nos perfis marginais na etapa de análise descritiva (ver Tabelas 39 a 70 – Anexo B). Sendo assim, foram selecionadas as seguintes questões: Q13, Q16, Q23, Q25, Q27, Q31, Q32, Q33 e Q34.

Em seguida, foi conduzido um procedimento *stepwise* de seleção de modelos pelo Critério de Informação de Akaike (AKAIKE, 1974). Ao final desse procedimento, chegou-se ao modelo final com as variáveis Q25, Q31 e Q32, cujas estimativas encontram-se indicadas na Tabela 107 (ver Anexo D). Esse modelo gerado de forma totalmente independente se mostrou altamente concordante com o resultado obtido por busca em profundidade, através de modelos de análise discriminante. Cabe lembrar que a Q31 diz respeito a uma técnica específica dos Métodos Ágeis (revisão da codificação por outro programador), podendo ser inserida no contexto da Q30.

Assim, esta modelagem alternativa complementa a estratégia anterior, dando sustentação aos resultados obtidos nessa etapa da pesquisa.

# 4.3 Análise das Variáveis Relacionadas ao Enfoque de Gerenciamento de Projetos

As duas abordagens – análise discriminante e regressão logística – foram repetidas para a avaliar as variáveis relacionadas com o enfoque de gerenciamento de projetos. Os resultados são discutidos a seguir.

#### 4.3.1 Resultados da Análise Discriminante

Neste momento, a análise discriminante é conduzida de forma a promover uma separação entre as duas categorias relativas aos enfoques de gerenciamento de projetos – "Ágil" e "Moderado - Clássico". Esta mesma separação já foi utilizada na etapa de análise descritiva e foi mantida propositalmente para permitir a comparação de resultados e para evitar eventuais problemas com o baixo número de observações em cada categoria. A alocação entre os grupos é determinada, então, pela resposta dada à Q8 do instrumento de pesquisa (ver Anexo A),

relativa ao grau de combinação dos enfoques de gerenciamento de projetos. Para esta questão, as respostas (1) e (2) correspondem à categoria denominada "Ágil" e as respostas (3), (4) e (5) à categoria "Moderado – Clássico". Novamente, as variáveis explicativas (preditoras) usadas para discriminar os indivíduos correspondem às questões Q13 a Q34 do instrumento de pesquisa, referentes às técnicas e características dos Métodos Ágeis.

Similarmente à análise feita para a Q7, foram gerados 9.108 modelos com até quatro variáveis, sendo separados os 100 melhores modelos, de acordo com a proporção de acerto. Dentre esses 100 melhores modelos, 99% deles contêm a Q14 e mais de 50% deles incluem a Q33 e a Q34. O restante das questões aparece em menos de 20% dos 100 melhores modelos. Sendo assim, estas três questões foram selecionadas para integrar o modelo final.

Fazendo uma comparação da distribuição da proporção de acerto para os 9.108 modelos de análise discriminante, de acordo com a presença ou não das três questões selecionadas – Q14, Q33 e Q34, observa-se um aumento significativo na chance de acerto nos modelos que consideram estas questões (Gráfico 3 – Anexo C). Os modelos que não contemplam a Q14, a Q33 e a Q34 têm proporção de acerto média de aproximadamente 77%, enquanto aqueles que incluem tais questões têm sua proporção de acerto média na ordem de 92%. A lista dos dez melhores ajustes obtidos, com as respectivas proporções de acerto, número de falsos positivos e tamanho amostral considerado pode ser visualizada na Tabela 105 (ver Anexo C).

A equação discriminante final para este modelo relacionado ao enfoque de gerenciamento de projetos, considerando a amostra desta pesquisa, é:

$$D = -0.805*Q14 - 0.350*Q33 - 0.100*Q34.$$

É importante observar que as três questões – Q14, Q33 e Q34 – já foram destacadas como aquelas que apresentaram maior nível de significância quando associadas à Q8, durante a análise descritiva, sendo a relação entre estas questões e a Q8 discutida naquela oportunidade (tópico 4.1.4). Estas questões se referem, respectivamente, ao apoio da alta administração ao projeto, ao enfoque de gerenciamento de projetos mais adequado para o desenvolvimento de software realizado com o uso de Métodos Ágeis e à importância do uso de um desses métodos para o sucesso do projeto.

Cabe complementar para este modelo, a análise do sinal dos coeficientes. O sinal negativo presente nos coeficientes de todas as variáveis indica que quanto maiores os associados à questões Q14, Q33 e Q34, menor o valor predito da Q8, ou seja, quanto maiores estes valores, maior a tendência à adoção do Gerenciamento Ágil de Projetos. Esta inversão de sinal deve-se exclusivamente à codificação utilizada para a Q8.

Com relação à tabela de acertos, quando se trata do enfoque de gerenciamento de projetos "Moderado – Clássico" houve 100% de acerto na predição. No entanto, 30% das respostas referentes ao Gerenciamento Ágil de Projetos foram erroneamente classificadas como do grupo "Moderado – Clássico". A elevada taxa de acertos relativa ao grupo "Moderado – Clássico" é explicada pelo efeito da categorização adotada, que agregou as respostas "Moderado", "Significativo" e "Muito Significativo" da Q8, nesta categoria.

Assim como na análise discriminante anterior, o modelo discriminante aqui gerado é muito aderente aos resultados da etapa descritiva e, por conseguinte, à teoria apresentada no Capítulo 2 – Revisão Bibliográfica.

# 4.3.2 Resultados da Regressão Logística

Similarmente ao que foi exposto anteriormente para a Q7, procedeu-se a outra abordagem de análise por meio da regressão logística e do procedimento de seleção de modelos *stepwise*, tendo por base as respostas à Q8 e, como critério de pré-seleção das variáveis, os níveis de significância obtidos nos perfis marginais na análise descritiva. Por meio desse procedimento foram encontradas as seguintes variáveis integrantes do modelo final: Q14, Q16, Q27, Q31, Q32 e Q34.

A utilização deste tipo de modelagem foi menos poderosa na identificação final das questões, sendo o ajuste prejudicado pelo pequeno tamanho amostral, limitado neste caso a 36, haja vista que se considerou somente as observações completas no modelo inicial.

Em face deste cenário, não são apresentadas as estimativas dos parâmetros para o modelo de regressão logística. Todavia, pode-se observar que dentre as variáveis do modelo final,

ficaram as três questões selecionadas na análise discriminante realizada anteriormente – Q14, Q33 e Q34 – que de certa forma demonstra uma concordância entre os resultados.

## 4.4 Considerações Finais e Limitações Observadas

Por meio das análises propostas – análise descritiva, análise discriminante e regressão logística – foi possível identificar um conjunto de questões que têm o maior poder de previsão para as respostas às questões Q7 e Q8. Contudo, deve-se deixar claro algumas limitações observadas.

Primeiramente, na maioria dos problemas que envolve a seleção de variáveis, o procedimento usualmente empregado é a divisão da amostra total em dois conjuntos de trabalho: um "conjunto de aprendizado" utilizado para selecionar as variáveis e um "conjunto de validação" usado para verificar os resultado obtidos, comparando-os com os primeiros (MILLER, 1984). Devido ao tamanho da amostra, este procedimento não pôde ser aplicado na presente pesquisa, podendo haver um certo viés na seleção dos modelos e na identificação das variáveis explicativas.

Em segundo lugar, a utilização de variáveis explicativas ordinais na análise discriminante também pode acarretar uma certa perda de poder no ajuste dos modelos, apesar de esta aplicação ser defendida por Conover (1999).

Outro ponto a ser ressaltado é que por se tratar de um assunto relativamente novo, os respondentes poderiam não ter conseguido reconhecer ou entender, de forma imediata, o significado do termo *ágil*. Mas ao se avaliar os resultados da pesquisa, percebe-se que este reconhecimento se deu pela identificação dos principais Métodos Ágeis (XP, *Scrum*, FDD, entre outros). Vale mencionar que este problema foi minimizado ao se optar pela amostragem intencional por julgamento, selecionando um grupo específico de respondentes com experiência e/ou interesse no assunto.

Por outro lado, este tipo de amostragem selecionado pode trazer um viés aos resultados do estudo, sendo esta considerada a maior limitação da presente pesquisa.

Para minimizar a possibilidade de viés na seleção de modelos ou na escolha das variáveis e eventuais efeitos da perda de poder no ajuste dos modelos, decidiu-se pela aplicação de duas abordagens alternativas – análise discriminante e regressão logística – que têm suposições completamente diferentes. Estas duas abordagens chegaram em conjuntos de variáveis equivalentes, indicando uma concordância entre as modelagens e aportando sustentação aos resultados obtidos. Desta forma, há uma boa indicação da validade das conclusões que podem ser tiradas em relação às variáveis que têm maior influência nas questões de interesse. Estas conclusões são apresentadas no próximo capítulo deste trabalho.

# 5 CONCLUSÕES

A dissertação teve como objetivo principal responder o questionamento sobre qual o enfoque de gerenciamento de projetos – ágil ou clássico – é mais apropriado para o desenvolvimento de software conduzido com o uso de um Método Ágil. Como objetivo secundário, buscou investigar os fatores críticos de sucesso dos projetos desta natureza.

A análise restringiu-se aos dados coletados junto a participantes de grupos especializados na troca de experiência em desenvolvimento de software realizado com o uso de Métodos Ágeis, havendo portanto, maior chance de viés nos resultados do estudo. O fato de não se ter acesso à população-alvo e se trabalhar com uma amostra extraída da população-disponível foi considerada a maior limitação da presente pesquisa. Sendo assim, dado o tipo de amostragem empregado, as conclusões apresentadas ficaram restritas ao âmbito desta pesquisa, não podendo ser automaticamente generalizadas ou inferidas para situações distintas das abordadas.

Por meio desta pesquisa, pôde-se concluir que tanto os Métodos Ágeis como o Gerenciamento Ágil de Projetos, apesar de recentes, fazem parte da realidade brasileira e são utilizados por profissionais de várias organizações.

Quanto à resposta à pergunta-problema, considerando-se os dados disponíveis, não houve evidência amostral para encontrar uma associação estatisticamente significativa entre o desempenho de um projeto de desenvolvimento de software e o enfoque de gerenciamento de projetos adotado. Desta forma, não foi possível comprovar com rigor estatístico a existência de um enfoque de gerenciamento de projetos mais apropriado para o desenvolvimento de software realizado com o uso de Métodos Ágeis. No entanto, constatações importantes sobre esta associação, advindas da análise descritiva dos dados, puderam ser feitas.

Verificou-se que tanto o Gerenciamento Ágil de Projetos, quanto o Gerenciamento Clássico de Projetos, ou mesmo uma combinação entre eles, podem ser utilizados para gerenciar os desenvolvimentos de software realizados com o uso de Métodos Ágeis. O enfoque moderado,

que reúne práticas dos enfoques ágil e clássico de gerenciamento de projetos, foi o que recebeu a maior concentração de respostas, mas a pesquisa não permitiu a identificação do porquê desta concentração.

A possibilidade de adoção de qualquer um dos enfoques para gerenciar projetos desta natureza encontra-se totalmente alinhada ao referencial teórico, uma vez que não há uma posição única ou um consenso entre os autores sobre o enfoque de gerenciamento de projetos a ser adotado no desenvolvimento de software conduzido com o uso de Métodos Ágeis. Autores como Thomsett (2002), Highsmith (2004) e Chin (2004), defendem o uso do Gerenciamento Ágil de Projetos para o desenvolvimento de software (e em quaisquer outras iniciativas que envolvam certo grau de inovação), enquanto Paulk (2001) argumenta que não há incompatibilidade na aplicação do SW-CMM (que tem por base o Gerenciamento Clássico de Projetos) no desenvolvimento de software realizado com o emprego de Métodos Ágeis. Highsmith (op. cit.) e Chin (op. cit.) ainda admitem a possibilidade de combinação entre os enfoques ágil e clássico de gerenciamento de projetos em determinadas situações.

Os resultados da pesquisa indicaram que 91% dos respondentes que optaram pelo Gerenciamento Ágil de Projetos consideraram seus projetos bem-sucedidos e concordaram ser este, o enfoque mais apropriado para o gerenciamento de projetos de desenvolvimento de software conduzidos com o uso de Métodos Ágeis. Por outro lado, 64% dos respondentes que escolheram a combinação dos enfoques ágil e clássico ou a utilização exclusiva do Gerenciamento Clássico de Projetos qualificaram seus projetos como de sucesso, sendo que 52% deste mesmo grupo apontaram o Gerenciamento Ágil de Projetos como o enfoque mais indicado para as iniciativas desta natureza.

Apesar da associação entre o desempenho de um projeto de desenvolvimento de software e o enfoque de gerenciamento de projetos adotado não ter se mostrado estatisticamente significativa, os resultados da análise descritiva sugeriram que a maioria dos respondentes, independente do grupo a que pertençam, concordou que o Gerenciamento Ágil de Projetos é o mais apropriado para o desenvolvimento de software realizado com o uso de um Método Ágil.

Os projetos de desenvolvimento de software aqui analisados apresentaram um desempenho bastante superior aos resultados publicados em outras pesquisas. Cerca de 69% dos respondentes classificaram seus projetos de desenvolvimento de software como bem-

sucedidos, enquanto 31% qualificaram-nos como de sucesso parcial, não havendo nenhuma menção a projetos mal-sucedidos. As pesquisas até então divulgadas pelo STANDISH GROUP INTERNATIONAL (1999; 2001; 2003) e a versão parcial publicada em 2004 (STANDISH GROUP INTERNATIONAL, 2004) retrataram uma situação inversa, com cerca de 30% dos projetos classificados como de sucesso e 70% deles com algum problema de desempenho (projetos qualificados como de sucesso parcial ou de insucesso).

O melhor desempenho apresentado pelos projetos de desenvolvimento de software aqui pesquisados vem corroborar a posição dos defensores do *Manifesto para o Desenvolvimento* Ágil de Software (BECK et al, 2001) e de Highsmith (2004) e Chin (2004) de que os Métodos Ágeis e o Gerenciamento Ágil de Projetos surgiram como soluções promissoras, capazes de reverter os problemas de desempenho usualmente enfrentados pelos projetos desta natureza.

Há que se expor, porém, uma diferença notada no critério de mensuração do desempenho dos projetos nas pesquisas. Os estudos realizados pelo STANDISH GROUP INTERNATIONAL (1999; 2001; 2003) adotaram como indicador de desempenho de um projeto, o cumprimento dos objetivos de prazo, custo e qualidade. Nesta pesquisa, o principal critério identificado para a avaliação do desempenho dos projetos foi o atendimento às expectativas dos principais interessados no projeto, sendo esta escolha muito mais nítida entre aqueles que optaram pelo Gerenciamento Ágil de Projetos. Cohen e Graham (2002) e Thomsett (2002) mencionam que a seleção de um critério de mensuração de desempenho adequado pode influenciar diretamente o sucesso ou o alcance dos objetivos do projeto, uma vez que direciona o foco do gerenciamento e do controle. Talvez a diferença entre os critérios de mensuração de sucesso dos projetos analisados por estas pesquisas possa explicar, em parte, a grande diferença de desempenho encontrada. Este assunto pode se tornar objeto de uma pesquisa futura.

A absorção das solicitações de mudanças no decorrer do projeto e, principalmente, a adaptação tranquila dos requisitos do software às necessidades do cliente ou do negócio, surgiram como outra possível explicação para o melhor desempenho dos projetos analisados nesta pesquisa. Estas características estavam presentes com maior ênfase nos projetos classificados como bem-sucedidos e conduzidos segundo os princípios do Gerenciamento Ágil de Projetos.

Os resultados encontrados permitiram tirar conclusões quanto ao Método Ágil mais utilizado, às características principais dos projetos de desenvolvimento de software realizados com o uso dos Métodos Ágeis e às variáveis que influenciam o desempenho destes projetos, consideradas fatores críticos de sucesso. Foi possível traçar um paralelo destes resultados com os da pesquisa realizada por Lazarevic (2003), cujo conteúdo serviu como base para a estruturação deste trabalho.

Ratificando a posição de Cohen *et al* (2003), Lazarevic (2003) e Turk *et al* (2005), concluiuse que o Método Ágil de maior expressão é o *Extreme Programming* (XP). Outros métodos, como o *Scrum*, o *Feature Driven Development* (FDD), o *Adaptative Software Development* (ASD) e o *Lean Development* (LD) foram citados, porém com pequena expressão.

Com relação às características dos projetos de desenvolvimento de software pesquisados, pode-se dizer que, de forma geral:

- Eram projetos de pequena ou média duração;
- As equipes de desenvolvimento eram compostas por menos de dez integrantes;
- Os profissionais alocados ao projeto tinham boa qualificação técnica;
- Os projetos foram realizados de forma iterativa (baseada em incrementos de funcionalidade);
- Houve várias solicitações de alteração (adaptações) dos requisitos do software no decorrer do projeto;
- Houve um grande apoio da alta administração;
- Houve boa coordenação e colaboração entre as diferentes equipes envolvidas no projeto;
- As equipes de projeto encontravam-se bastante motivadas com o projeto, havendo um clima de descontração e alegria durante o desenvolvimento do software;
- As práticas dos Métodos Ágeis foram adotadas no dia-a-dia do projeto, apesar de não ter havido um treinamento formal da equipe de desenvolvimento.

Algumas destas características estavam presentes com maior ou menor intensidade, dependendo do enfoque de gerenciamento de projetos adotado.

Com relação aos fatores críticos de sucesso para os projetos de desenvolvimento de software realizados com o uso de Métodos Ágeis, concluiu-se que o fator principal é a motivação da equipe de desenvolvimento: quanto maior o grau de motivação dos integrantes da equipe, melhor o desempenho destes projetos. Este resultado valida as conclusões do Lazarevic (2003). Outros estudos já haviam mostrado a existência de uma relação entre o uso de Métodos Ágeis e a motivação da equipe, mas sem tratá-la como um fator crítico de sucesso (COHEN *et al*, 2003; COCKBURN; HIGHSMITH, 2001b; REIFER, 2002; BONATO, 2004). Concordando com Lazarevic (2003), este resultado não traz surpresas, valendo retomar a citação de Urban e Hauser (1993), de que na era da "economia baseada no conhecimento" as pessoas são consideradas o fator primordial para o sucesso de qualquer organização.

O outro fator crítico de sucesso que merece destaque é a familiaridade e o emprego das técnicas relativas aos Métodos Ágeis no dia-a-dia do projeto. Esta relação entre o domínio das técnicas dos Métodos Ágeis e o desempenho de um projeto de desenvolvimento de software, foi discutida por Bohem (2002), Cohen *et al* (2003), Fowler (2003) e Nerur *et al* (2005). Estes autores destacam ser fundamental que a equipe de desenvolvimento e os gerentes de projeto estejam a par desta nova abordagem, aplicando as técnicas no dia-a-dia do projeto, uma vez que os Métodos Ágeis valorizam e depositam elevado grau de confiança no conhecimento tácito e na capacidade de tomada de decisões de cada indivíduo.

Na pesquisa realizada por Lazarevic (2003), este fator não aparece de forma direta, mas sim indiretamente. O segundo fator crítico de sucesso identificado por este autor foi o grau de propriedade coletiva do código, que corresponde a uma das técnicas dos Métodos Ágeis. Desta forma, há novamente uma concordância entre os resultados das duas pesquisas. Cabe salientar que no presente estudo, o grau de propriedade coletiva do código aparece como uma das variáveis que têm associação estatisticamente significativa com o enfoque de gerenciamento de projetos, sendo sua presença mais marcante nos desenvolvimentos de software conduzidos segundo as práticas do Gerenciamento Ágil de Projetos.

Na análise dos fatores que influenciaram a adoção dos enfoques ágil ou clássico de gerenciamento de projetos, observou-se o papel preponderante do apoio da alta administração. Destacado por Cockburn e Highsmith (2001b) e Nerur *et al* (2005) como um fator crítico para a implantação dos Métodos Ágeis, é retomado por Highsmith (2004) no contexto do Gerenciamento Ágil de Projetos. Os resultados da presente pesquisa ratificaram as colocações

destes autores, uma vez que todos os respondentes que indicaram a utilização do Gerenciamento Ágil de Projetos concordaram que houve um forte apoio da alta administração em suas iniciativas de desenvolvimento de software. Este apoio de fato se faz necessário, uma vez que a adoção de um enfoque ágil de gerenciamento de projetos não acarreta apenas a substituição de processos e ferramentas, mas pressupõe uma mudança de cultura e da forma de gerenciamento e de controle dos projetos, além do estabelecimento de um ambiente de respeito e confiança mútua entre clientes e fornecedores, onde os indivíduos sejam valorizados e incentivados a tomar decisões (COCKBURN; HIGHSMITH, 2001b; HIGHSMITH, 2004, NERUR *et al.*, 2005).

Finalmente, por meio desta dissertação, foi possível consolidar um conjunto de conhecimento sobre o tema e prover evidências empíricas da realidade brasileira, contribuindo para a ampliação do conhecimento em Administração de Projetos. Espera-se também que os resultados alcançados sirvam para direcionar ou inspirar futuras pesquisas nesta temática.

Como recomendação aos trabalhos vindouros, sugere-se a manutenção do envio questionário eletrônico, porém com uma reestruturação da lógica de resposta, de forma a permitir que os respondentes sem experiência em Métodos Ágeis possam preencher a seção de qualificação do instrumento de pesquisa por completo, possibilitando a obtenção de informações mais detalhadas desta parcela da população. Indica-se também a redução do número de questões do instrumento de pesquisa.

Estudos adicionais podem ser conduzidos tendo como foco a investigação dos motivos que levam à adoção de um determinado enfoque de gerenciamento de projetos nas iniciativas de desenvolvimento de software com o uso de Métodos Ágeis, a identificação dos resultados e/ou benefícios da utilização do Gerenciamento Ágil de Projetos, ou mesmo, a avaliação da prontidão de uma organização para a adoção do Gerenciamento Ágil de Projetos. A identificação de critérios de mensuração de desempenho dos projetos de desenvolvimento de software pode ser tema de um futuro estudo.

Um bom incentivo para as pesquisas vindouras são as diversas iniciativas que estão surgindo no Brasil, visando ao aprimoramento e ao crescimento das exportações do setor de software. O gerenciamento de projetos de desenvolvimento de software assume, assim, importância

cada vez maior no meio acadêmico e pode contribuir de forma significativa para o crescimento do país.

# REFERÊNCIAS

ABRAHAMSSON, P et al. New directions on agile methods: a comparative analysis. In: Proceedings of the 25<sup>th</sup> International Conference on Software Engineering. [S.l.]. IEEE Software Society, 2003, p. 244-254.

AGILE ALLIANCE. *Manifesto for agile software development*. Disponível em <a href="http://www.agilemanifesto.org/">http://www.agilemanifesto.org/</a>>. Acesso em janeiro, 2005.

AGRESTI, A. Categorical data analysis. New Jersey: John Wiley & Sons, 2002, 2.ed.

AKAIKE, H. A new look at the statistical model identification. **IEEE Transactions on Automatic Control**, [S.l.], v. 19, p. 716-723, 1974.

ARCHIBALD, R. D. *Managing high-technology programs and projects*. 2nd ed. New York: John Wiley & Sons, 1992.

AUER, K.; MILLER, R. *Extreme programming applied*. Boston: Addison-Wesley, 2002.

AMBLER, S. W. Agile modeling: effective practices for extreme programming and the unified process. John Wiley & Sons, Inc., 2002a.

AMBLER, S.W. Lessons in agility from internet-based development. **IEEE Software**, [S.l.], v. 19, n. 2, 2002b, p. 66–73.

AMBLER, S.W. *When does(n't) agile modeling make sense*. Apr, 2002c. Disponível em <a href="http://www.agilemodeling.com/essays/whendoesAmWork.htm">http://www.agilemodeling.com/essays/whendoesAmWork.htm</a>. Acesso em julho de 2005.

AMBLER, S. W. *Quality in an agile world.* **Software Quality Professional**, [S.L.], v. 7, n. 4, sep. 2005.

BAKER, Bruce N. et al. Factors affecting project success. In: CLELAND, David I; KING, William R. Project Management Handbook. New York: Van Nostrand Reinhold, 1983, cap. 33, p. 669-685.

BASKERVILLE, R. et al. Is internet-speed software development different? **IEEE Software**, [S.l.], v. 20, n. 6, p. 70–77, 2003.

BATEMAN, T. S.; SNELL, S. A. Administração: construindo a vantagem competitiva. São Paulo: Atlas, 1997.

BEAUMONT, L.R. *Metrics: a Practical Example. In:* ROSENAU JR., M.D., GRIFFIN, A., CASTELLION, G.A., ANSCHUETZ, N.F. (Eds.) *The PDMA handbook of new product development.* New York: John Wiley & Sons, Inc., 1996. Part. IV, cap. 32, p. 463-485.

BECK, K. *Embrance Change with Extreme Programming*. **IEEE Computer Magazine**, [S.l.], Oct 1999, p. 70-77.

Extreme programming explained. Boston: Addison-Wesley, 200	. Extreme	programming	explained. Bosto	on: Addison	ı-Wesley.	200
--	-----------	-------------	------------------	-------------	-----------	-----

software.

BECK, Kent. *et al. Chrysler goes to "extremes"*. Oct, 1998. Disponível em <a href="http://www.xprogramming.com/publications/dc9810cs.pdf">http://www.xprogramming.com/publications/dc9810cs.pdf</a>>. Acesso em julho, 2005.

BECK, Kent *et al. Manifesto for agile software development*. Feb. 2001. Disponível em <a href="http://www.agilemanifesto.org/">http://www.agilemanifesto.org/</a>>. Acesso em janeiro, 2005.

BECK, Kent; FOWLER, M. Extreme programming applied. Boston: Addison-Wesley, 2001.

BECK, Kent; MEE, R. Enhancing brazilian software's global competitiveness. Jan, 2003. Disponível em: <a href="http://www.xispe.com.br/wiki/wiki.jsp?topic=EnhancingBrazilsSoftwareGlobalCompetitiveness">http://www.xispe.com.br/wiki/wiki.jsp?topic=EnhancingBrazilsSoftwareGlobalCompetitiveness</a>. Acesso em novembro, 2004.

BELL, Martin. 'Learning' and the Accumulation of Industrial Technological Capacity in Developing Countries. In: FRANSMAN, M., KING, K. Technological capability in the third world, New York: St Martin's Press, p. 187-209, 1984.

BEST, J.W. Como investigar en educación. 2.ed. Madrid: Morata, 1972, cap. 7.

BOGER, M. et al. Extreme modeling. In: SUCCI, G.; Marchesi, M. (eds). Extreme programming examined. Boston: Addison-Wesley, 2001.

BOHEM, Barry. *Get ready for agile methods, with care*. **IEEE Computer Magazine**, [S.l.], Jan. 2002, p. 64-69.

BOHEM, Barry.; TURNER, Richard. Balancing discipline and agility: evaluating and integration plan-driven methods. In: Proceedings of the 26<sup>th</sup> Conference on Agile Development. IEEE COMPUTER SOCIETY, [S.l.], May 2004, p. 718-719.

BLOCK, Thomas R., FRAME, J. Davidson. *The project office, a key to managing projects effectively*. New York: Crisp Publications, 1998.

BONATO, A. S. F. **Uma experiência de aplicação do processo** *Extreme Programming* **em pequenos projetos**. São Paulo, 2004. Dissertação (Mestrado em Engenharia) – Programa de Pós-Graduação em Engenharia, Escola Politécnica da Universidade de São Paulo.

BOOCH, G. *Developing the future*. **Communications of the ACM**. ACM Press, [S.l.], v. 44, n. 3, p. 118–121, 2001.

BRASIL. Ministério da Ciência e Tecnologia. Secretaria de Política de Informática. **Levantamento do universo de Associadas Softex**. Brasília - DF, 2001a.

Ministéri	o da Ciê	ncia e Tecn	ologia. Sec	retaria de Po	olítica de Info	ormática. Oual	idade
e Produtividade no			_				
Ministér	io da C	iência e T	ecnologia.	Tecnologia	de Inforn	nação: Progra	amas
Prioritários em <a href="http://www.mct.g">http://www.mct.g</a>	Tecn	ologia de	Inform	<b>ação</b> . Bras	sília, 2002.	Disponível	
Ministéri	o da Ciêr	ncia e Tecno	ologia. Agê	encia Brasil.	Governo qu	er exportar 2	bi de

2005a.

DF.

Brasília,

Disponível

em

<a href="http://www.mct.gov.br/Temas/info/Imprensa/Noticias_5/Software_5.htm">http://www.mct.gov.br/Temas/info/Imprensa/Noticias_5/Software_5.htm</a> Acesso em agosto, 2005.
Ministério da Ciência e Tecnologia. <b>Qualidade e produtividade no setor de software brasileiro</b> . Brasília, DF 2005b. Disponível em <a href="http://www.mct.gov.br/Temas/info/Dsi/Quali2004/formulario/">http://www.mct.gov.br/Temas/info/Dsi/Quali2004/formulario/</a> >. Acesso em agosto, 2005.
BRYMAN, A. <i>Research methods and organizational studies</i> . London: Routledge, 1995.
CARVALHO, M. M. et al. Information technology project management to achieve efficiency in Brazilian Companies. In: KAMEL, Sherif. (Org.). Managing Globally with Information Technology. Hershey, 2003, p. 260-171.
CHARVAT, J. <i>Project management methodologies:</i> selecting, implementing and supporting methodologies and processes for projects. John Wiley & Sons, 2003. Disponível em <a href="http://pmi.books24x7.com/toc.asp?bookid=5400">http://pmi.books24x7.com/toc.asp?bookid=5400</a> >. Acesso em janeiro, 2005.
CHIN, Gary. Agile project management: how to succeed in the face of changing project requirements. NY: Amacon, 2004.
CLARK, K. B., FUJIMOTO T. <i>Product development performance:</i> strategy, organization, and management in the world auto industry. Boston, Mass.: Harvard Business School Press, 1991.
CLARK, K. B., WHEELWRIGHT S. C. <i>The aggregate project plan. In:</i> CLARK, K. B., WHEELWRIGHT S. C. <i>Managing new product and process development:</i> text and cases. New York: Maxwell Macmillan International, 1993a, p. 233-289.
CLARK, K. B., WHEELWRIGHT S. C. Learning from development projects. In: CLARK, K. B., WHEELWRIGHT S. C. <i>Managing new product and process development:</i> text and cases. New York: Maxwell Macmillan International, 1993b, p. 731-760.
CLELAND, David I. <i>Project management: strategic design and implementation</i> . 2nd ed. Boston: McGraw-Hill, 1994.
COCKBURN, A. <i>Crystal clear:</i> a human-powered methodology for small teams. Boston: Addison-Wesley, 2004.
Agile software development. Boston: Addison-Wesley, 2001.
Learning from agile software development – part one. Crosstalk, The Journal of

**Defense Software Engineering**, [S.l.], October 2002.

COCKBURN, A.; HIGHSMITH, J. *Agile software development: the business of inovation*. **IEEE Computer Magazine**, [S.l.], p. 131-133, sep 2001a.

\_\_\_\_\_. *Agile software development: the people factor.* **IEEE Computer Magazine**, [S.l.], p. 131-133, nov 2001b.

COHEN, David et al. Agile software development: a DACS state of art report. NY: Data Analysis Center for Software - Fraunhouser Center for Experimental Software Engineering

Maryland and The University of Maryland, 2003. Disponível em <a href="http://www.thedacs.com/techs/agile/">http://www.thedacs.com/techs/agile/</a>. Acesso em abril, 2005.

COHEN, Dennis. J.; GRAHAM, Robert. J. **Gestão de projetos:** MBA executivo. Trad. Afonso Celso da Cunha Serra. Rio de Janeiro: Campos, 2002.

COHN, Martin. *The scrum development process*. Disponível em <www.mountaingoatsoftware.com/scrum> Acesso em julho, 2005.

COHN, Martin; FORD, Doris. *Introducing an agile process to an organization*. **IEEE Computer Magazine**, June 2003, [S.l.], p. 74-78.

CONOVER, W. J. *Practical nonparametric statistics*. New York: John Wiley & Sons, 3.ed., 1999.

COOPER, R. G. A process model for industrial new product development. **IEEE Transactions on Engineering Management**, [S.l.], v.EM-30, n.1, p. 2-11, feb. 1983.

COOPER, Donald R. SCHINDLER, Pamela S. **Métodos de pesquisa em administração**; trad. Luciana de Oliveira Rocha. 7ed - Porto Alegre: Bookman, 2003.

COPAS, J. B; LONG, T. Estimating the residual variance in orthogonal regression with variable selection. **The Statistician**, v. 40 (1), p. 51-59, 1991.

CRESWELL, J.W. *Research design: qualitative & quantitative approaches*. United States of America: Sage, 1994.

CROW, K. *Benchmarking best practices to improve product development*. Disponível em: <a href="http://www.soce.org/papers/crow-bench/crow-bench.htm">http://www.soce.org/papers/crow-bench/crow-bench.htm</a>> Acesso em: 05/04/2003

DEMING, W. E. **Qualidade:** a revolução da administração. Rio de Janeiro: Marques Saraiva, 1990.

DINSMORE, P. C. *The AMA handbook of project management*. New York: AMACON, 1993.

DINSMORE, P. C.; NETO, F. H. S. **Gerenciamento de projetos:** como gerenciar seu projeto com qualidade, dentro do prazo e custos previstos. Rio de Janeiro: Qualitymark, 2004.

DOWNEY, H. K.; IRELAND, R. D. *Quantitative versus qualitative: the case of environmental assessment in organizational.* **Science Quarterly**, [S.l.], vol. 24, no. 4, December 1979, p. 630-637.

DRAPER, N. R.; SMITH, H. Applied regression analysis. New York: John Wiley, 2.ed., 1981.

DRUKER, P. F. Managing for business effectiveness. **Harvard Business Review**, [S.l.], v. 41, n. 3, May/June, 1963.

DUFFY, M. E. Methodological triangulation: a vehicle for merging quantitative and qualitative research methods. **Journal of Nursing Scholarship**, v.19, n.3, 1987.

EXTREME PROGRAMMING BRASIL 2002. **Programação**. Disponível em <a href="http://www.xispe.com.br/evento2002/descricao.html">http://www.xispe.com.br/evento2002/descricao.html</a>>. Acesso em setembro, 2005.

FERREIRA, A. A., REIS, A. C. F., PEREIRA, M. I. **Gestão empresarial:** de Taylor aos nossos dias. São Paulo: Thomson, 2002, p. 146-164.

FOWLER, Martin. *The new methodology*. April, 2003. Disponível em <a href="http://www.martinfowler.com/articles/newMethodology.html">http://www.martinfowler.com/articles/newMethodology.html</a>>. Acesso em julho, 2005.

FORSBERG, K. et al. Visualizing project management: a model for business and technical success. John Wiley & Sons, Inc, 1996.

GAWLAS, J. Mission critical development with XP & agile process: common code ownership and lots of testing. **Dr. Dobb's Journal**, [S.l.], p. 21-24, Jan. 2004.

GLASS, Robert. Extreme programming: the good, the bad and the bottom line. **IEEE Software**, [S.1.], Nov. 2001, p. 111-112.

GRAY, F. C, LARSON, E. W. *Project management:* the managerial process. McGraw-Hill Irwin, 2 ed., 2003.

GIGA INFORMATION GROUP. *Lessons learned practicing agile development*. Boston: Giga Group, Apr. 2002.

GIL, A. Como elaborar projetos de pesquisa. São Paulo: Atlas, 1988.

GLAZER, H. Dispelling the process myth: having a process does not mean sacrificing agility or criativity. Crosstalk, The Journal of Defense Software Engineering, [S.1.], Nov. 2001.

GODOY, A. S. *Introdução à pesquisa qualitativa e suas possibilidades*. **Revista de Administração de Empresas**, v. 35, n. 2, Mar./Abr., 1995.

HAMEL, G., PRAHALAD, C. K. Competindo pelo futuro: estratégias inovadoras para se obter controle de seu setor e criar mecanismos de amanhã. Rio de Janeiro: Campos, 1995.

HIGHSMITH, Jim. Adaptative management: patterns for the e-business era. Cutter IT Journal, [S.l.], v. XII, n. 9, sep. 1999.

HIGHSMITH, Jim. Agile software development ecosystems. Boston: Addison-Wesley, 2002.

\_\_\_\_\_. Agile project management: creating innovative products. Boston: Addison-Wesley, 2004.

HIGHSMTIH, Jim *et al. Extreme programming: e-business application delivery.* Feb. 2002. Disponível em <a href="http://www.cutter.com/freestuff/ead0002.pdf">http://www.cutter.com/freestuff/ead0002.pdf</a>>. Acesso em julho, 2004.

INTERNATIONAL DATA CORPORATION – IDC. *Directions* **2004**. [S.l.]: 2004. Disponível em <www.idc.com>. Acesso em dezembro, 2004.

ISAAC, Stephen. *Handbook in research and evaluation*. San Diego, California: Edits Publishers, 1976.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION – ISO. *ISO 10006 – Quality management – guidelines to quality in project management*, 1997.

JEFFRIES, R. et al. Extreme programming installed. Boston: Addison-Wesley, 2001.

JIANG, J. J. et al. Ranking of system implementation success factors. Project Management Journal, [S.l.], December, 1996.

JOHNSON, J. H. *Micro projects causes constant changes*. Dec, 2002. Disponível em <a href="http://www.agilealliance.org/articles/articles/Chapter30-Johnson.pdf">http://www.agilealliance.org/articles/articles/Chapter30-Johnson.pdf</a>>. Acesso em agosto, 2005.

JOHNSON, R. A.; WICHERN, D. W. *Applied multivariate statistical analysis*. New Jersey: Prentice Hall, 2.ed., 1982.

JURAN, J. M.; GRYNA, F. M. Controle da qualidade – *handbook*: conceitos, políticas e filosofia da qualidade. São Paulo: Makron, McGraw-Hill, 1991

JURIC, R. Extreme Programming and its Development Practices. In: Proceedings of 22nd international conference on information technology interfaces. 2002, p. 97–104.

KERZNER, Harold. *Project Management:* a systems approach to planning, scheduling and controlling. New York: Van Nostrand Reinhold, 1992.

\_\_\_\_\_. **Gestão de Projetos:** as melhores práticas. Trad. Marco Antonio Viana Borges, Marcelo Klippel e Gustavo Severo de Borba. Porto Alegre: Bookman, 2002.

KILLING, R. **Gestão de Projetos:** uma abordagem global. Trad. Cid Knipel Moreira. São Paulo: Saraiva, 2002.

KUDO, A.; TARUMI, T. An algorithm related to all possible regression and discriminant analysis. **Journal of the Japan Statistical Society**, v. 4, 47-56, 1975.

LAITINEN, M. et al. Thinking objectively: the problem with scalability. Communications of the ACM. ACM Press, [S.l.], v. 43, n. 9, p. 105–107, 2000.

LARSON, Carl E.; LAFASTO, Frank M. J. Teamwork: What must go right, what can go wrong. Newberry Park, CA: Sage, 1989

LAURINDO, F. J. B., PESSÔA, M. S. P. Sistemas Integrados de Gestão. In NETO, A. (org) **Manufatura classe mundial: conceitos, estratégias, aplicações.** São Paulo: Atlas, 2001, p. 114-130.

LAZAREVIC, George. *An exploratory study of the new product development utilized by software companies using agile product development approach.* Oct. 2003. Disponível em <a href="http://www.agilealliance.org/articles/articles/agileOct.pdf">http://www.agilealliance.org/articles/articles/agileOct.pdf</a>>. Acesso em agosto, 2005.

LEWIS, J. P. Fundamentals of project management. New York: Amacon, 1997.
<b>Metodologia científica</b> . São Paulo: Atlas, 1988.
<i>The project manager's desk reference</i> , 2. ed., Boston : MacGraw-Hill, 2000.

LÜDKE, M.; ANDRÉ, M.E.D.A. **Pesquisa em educação: abordagens qualitativas**. São Paulo: Pedagógica universitária, 2 ed., 1988.

MARCONI, M. A.; LAKATOS, E.M. **Fundamentos de metodologia científica**. São Paulo: Atlas, 6 ed., 2005.

MANNING, P. K. *Metaphors of the field: varieties of organization discourse*. **Science Quarterly**, [S.l.], vol. 24, no. 4, December 1979, p. 630-637.

MARTIN, P.; TATE, K. *Getting started in project management*. New York: John Wiley & Sons, 2001.

MARTINS, G. A. Metodologias convencionais e não convencionais e a pesquisa em administração. Caderno de Pesquisas em Administração, v.00, n.0, 2 sem. 1994.

MASSACHUSSETTS INSTITUTE OF TECHNOLOGY – MIT; SOFTEX CAMPINAS. A indústria de software no Brasil 2002: fortalecendo a economia do conhecimento. Campinas, Brasil, 2003.

MARDIA, K. V.; KENT, J. T.; BIBBY, J. M. *Multivariate analysis*. London: Academic Press, 1979.

MAURER, Frank.; MARTEL, Sebastien. *Extreme programming: rapid development for webbased applications.* **IEEE Internet Computing,** [S.l.], v. 6, n. 1, p. 86–90, Jan. 2002.

\_\_\_\_\_. *On the productivity of agile software practices:* an industrial case study. Abr. 2004. Disponível em: <a href="http://sern.ucalgary.ca/milos/papers/2002/MaurerMartel2002c.pdf">http://sern.ucalgary.ca/milos/papers/2002/MaurerMartel2002c.pdf</a>>. Acesso em Agosto, 2005.

MAXIMIANO, A. C. A. **Administração de projetos:** como transformar idéias em resultados. São Paulo: Atlas, 2 ed. 2002.

McBREEN, P. Questioning extreme programming. Boston: Addison-Wesley, 2003.

McCABE, R.; POLEN, M. Should you be more agile? Crosstalk, The Journal of Defense Software Engineering, [S.l.], Oct, 2002.

MEREDITH, Jack R., MANTEL, Samuel J. *Project management:* a managerial approach, New York: John Wiley & Sons, Inc., 2000.

MILLER, A. Selection of Subsets of Regression Variables. **Journal of the Royal Statistical Association**, v. 147 (A), 389-425, 1984.

NEVES, J. L. Pesquisa qualitativa – características, usos e possibilidades. **Caderno de Pesquisas em Administração**, v.1, n.3, 2 sem. 1996.

NERUR, S. et al. Challenges of migrating to agile methodologies: organizations must carefully assess their readiness before treading the path of agility. **Communication of the ACM**, [S.l.], v.48, n.5, p 73-78, May 2005.

PÁDUA, E. M. M. **Metodologia de pesquisa:** abordagem teórico-prática, Campinas: Papirus, 1997.

PALMER, S. R.; FELSING, M. A practical guide to feature-driven development. [S.l.]: Pearson Education, 2001. PAULK, Mark. C. ExtremepProgramming from a SW-CMM perspective. **IEEE Software**, [S.l.], v. 18, n. 6, p. 19–26, Nov. 2001. \_. Agile methodologies and process discipline. Crosstalk - The Journal of Defense **Software Engineering**, [S.l.], v. 15, n. 10, p. 15–28, Oct. 2002. PINTO, Jeffrey K; SLEVIN, Dennis P. Critical factors in successful project implementation. **IEEE Transactions an Engineering Management**, [S.l.], Feb. 1987. . Critical success factors across the project life cycle. Project Management Journal, Drexel Hill, v. XIX, n.3, p. 67-65, June, 1988. PINTO, Jeffrey K.; KHARBANDA, O.P. Successful project managers. New York: Van Nostrand Reinhold, 1995a. \_\_\_\_\_. Lessons for an accidental profession. **Business Horizons**, Indiana, 1995b. PINTO, Ricardo L. Evolução da estrutura organizacional ao longo do ciclo de vida do projeto: um estudo de caso. São Paulo, 2002. Dissertação (Mestrado em Administração) – Departamento de Administração da Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo. POLEN, M.; McCABE, R. Should you be more agile? Crosstalk, The Journal of Defense Software Engineering., [S.1.], Oct. 2002. PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO. Departamento de Informática. Ensino de engenharia de software: relato de experiência. Rio de Janeiro, 2004. Disponível em < http://www-di.inf.puc-rio.br/~julio/silva\_wei2004.pdf>. Acesso em setembro de 2005. POOLE, C.; HUISMAN, J. W. Using extreme programming in a maintenance environment. **IEEE Software Computer**, [S.l.], v. 18, n. 6, p. 42–50, 2001. POPPENDIECK. M. Lean programming. Software Development Magazine. May-June, 2001 Disponível em <a href="http://www.agilealliance.org/articles/articles/LeanProgramming.htm">http://www.agilealliance.org/articles/articles/LeanProgramming.htm</a>. Acesso em julho, 2005. POPPENDIECK. M.; POPPENDIDIECK. T. Lean software development. Boston: Addison-Wesley, 2003. PROJECT MANAGEMENT INSTITUTE - PMI. PMBOK guide: Um guia do conjunto de conhecimentos do gerenciamento de projetos. Pennsylvania: Project Management Institute, 2000 ed, 2000. \_\_\_\_. OMP3: Organizational project management maturity model. Pennsylvania: Project Management Institute, 2003.

\_\_. Guia PMBoK: Um guia do conjunto de conhecimentos do gerenciamento de

projetos. Pennsylvania: Project Management Institute, 3. ed, 2004.

- \_\_\_\_. **São Paulo, Brasil Chapter**, 2005. Disponível em <a href="http://www.pmisp.org.br/home.asp">http://www.pmisp.org.br/home.asp</a>>. Acesso em setembro, 2005.
- PORTER, Michael E. **A vantagem competitiva das nações**. Trad. Elizabeth Maria de Pinto Braga. Rio de Janeiro: Campus, 25 ed. 1989.
- PORTER, M. E.; MILLAR, V. E. *How information gives you competitive advantage*. **Harvard Business Review**, v. 63, n. 4, p. 149-160, Jul.-Aug. 1985.
- PRAHALAD, C. K., RAMASWAMY, V. **O futuro da competição:** como desenvolver diferenciais inovadores em parceria com os clientes. Trad. Afonso Carlos da Cunha Senha. Rio de Janeiro: Elsevier, 2004.
- R PROJECT FOR STATISTICAL COMPUTING. *R: A language and environment for statistical computing*. Viena, Áustria, 2005. Disponível em <a href="http://www.R-project.org">http://www.R-project.org</a>. Acesso em agosto de 2005.
- RAD, P. F., RAGHAVAN, A. Establishing an organizational project office. AACE International Transactions, [S.1.], 2000.
- RAMOS, M. Y. *et al.* Avaliação do desempenho do processo de desenvolvimento de novos produtos: proposição de uma tipologia para a construção de sistemas integrados de medidas. **XXIII Simpósio de Gestão da Inovação Tecnológica**. Curitiba PR, Outubro, 2004.
- REIFER, Donald J. How good are agile methods? **IEEE Software**, [S.l.], p. 14-17, jul-ago, 2002.
- ROCKART, J.F., SHORT, J.E. *The networked organization and the management of interdependence*. In: **The Corporation of the 1990s**. New York: Oxford University Press, 1991.
- ROYCE, W. W. Managing the development of large software systems: concepts and techniques. Proc Wescon, 1970, p. 1-9.
- RUS, I. et al. Process diversity in software maintenance guest editors's introduction. **Software Maintenance Research and Practice**, Dec. 2002.
- SANJIV, Augustine; WOODCOCK, Susan. *Agile project management: emergent order through visionary leadership.* May, 2003. Disponível em <a href="http://www.ccpace.com/resources/AgileProjectManagement.pdf">http://www.ccpace.com/resources/AgileProjectManagement.pdf</a> . Acesso em agosto de 2005.
- SBRAGIA, Roberto. Avaliação do desempenho de projetos em instituições de pesquisa: um estudo empírico dentro do setor de tecnologia industrial. **Revista de Administração**, vol. 19 (1). jan-mar 1984, p. 83-93.
- \_\_\_\_\_. A interface entre gerentes de projeto e gerentes funcionais em estruturas matriciais. **Revista de Administração**, v. 20, n. 2, p, 48-55, abr-jun 1985.
- SBRAGIA, Roberto et al. Los indicadores de I&D&I en las empresas mas y menos innovadoras. Espacios, [S.l.], vol. 20, no. 1, 1999, p. 5-22.

SCHRADER, A. Introdução à pesquisa social empírica. Porto Alegre: Editora Globo, 1974.

SCHWABER, K.; BEEDLE, M. Agile software development with scrum. NJ: Pretence Hall, 2001.

SCHWABER, K. *Controlled chaos:* living on the edge. 2002. Disponível em <a href="http://www.agilealliance.org/articles/articles/ap.pdf">http://www.agilealliance.org/articles/articles/ap.pdf</a>>. Acesso em junho, 2005.

SCHILLING, M.A., HILL, C.W.L. Managing the new product development process: strategic imperatives. Academy of Management Executive, [S.l.:s.n], v.12, n.13, p. 67-81, 1998.

SCOTT, B. *The secrets of software success*. Jul, 2001. Disponível em <a href="http://www.cio.com/archive/070101/secret.html">http://www.cio.com/archive/070101/secret.html</a>>. Acesso em agosto de 2005.

SELLTIZ, C. *et al.* **métodos de pesquisa nas relações sociais**. Trad. Dante Moreira Leite, São Paulo: EDUSP, 1974.

SHENHAR, A. J. et al. Mapping the dimensions of project success. **Project Management Journal**, [S.l.], June, 1997.

SHTUB, A. et al. **Project management:** engineering, technology and implementation. Englewood Cliffs: Prentice-Hall, 1994.

SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE – SBES, 17, 2003. Manaus. Disponível em <a href="http://www.sbbd.fua.br/index.htm">http://www.sbbd.fua.br/index.htm</a>. Acesso em setembro, 2005.

SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE – SBES, 19, 2005. Uberlândia. Disponível em <a href="http://www.sbbd-sbes2005.ufu.br/sbes\_tutoriais.aspx">http://www.sbbd-sbes2005.ufu.br/sbes\_tutoriais.aspx</a>. Acesso em setembro, 2005.

SOFTWARE ENGINEERING INSTITUTE – SEI. *The capability maturity model for software:* guidelines for improving the software process. Boston: Addison-Wesley, 1995.

STANDISH GROUP INTERNATIONAL. *The chaos report, 1999*. Disponível em <a href="http://www.standishgroup.com/sample\_research/PDFpages/Chaos\_1999.pdf">http://www.standishgroup.com/sample\_research/PDFpages/Chaos\_1999.pdf</a>>. Acesso em novembro 2004.

com/sam	,				<a href="http://www.standishgroup.">http://www.standishgroup.</a> novembro 2004.
50%. Ma	isponível	_	-	•	ccess rates have improved by .com/press/article.php?id=2>.
report.	em <ht< th=""><th>tp://www</th><th>- 0</th><th></th><th>excerpt from the third quarter mple_research/PDFpages/q3-</th></ht<>	tp://www	- 0		excerpt from the third quarter mple_research/PDFpages/q3-

STRAUSS, A. L., CORBIN, J. Basics of qualitative research, 1990.

SOCIEDADE DOS USUÁRIOS DE INFORMÁTICA E TECLECOMUNICAÇÕES - SUCESU. Calendário de eventos. [S.l.], 2005a. Disponível em <a href="http://www.sucesu.org.br">http://www.sucesu.org.br</a>. Acesso em agosto, 2005.

SOCIEDADE DOS USUÁRIOS DE INFORMÁTICA E TECLECOMUNICAÇÕES - SUCESU. Grupo de Usuários de Métodos Ágeis. Porto Alegre, 2005b. Disponível em <a href="http://www.rs.sucesu.org.br/grupos\_usuario/www.rs.sucesu.org.br/grupos\_usuario/GUMA">http://www.rs.sucesu.org.br/grupos\_usuario/GUMA</a>>. Acesso em agosto, 2005.

SUGAR LOAF PLOP 2005. *5th Latin America Conference on Pattern Languages of Programming*. Campos do Jordão. Agosto, 2005. Disponível em <a href="http://sugarloafplop2005.icmc.usp.br/papers.html">http://sugarloafplop2005.icmc.usp.br/papers.html</a>>. Acesso em setembro, 2005.

SUTHERLAND, J. Agile can scale: inventing and reinventing scrum in five companies. **Cutter IT Journal**, [S.l.], v. 14, n. 12, 2001.

THOMSETT, R. Radical Project Management. NJ: Prentice Hall, 2002.

TICEHURST, G. W. & VEAL, A. J. Business research methods: a managerial approach. Longman, 1999.

TURK, D. *et al. Limitations of agile software processes*. Jan, 2003. Disponível em <a href="http://www.agilealliance.org/articles/articles/LimitationsofAgile.pdf">http://www.agilealliance.org/articles/articles/LimitationsofAgile.pdf</a> Acesso em Agosto, 2005.

\_\_\_\_\_. Assumptions underlying agile software development process. Colorado State University, 2005.

UNIVERSIDADE DE SÃO PAULO - USP. Instituto de Matemática e Estatística. **Métodos ágeis de desenvolvimento de software.** São Paulo, junho 2004. Disponível em <a href="https://www.ime.usp.br/~kon/presentations/XP2004.ppt">www.ime.usp.br/~kon/presentations/XP2004.ppt</a>>. Acesso em setembro, 2005.

UNIVERSIDADE FEDERAL DA BAHIA - UFBA. Departamento de Ciências da Computação. **Relatório de atividades de 2003.** Salvador, 2003. Disponível em <a href="http://twiki.im.ufba.br/bin/view/Aside/RelatorioGeral2003">http://twiki.im.ufba.br/bin/view/Aside/RelatorioGeral2003</a>>. Acesso em setembro de 2005.

URBAN, G; HAUSER, J. Design and marketing of new projects. NJ: Prentice-Hall, 1993.

UDO, N., KOPPENSTEINER, S. Will agile change the way we manage software projects? Agile from a PMBoK guide perspective. Projectway, [S.1.], 2003.

VASCONCELLOS, Eduardo; HEMSLEY, James R. **Estrutura das organizações**. São Paulo: Pioneira, 1986.

VELOSO, F. et al. Slicing the knowledge-based economy in Brazil, China and India: a tale of 3 software industries. Massachusetts, 2003.

VERZUH, E. *The fast foward in MBA in project management. John Wiley & Sons, Inc.*, 1999.

WOMACK, James. P. et al. The machine that changes the world: the story of lean production. New York: Rawson Associates, 1990.

#### **ANEXOS**

ANEXO A – QUESTIONÁRIO

ANEXO B – TABELAS DA ESTATÍSTICA DESCRITIVA

ANEXO C – TABELAS E GRÁFICOS ANÁLISE DISCRIMINANTE

ANEXO D – TABELA DA REGRESSÃO LOGÍSTICA

## ANEXO A - QUESTIONÁRIO

## MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SISTEMAS DE TI

#### Seção 1 - Introdução

Esta pesquisa tem por obietivos identificar os fatores críticos de sucesso e a abordagem de gerenciamento de

projetos mais indicada para o desenvolvimento de sistemas de TI, conduzidos segundo um enfoque <i>ágil</i> . Os dados aqui coletados serão utilizados única e exclusivamente em pesquisa acadêmica, sem qualquer finalidade comercial.  Você não levará mais do que 5 minutos responde as questões. Muito obrigada!				
Marisa Villas Bôas Dias				
Mestranda em Administração (FEA – USP)				
marisa.dias@terra.com.br				
Seção 2 – Qualificação do Respondente				
Você já participou de algum projeto de desenvolvimento de software que utilizasse Métodos Ágeis (ex: XP, SRUM, FDD, ASD ou outro) para a execução e/ou gerenciamento do projeto?				
Sim Não Obs: se a opção for NÃO, o respondente é direcionado automaticamente ao final da pesquisa – comentários e agradecimento. Se a opção for SIM, o respondente está qualificado a responder a demais questões.				
Seção 3 – Caracterização do Respondente				
1. Qual é o seu cargo?				
Gerente ou Diretor de Projeto Programador / Analista / Consultor				
Gerente ou Diretor de Produto  Presidente ou Vice-Presidente				
Gerente ou Diretor de Sistemas (TI)  Outro. Especificar				
2. Há quanto tempo você trabalha com Métodos Ágeis?				
< 1 ano 1 a 4 anos 5 a 10 anos 11 a 15 anos > 15 anos				
3. Qual o tipo de Método Ágil você utiliza?				
XP Crystal SCRUM FDD DSDM ASD Outro. Especificar				
4. Qual o percentual de sistemas desenvolvidos em sua empresa com a utilização de Métodos Ágeis. $<10\%$ $10-30\%$ $31-50\%$ $51-70\%$ $71-90\%$ $>91\%$				
Sação 4. Qualificação do Projeto				

#### Seção 4 – Qualificação do Projeto

Pense em projeto de desenvolvimento de um sistema de TI (Sistema X) que foi conduzido segundo um enfoque ágil (com a utilização de Métodos Ágeis). Por favor, responda a pesquisa tendo em mente apenas este projeto, que pode ter sido bem-sucedido, mal-sucedido ou ocupar alguma posição intermediária. É muito importante que todas as questões sejam respondidas!

6. Siste	Selecione o principal critério utilizado para mensurar o sucesso do projeto de ema X?	dese	nvolv	imen	to do				
	Satisfação das expectativas dos <i>stakehoders</i> (principais interessados no projeto)								
	Atendimento aos objetivos / requisitos do projeto								
	Entrega do projeto dentro do orçamento previsto								
	Entrega do projeto dentro do prazo previsto								
	Entrega segundo os padrões de qualidade acordados								
	Satisfação do time de projeto								
	Indicadores financeiros (ROI, Período de <i>Payback</i> , Taxa Interna de Retorno ou outro)								
	Outro. Especificar:								
7.	Como você classificaria o grau de sucesso do projeto para desenvolvimento	lo Sis	tema	X?					
	Sucesso Sucesso Parcial Insucesso								
8. dura	Classifique o grau de combinação dos enfoques clássico e ágil de gerenciamente o desenvolvimento do Sistema X.	ento d	e proj	etos	utiliza	ado			
	Não significativo Pouco significativo Moderado Significati	vo	M	uito s	ignifi	cativo			
		L							
9.	Um champion do produto (sistema) estava envolvido no projeto?								
	Não Sim, de uma área de negócio Sim, de sistemas ou TI								
	Sim, do corpo diretivo Sim, outro. Especificar:								
	<u> </u>								
10.	10. Qual o prazo previsto para a entrega do projeto de desenvolvimento do Sistema X?								
	< 1 mês								
•									
11.	Quantas pessoas compunham o time central (core team) de desenvolvimento	?		_					
	1-3 4-10 11-20 21-50 51-100 101-2	200		>20	0				
12.	Qual o percentual as funcionalidades finais do sistema que foram liberadas n	a prin	neira '	versã	o?				
	0-20% 21-40% 41-60% 61-80% 81-100%								
	Seção 5 — Técnicas e Características dos Métodos Ágeis								
	responder as questões a seguir, assinale a alternativa que melhor expresse seus se								
	cordo Totalmente; DP – Discordo Parcialmente; N – Neutro, CP – Concordo Parcialmente.	almer	ite; C	1 – (	onco	rdo			
		DT	DP	N	СР	СТ			
13.	A coordenação entre os grupos envolvidos no projeto, dentro da empresa, foi bem-sucedida.	-							
14.	Houve um grande apoio da alta administração no projeto.								
15.									
16.	3 1 1 1 0								

		DT	DP	N	CP	СТ
17.	As entregas das versões foram gerenciadas de forma eficiente.					
18.	As sessões de planejamento das entregas de versões foram bem gerenciadas.					
19.	Os testes da codificação foram bem-sucedidos.					
20.	A equipe de desenvolvimento foi composta por profissionais bem qualificados tecnicamente.					
21.	A equipe de desenvolvimento recebeu treinamento formal no enfoque ágil.					
22.	A equipe de desenvolvimento foi informalmente treinada no enfoque ágil.					
23.	Os desenvolvimentos sofreram modificações (adaptações) ao longo do projeto.					
24.	As solicitações de mudanças (adaptações) foram incorporadas sem problemas no decorrer das várias versões ( <i>releases</i> ).					
25.	Os programadores e analistas estavam muito motivados com o projeto.					
26.	Os programadores e analistas estavam bem familiarizados com o enfoque ágil.					
27.	O código foi desenvolvido em um sentimento de propriedade coletiva (os programadores se sentiam à vontade e capazes de alterar o código desenvolvido por outro profissional).					
28.	O protótipo do sistema foi liberado em um estágio inicial do projeto.					
29.	Os participantes da equipe de desenvolvimento estavam totalmente dedicados ao projeto (comprometidos com o projeto, sem demandas concorrentes ou outras prioridades impostas pelo negócio ou área).					
30.	Os desenvolvimentos diários de sistemas eram entregues com a utilização de técnicas dos métodos ágeis.					
31.	Os membros da equipe de desenvolvimento usualmente reviam a codificação desenvolvida por outra pessoa, antes da liberação final.					
32.	Houve um clima de descontração, alegria e diversão durante o desenvolvimento.					
33.	Os enfoque ágil é mais indicado ao gerenciamento de projetos de desenvolvimento de sistemas de TI do que o enfoque clássico de gerenciamento de projetos (como por exemplo, o enfoque baseado em processos proposto pelo PMBoK - PMI).					
34.	O uso de um Método Ágil foi fundamental para o sucesso deste projeto de desenvolvimento do Sistema X.					

### Seção 6 - Comentários

Obrigada por sua col	laboração!
----------------------	------------

Sinta-se à vontade para registrar seus comentários ou fornecer alguma informação que julgar importante no espaço abaixo.

## ANEXO B - TABELAS DA ESTATÍSTICA DESCRITIVA

1) Tabela relativa à qualificação do respondente

Tabela 26 - Experiência em desenvolvimento de software com o uso de Métodos Ágeis

Experiência	Contagem / Percentual
Sim	130 (55,3%)
Não	105 (44,7%)

2) Tabelas relativas à caracterização do respondente

Tabela 27 - Qualificação do respondente quanto ao cargo (Q1)

Cargo	Contagem / Percentual
Gerente ou Diretor de Projeto	34 35,1%
Gerente ou Diretor de Produto	1 1%
Gerente ou Diretor de Sistemas (TI)	10 10,3%
Programador / Analista / Consultor	43 44,3%
Presidente ou Vice-Presidente	3 3,1%
Outro	6 6,2%

Tabela 28 - Tempo de experiência com Métodos Ágeis (Q2)

Tempo	Contagem / Percentual
Menos de 1 ano	32 42,1%
1 a 4 anos	44 57,9%

Tabela 29 - Método ágil utilizado (Q3)

	, ( <b>L</b> -)
Método Ágil	Contagem / Percentual
XP	78 81,2%
Crystal Methods	0 0%
Scrum	5 5,2%
FDD	6 6,2%
ASD	0 0%
Outro	7 7,3%

Tabela 30 - Percentual de projetos realizados com o uso de Métodos Ágeis (Q4)

Método Ágil	Contagem / Percentual
< 10%	18 18,9%
10 – 30%	20 21,1%
31 – 50 %	14 14,7%
51 – 70 %	14 14,7%
71 – 90 %	7 7,4%
> 91%	22 23,2%

3) Tabelas relativas à caracterização do projeto

Tabela 31 - Principal critério para mensurar o sucesso de um projeto (Q6)

Método Ágil	Contagem / Percentual
Satisfação das expectativas dos stakeholders	28 45,9%
Atendimento aos objetivos / requisitos	12 19,7%
Entrega dentro do orçamento previsto	1 1,6%
Entrega dentro do prazo previsto	9 14,8%
Entrega segundo padrões de qualidade	3 4,9%

Método Ágil	Contagem / Percentual	
Satisfação do time	4 6,6%	
Indicadores financeiros	2 3,3%	
Outro	2 3,3%	

Tabela 32 - Desempenho do projeto (Q7)

Desempenho do Projeto	Contagem / Percentual
Sucesso	42 68,9%
Sucesso parcial	19 31,1%
Insucesso	0 0%

Tabela 33 - Grau de Combinação entre os enfoques ágil e clássico de gerenciamento de projetos (Q8)

Grau de Combinação	Contagem / Percentual
Não significativo	4 8,5%
Pouco significativo	7 14,9%
Moderado	26 55,3%
Significativo	8 17%
Muito significativo	2 4,3%

Tabela 34 - Envolvimento do champion do sistema (Q9)

Envolvimento do Champion	Contagem / Percentual
Não	15 34,1%
Sim, do corpo diretivo	4 9,1%
Sim, de uma área de negócio	17 38,6%
Sim, de sistemas de TI	7 15,9%
Sim, outros.	1 2,3%

Tabela 35 - Prazo para a entrega do projeto (Q10)

Tempo	Contagem / Percentual
< 1 mês	1 1,7%
1-3 meses	20 33,3%
4 – 6 meses	19 31,7%
7 – 12 meses	15 25%
> 12 meses	5 8,3%

Tabela 36 – Número de integrantes da equipe de desenvolvimento (Q11)

Número de integrantes	Contagem / Percentual
1-3	15 29,4%
4 - 10	31 60,8%
11 - 20	4 7,8%
21 – 50	0 0%
51 – 100	1 2%
101 - 200	0 0%
> 200%	0 0%

Tabela 37 - Percentual de funcionalidades liberadas na primeira versão (Q12)

	1
Percentual de Funcionalidades	Contagem / Percentual
0 - 20%	18 36%
21 – 40%	12 24%
41 – 60%	8 16%
61 – 80%	7 14%
81 – 100%	5 10%

# 4) Tabelas relativas às práticas e características dos Métodos Ágeis

Tabela 38 – Respostas das questões 13 a 34

Tabela 38 – Respostas das questoes 13 a 34						
Questão	DT	DP	N	CP	CT	Média
Q13	4% (2)	6% (3)	10% (5)	40% (21)	40% (21)	4.08
Q14	2% (1)	8% (4)	20% (10)	33% (17)	37% (19)	3.96
Q15	2% (1)	2% (1)	12% (6)	37% (18)	47% (23)	4.24
Q16	6% (3)	12% (6)	22% (11)	35% (18)	25% (13)	3.63
Q17	4% (2)	6% (3)	6% (3)	56% (28)	28% (14)	3.98
Q18	2% (1)	8% (4)	22% (11)	39% (20)	29% (15)	3.86
Q19	4% (2)	10% (5)	8% (4)	40% (20)	38% (19)	3.98
Q20	4% (2)	10% (5)	0% (0)	29% (15)	57% (29)	4.25
Q21	33% (17)	24% (12)	18% (9)	16% (8)	10% (5)	2.45
Q22	12% (6)	10% (5)	8% (4)	35% (18)	35% (18)	3.73
Q23	2% (1)	2% (1)	6% (3)	26% (13)	64% (32)	4.48
Q24	2% (1)	6% (3)	6% (3)	31% (16)	55% (28)	4.31
Q25	4% (2)	4% (2)	2% (1)	27% (14)	63% (32)	4.41
Q26	8% (4)	30% (15)	18% (9)	34% (17)	10% (5)	3.08
Q27	4% (2)	10% (5)	14% (7)	37% (19)	35% (18)	3.90
Q28	10% (5)	4% (2)	8% (4)	24% (12)	55% (28)	4.10
Q29	12% (6)	14% (7)	6% (3)	33% (17)	35% (18)	3.67
Q30	10% (5)	10% (5)	16% (8)	39% (20)	25% (13)	3.61
Q31	16% (8)	20% (10)	12% (6)	39% (20)	14% (7)	3.16
Q32	4% (2)	8% (4)	4% (2)	35% (18)	49% (25)	4.18
Q33	12% (6)	8% (4)	14% (7)	29% (15)	37% (19)	3.73
Q34	6% (3)	6% (3)	16% (8)	33% (17)	39% (20)	3.94
	•		•	•		

Legenda: DT – Discordo Totalmente; DP – Discordo Parcialmente; N – Neutro; CP- Concordo Parcialmente; CT – Concordo Totalmente.

# 5) Tabelas de Contingência – Todas as questões contra a Q7.

Tabela 39 - Questão 1 versus Questão 7

Course	Desempenho do Projeto		
Cargo	Sucesso Parcial	Sucesso Total	
Gerente ou Diretor de Projeto	9 (0,50)	16 (0,38)	
Gerente ou Diretor de Produto	0 (0,00)	0 (0,00)	
Gerente ou Diretor de Sistemas (TI)	0 (0,00)	8 (0,19)	
Programador / Analista / Consultor	6 (0,33)	15 (0,36)	
Presidente ou Vice - Presidente	0 (0,00)	1 (0,02)	

Cargo	Desempenho do Projeto		
Cargo	Sucesso Parcial Sucesso Total		
Outro	3 (0,17)	2 (0,05)	
P = 0,168			

Tabela 40 - Questão 2 versus Questão 7

Tempo (anos)	Desempenh	o do Projeto
Tempo (anos)	Sucesso Parcial Sucesso Total	
< 1	7 (0,47)	11 (0,34)
1 a 4	8 (0,53)	21 (0,66)
P = 0,627		

Tabela 41 - Questão 3 versus Questão 7

Método	Desempenho do Projeto		
Ivictodo	Sucesso Parcial	Sucesso Total	
XP	13 (0,72)	34 (0,81)	
SCRUM	1 (0,06)	1 (0,02)	
FDD	2 (0,11)	3 (0,07)	
Outro	2 (0,11)	4 (0,10)	
P = 0.856			

Tabela 42 - Questão 4 versus Questão 7

Tabela 42 - Questao 4 versus Questao 7		
Percentual	Desempenho do Projeto	
(%)	Sucesso Parcial	Sucesso Total
< 10	5 (0,28)	5 (0,12)
10 - 30	2 (0,11)	9 (0,21)
31 - 50	2 (0,11)	9 (0,21)
51 - 70	3 (0,17)	7 (0,17)
71 - 90	2 (0,11)	3 (0,07)
> 91	4 (0,22)	9 (0,21)
P = 0,607		

Tabela 43 - Questão 6 versus Questão 7

Critério	Desempenho do Projeto	
	Sucesso Parcial	Sucesso Total
Satisfação das expectativas dos stakeholders	6 (0,32)	22 (0,52)
Atendimento aos objetivos / requisitos	6 (0,32)	6 (0,14)
Entrega dentro do orçamento previsto	0 (0,00)	1 (0,02)
Entrega dentro do prazo previsto	3 (0,16)	6 (0,14)

Dese		10 do Projeto
Entrega segundo padrões de qualidade	1 (0,05)	2 (0,05)
Satisfação do time	3 (0,16)	1 (0,02)
Indicadores financeiros	0 (0,00)	2 (0,05)
Outro	0 (0,00)	2 (0,05)
P = 0.246		

Tabela 44 - Questão 8 versus Questão 7

Escore	Desempenho do Projeto	
Escore	Sucesso Parcial	Sucesso Total
Não significativo	0 (0,00)	4 (0,12)
Pouco significativo	1 (0,07)	6 (0,18)
Moderado	10 (0,71)	16 (0,48)
Significativo	3 (0,21)	5 (0,15)
Muito significativo	0 (0,00)	2 (0,06)
P = 0,341		

Tabela 45 - Questão 9 versus Questão 7

Champion opvolvido	Desempenho do Projeto	
Champion envolvido	Sucesso Parcial	Sucesso Total
Não	3 (0,21)	12 (0,40)
Sim, de uma área de negócio	2 (0,14)	2 (0,07)
Sim, de sistemas ou TI	6 (0,43)	11 (0,37)
Sim, do corpo diretivo	3 (0,21)	4 (0,13)
Sim, outro	0 (0,00)	1 (0,03)
	P = 0,639	

Tabela 46 - Questão 10 versus Questão 7

Prazo	Desempenho do Projeto	
(meses)	Sucesso Parcial	Sucesso Total
< 1	0 (0,00)	1 (0,02)
1 - 3	5 (0,26)	15 (0,37)
4 - 6	4 (0,21)	15 (0,37)
7 - 12	7 (0,37)	8 (0,20)
> 12	3 (0,16)	2 (0,05)
P = 0,256		

Tabela 47 - Questão 11 versus Questão 7

Pessoas	Desempenho do Projeto	
1 essuas	Sucesso Parcial	Sucesso Total
1 - 3	4 (0,27)	11 (0,31)
4 - 10	8 (0,53)	23 (0,64)
11 - 20	2 (0,13)	2 (0,06)
21 - 50	0 (0,00)	0 (0,00)
51 - 100	1 (0,07)	0 (0,00)
> 100	0 (0,00)	0 (0,00)
P = 0,325		

Tabela 48 - Questão 12 versus Questão 7

Sucesso Total 14 (0,40)
9 (0.22)
8 (0,23)
5 (0,14)
4 (0,11)
4 (0,11)
_

Tabela 49 - Questão 13 versus Questão 7

Escore	Desempenho do Projeto	
Escore	Sucesso Parcial	Sucesso Total
1	1 (0,06)	1 (0,03)
2	3 (0,18)	0 (0,00)
3	2 (0,12)	3 (0,09)
4	4 (0,24)	17 (0,49)
5	7 (0,41)	14 (0,40)
média	3,8	4,2
	P = 0,079	

Tabela 50 - Questão 14 versus Questão 7

Escore	Desempenho do Projeto	
Escore	Sucesso Parcial	Sucesso Total
1	0 (0,00)	1 (0,03)
2	3 (0,18)	1 (0,03)
3	3 (0,18)	7 (0,21)
4	4 (0,24)	13 (0,38)
5	7 (0,41)	12 (0,35)
média	3,9	4,0
P = 0,341		

Tabela 51 - Questão 15 versus Questão 7

Escore	Desempenho do Projeto	
Escore	Sucesso Parcial	Sucesso Total
1	0 (0,00)	1 (0,03)
2	1 (0,06)	0 (0,00)
3	3 (0,19)	3 (0,09)
4	6 (0,38)	12 (0,36)
5	6 (0,38)	17 (0,52)
média	4,1	4,3
P = 0.430		

Tabela 52 - Questão 16 versus Questão 7

Escore	Desempenho do Projeto	
	Sucesso Parcial	Sucesso Total
1	2 (0,12)	1 (0,03)
2	3 (0,18)	3 (0,09)
3	5 (0,29)	6 (0,18)
4	6 (0,35)	12 (0,35)
5	1 (0,06)	12 (0,35)
média	3,1	3,9
P = 0.145		

Tabela 53 - Questão 17 versus Questão 7

Escore Desempenho do Projeto		o do Projeto
Liscore	Sucesso Parcial	Sucesso Total
1	1 (0,06)	1 (0,03)
2	2 (0,12)	1 (0,03)
3	1 (0,06)	2 (0,06)

	Desempenho do Projeto	
4	10 (0,62)	18 (0,53)
5	2 (0,12)	12 (0,35)
média	3,6	4,1
	P = 0.386	

Tabela 54 - Ouestão 18 versus Ouestão 7

Escore	Desempenho do Projeto	
Escore	Sucesso Parcial	Sucesso Total
1	0 (0,00)	1 (0,03)
2	2 (0,12)	2 (0,06)
3	4 (0,24)	7 (0,21)
4	7 (0,41)	13 (0,38)
5	4 (0,24)	11 (0,32)
média	3,6	4,1
	P = 0,849	

Tabela 55 - Questão 19 versus Questão 7

Escore	Desempenho do Projeto	
Liscore	Sucesso Parcial	Sucesso Total
1	0 (0,00)	2 (0,06)
2	3 (0,18)	2 (0,06)
3	2 (0,12)	2 (0,06)
4	9 (0,53)	11 (0,33)
5	3 (0,18)	16 (0,48)
média	3,7	4,1
	P = 0.142	

Tabela 56 - Questão 20 versus Questão 7

Escore	Desempenho do Projeto	
Escore	Sucesso Parcial	Sucesso Total
1	1 (0,06)	1 (0,03)
2	3 (0,18)	2 (0,06)
3	0 (0,00)	0 (0,00)
4	3 (0,18)	12 (0,35)
5	10 (0,59)	19 (0,56)
média	4,1	4,4
	P = 0.381	

Tabela 57 - Questão 21 versus Questão 7

Escore Desempenho do Projeto		o do Projeto
Liscore	Sucesso Parcial	Sucesso Total
1	8 (0,47)	9 (0,26)
2	2 (0,12)	10 (0,29)
3	3 (0,18)	6 (0,18)
4	2 (0,12)	6 (0,18)
5	2 (0,12)	3 (0,09)
média	2,3	2,5
	P = 0.510	

Tabela 58 - Questão 22 versus Questão 7

Escore	Desempenho do Projeto	
Liscore	Sucesso Parcial	Sucesso Total
1	1 (0,06)	5 (0,15)
2	3 (0,18)	2 (0,06)
3	2 (0,12)	2 (0,06)
4	6 (0,35)	12 (0,35)
5	5 (0,29)	13 (0,38)
média	3,6	3,8
	P = 0.541	

Tabela 59 - Ouestão 23 versus Ouestão 7

	Tabela 59 - Questao 25 versus Questao 7	
Escore	Escore Desempenho do Projeto	
Liscore	Sucesso Parcial	Sucesso Total
1	0 (0,00)	1 (0,03)
2	1 (0,06)	0 (0,00)
3	1 (0,06)	2 (0,06)
4	9 (0,53)	4 (0,12)
5	6 (0,35)	26 (0,79)
média	4,2	4,6
	P = 0.011	

Tabela 60 - Questão 24 versus Questão 7

Tubera of Questas 21707505 Questas 7		
Escore	Desempenho do Projeto	
Liscore	Sucesso Parcial	Sucesso Total
1	0 (0,00)	1 (0,03)
2	2 (0,12)	1 (0,03)
3	2 (0,12)	1 (0,03)
4	6 (0,35)	10 (0,29)

	Desempenho do Projeto	
5	7 (0,41)	21 (0,62)
média	4,1	4,4
	P = 0.342	

Tabela 61 - Questão 25 versus Questão 7

Escore	Desempenho do Projeto	
Escore	Sucesso Parcial	Sucesso Total
1	1 (0,06)	1 (0,03)
2	2 (0,12)	0 (0,00)
3	1 (0,06)	0 (0,00)
4	8 (0,47)	6 (0,18)
5	5 (0,29)	27 (0,79)
média	3,8	4,7
	P = 0,006	

Tabela 62 - Questão 26 versus Questão 7

Escore	Desempenho do Projeto	
Escore	Sucesso Parcial	Sucesso Total
1	0 (0,00)	4 (0,12)
2	9 (0,53)	6 (0,18)
3	4 (0,24)	5 (0,15)
4	3 (0,18)	14 (0,42)
5	1 (0,06)	4 (0,12)
média	2,8	3,2
	P = 0.050	

Tabela 63 - Questão 27 versus Questão 7

Escore	Desempenho do Projeto	
Escore	Sucesso Parcial	Sucesso Total
1	1 (0,06)	1 (0,03)
2	4 (0,24)	1 (0,03)
3	3 (0,18)	4 (0,12)
4	5 (0,29)	14 (0,41)
5	4 (0,24)	14 (0,41)
média	3,4	4,1
P = 0.144		

Tabela 64 - Questão 28 versus Questão 7

Escore	Desempenho do Projeto	
Liscore	Sucesso Parcial	Sucesso Total
1	2 (0,12)	3 (0,09)
2	1 (0,06)	1 (0,03)
3	1 (0,06)	3 (0,09)
4	7 (0,41)	5 (0,15)
5	6 (0,35)	22 (0,65)
média	3,8	4,2
P = 0.228		

Tabela 65 - Questão 29 versus Questão 7

Escore	Desempenho do Projeto	
Liscore	Sucesso Parcial	Sucesso Total
1	3 (0,18)	3 (0,09)
2	4 (0,24)	3 (0,09)
3	0 (0,00)	3 (0,09)
4	4 (0,24)	13 (0,38)
5	6 (0,35)	12 (0,35)
média	3,4	3,8
P = 0,312		

Tabela 66 - Questão 30 versus Questão 7

Escore	Desempenho do Projeto	
Liscore	Sucesso Parcial	Sucesso Total
1	3 (0,18)	2 (0,06)
2	4 (0,24)	1 (0,03)
3	2 (0,12)	6 (0,18)
4	8 (0,47)	12 (0,35)
5	0 (0,00)	13 (0,38)
média	2,9	4,0
P = 0.008		

Tabela 67 - Questão 31 versus Questão 7

Escore	Desempenho do Projeto	
Liscore	Sucesso Parcial	Sucesso Total
1	4 (0,24)	4 (0,12)
2	6 (0,35)	4 (0,12)
3	3 (0,18)	3 (0,09)
4	4 (0,24)	16 (0,47)

	Desempenho do Projeto	
5	0 (0,00) 7 (0,21)	
média	2,4	3,5
P = 0.039		

Tabela 68 - Questão 32 versus Questão 7

Escore	Desempenho do Projeto	
Liscore	Sucesso Parcial	Sucesso Total
1	1 (0,06)	1 (0,03)
2	2 (0,12)	2 (0,06)
3	0 (0,00)	2 (0,06)
4	8 (0,47)	10 (0,29)
5	6 (0,35)	19 (0,56)
média	3,9	4,3
P = 0,444		

Tabela 69 - Questão 33 versus Questão 7

Escore	Desempenho do Projeto	
Liscore	Sucesso Parcial	Sucesso Total
1	3 (0,18)	3 (0,09)
2	1 (0,06)	3 (0,09)
3	2 (0,12)	5 (0,15)
4	9 (0,53)	6 (0,18)
5	2 (0,12)	17 (0,50)
média	3,4	3,9
P = 0.037		

Tabela 70 - Questão 34 versus Questão 7

Tabeia 70 - Questão 54 versus Questão 7		
Escore	Desempenho do Projeto	
Liscore	Sucesso Parcial	Sucesso Total
1	2 (0,12)	1 (0,03)
2	2 (0,12)	1 (0,03)
3	4 (0,24)	4 (0,12)
4	6 (0,35)	11 (0,32)
5	3 (0,18)	17 (0,50)
média	3,4	4,2
P = 0,133		

6) Tabelas de Contingência – Todas as questões contra a Questão 8.

Tabela 71 - Questão 1 versus Questão 8

Cargo	Gerenciamento de Projeto	
Cargo	Ágil	Moderado - Clássico
Gerente ou Diretor de Projeto	6 (0,55)	14 (0,40)
Gerente ou Diretor de Produto	0 (0,00)	0 (0,00)
Gerente ou Diretor de Sistemas (TI)	2 (0,18)	5 (0,14)
Programador / Analista / Consultor	2 (0,18)	13 (0,37)
Presidente ou Vice - Presidente	0 (0,00)	0 (0,00)
Outro	1 (0,09)	3 (0,09)
P = 0.702		

Tabela 72 - Questão 2 versus Questão 8

Tempo (anos)	Gerenciamento de Projeto	
Tempo (anos)	Ágil	Moderado - Clássico
< 1	1 (0,09)	17 (0,50)
1 a 4	10 (0,91)	17 (0,50)
P = 0.040		

Tabela 73 - Questão 3 versus Questão 8

Tabela 73 - Questão 5 versus Questão 6		
Método	Gerenciamento de Projeto	
Mictodo	Ágil	Moderado - Clássico
XP	7 (0,64)	28 (0,80)
SCRUM	1 (0,09)	1 (0,03)
FDD	1 (0,09)	3 (0,09)
Outro	2 (0,18)	3 (0,09)
P = 0.624		

Tabela 74 - Questão 4 versus Questão 8

Percentual	Gerenciamen	to de Projeto
(%)	Ágil	Moderado - Clássico
< 10	0 (0,00)	7 (0,20)
10 - 30	3 (0,27)	7 (0,20)
31 - 50	3 (0,27)	5 (0,14)
51 - 70	0 (0,00)	6 (0,17)
71 - 90	0 (0,00)	4 (0,11)
> 91	5 (0,45)	6 (0,17)
P = 0.103		

Tabela 75 - Questão 6 versus Questão 8

Critério de Avaliação	Gerenciamento de Projeto	
Criterio de Avanação	Ágil	Moderado - Clássico
Satisfação das expectativas dos stakeholders	8 (0,73)	16 (0,44)
Atendimento aos objetivos / requisitos	2 (0,18)	7 (0,19)
Entrega dentro do orçamento previsto	0 (0,00)	0 (0,00)
Entrega dentro do prazo previsto	1 (0,09)	6 (0,17)
Entrega segundo padrões de qualidade	0 (0,00)	2 (0,06)
Satisfação do time	0 (0,00)	3 (0,08)
Indicadores financeiros	0 (0,00)	1 (0,03)
Outro	0 (0,00)	1 (0,03)
P = 0.705		

Tabela 76 - Questão 7 versus Questão 8

Qualificação	Gerenciamento de Projeto	
Quanneação	Ágil	Moderado - Clássico
Sucesso	10 (0,91)	23 (0,64)
Sucesso Parcial	1 (0,09)	13 (0,36)
P = 0.181		

Tabela 77 - Questão 9 versus Questão 8

Champion envolvido	Gerenciamento de Projeto	
Champion envolvido	Ágil	Moderado - Clássico
Não	1 (0,09)	14 (0,42)
Sim, de uma área de negócio	1 (0,09)	3 (0,09)
Sim, de sistemas ou TI	7 (0,64)	10 (0,30)
Sim, do corpo diretivo	1 (0,09)	6 (0,18)
Sim, outro	1 (0,09)	0 (0,00)
P = 0.075		

Tabela 78 - Questão 10 versus Questão 8

Tabela 78 - Questao 10 versus Questao 8		
Prazo	Gerenciamento de Projeto	
(meses)	Ágil	Moderado - Clássico
< 1	0 (0,00)	1 (0,03)
1 - 3	3 (0,27)	13 (0,37)
4 - 6	6 (0,55)	10 (0,29)
7 - 12	2 (0,18)	9 (0,26)
> 12	0 (0,00)	2 (0,06)
P = 0,558		

Tabela 79 - Questão 11 versus Questão 8

Pessoas	Gerenciamento de Projeto	
1 CSSOUS	Ágil	Moderado - Clássico
1 - 3	2 (0,18)	12 (0,33)
4 - 10	8 (0,73)	21 (0,58)
11 - 20	1 (0,09)	2 (0,06)
21 - 50	0 (0,00)	1 (0,03)
51 - 100	1 (0,07)	0 (0,00)
> 100	0 (0,00)	0 (0,00)
P = 0.705		

Tabela 80 - Questão 12 versus Questão 8

Percentual	Gerenciamento de Projeto	
(%)	Ágil	Moderado - Clássico
0 - 20	4 (0,36)	13 (0,37)
21 - 40	1 (0,09)	9 (0,26)
41 - 60	1 (0,09)	6 (0,17)
61 - 80	3 (0,27)	4 (0,11)
81 - 100	2 (0,18)	3 (0,09)
P = 0,476		

Tabela 81 - Questão 13 versus Questão 8

Escore	Gerenciamento de Projeto	
Liscore	Ágil	Moderado - Clássico
1	0 (0,00)	2 (0,07)
2	0 (0,00)	3 (0,10)
3	0 (0,00)	3 (0,10)
4	4 (0,40)	14 (0,47)
5	6 (0,60)	8 (0,27)
média	4,6	3,8
P = 0.275		

Tabela 82 - Questão 14 versus Questão 8

Escore	Gerenciamento de Projeto	
Liscore	Ágil	Moderado - Clássico
1	0 (0,00)	1 (0,03)
2	0 (0,00)	2 (0,07)
3	0 (0,00)	8 (0,28)
4	1 (0,10)	12 (0,41)
5	9 (0,90)	6 (0,21)

	Gerenciamento de Projeto	
média	4,9	3,7
P = 0,004		

Tabela 83 - Questão 15 versus Questão 8

Tabela 65 - Questao 15 versus Questao 6		
Escore	Gerenciamento de Projeto	
	Ágil	Moderado - Clássico
1	0 (0,00)	1 (0,03)
2	0 (0,00)	1 (0,03)
3	0 (0,00)	3 (0,10)
4	2 (0,22)	14 (0,48)
5	7 (0,78)	10 (0,34)
média	4,8	4,1
P = 0,236		

Tabela 84 - Questão 16 versus Questão 8

Escore	Gerenciamento de Projeto	
Liscore	Ágil	Moderado - Clássico
1	0 (0,00)	3 (0,10)
2	2 (0,20)	3 (0,10)
3	0 (0,00)	8 (0,28)
4	3 (0,30)	9 (0,31)
5	5 (0,50)	6 (0,21)
média	4,1	3,4
P = 0.158		

Tabela 85 - Questão 17 versus Questão 8

Escore	Gerenciamento de Projeto	
Liscore	Ágil	Moderado - Clássico
1	0 (0,00)	2 (0,07)
2	1 (0,10)	2 (0,07)
3	0 (0,00)	3 (0,10)
4	4 (0,40)	14 (0,48)
5	5 (0,50)	8 (0,28)
média	4,3	3,8
P = 0.550		

Tabela 86 - Questão 18 versus Questão 8

Escore	Gerenciamento de Projeto	
Escore	Ágil	Moderado - Clássico
1	0 (0,00)	1 (0,03)

	Gerenciamento de Projeto	
2	0 (0,00)	3 (0,10)
3	2 (0,20)	6 (0,21)
4	3 (0,30)	10 (0,34)
5	5 (0,50)	9 (0,31)
média	4,3	3,8
P = 0.704		

Tabela 87 - Questão 19 versus Questão 8

Tabela 67 - Questao 17 versus Questao 6		
Escore	Gerenciamento de Projeto	
Liscore	Ágil	Moderado - Clássico
1	0 (0,00)	2 (0,07)
2	1 (0,11)	2 (0,07)
3	1 (0,11)	2 (0,07)
4	2 (0,22)	12 (0,41)
5	5 (0,56)	11 (0,38)
média	4,2	4,0
P = 0,714		

Tabela 88 - Questão 20 versus Questão 8

Escore	Gerenciamento de Projeto	
Liscore	Ágil	Moderado - Clássico
1	0 (0,00)	2 (0,07)
2	0 (0,00)	3 (0,10)
3	0 (0,00)	0 (0,00)
4	2 (0,20)	9 (0,31)
5	8 (0,80)	15 (0,52)
média	4,8	4,1
P = 0.383		

Tabela 89 - Questão 21 versus Questão 8

Escore	Gerenciamento de Projeto	
	Ágil	Moderado - Clássico
1	3 (0,30)	10 (0,34)
2	1 (0,10)	9 (0,31)
3	3 (0,30)	4 (0,14)
4	2 (0,20)	4 (0,14)
5	1 (0,10)	2 (0,07)
média	2,7	2,3
P = 0.610		

Tabela 90 - Questão 22 versus Questão 8

Escore	Gerenciamento de Projeto	
Liscore	Ágil	Moderado - Clássico
1	3 (0,30)	3 (0,10)
2	0 (0,00)	3 (0,10)
3	0 (0,00)	3 (0,10)
4	4 (0,40)	11 (0,38)
5	3 (0,30)	9 (0,31)
média	3,4	3,7
P = 0.413		

Tabela 91 - Questão 23 versus Questão 8

Escore	Gerenciamento de Projeto	
Liscore	Ágil	Moderado - Clássico
1	0 (0,00)	1 (0,04)
2	0 (0,00)	0 (0,00)
3	0 (0,00)	2 (0,07)
4	1 (0,10)	7 (0,25)
5	9 (0,90)	18 (0,64)
média	4,9	4,5
P = 0.501		

Tabela 92 - Questão 24 versus Questão 8

1 abeia 32 - Questao 24 versus Questao o		
Escore	Gerenciamento de Projeto	
Liscore	Ágil	Moderado - Clássico
1	0 (0,00)	1 (0,03)
2	0 (0,00)	2 (0,07)
3	0 (0,00)	2 (0,07)
4	3 (0,30)	9 (0,31)
5	7 (0,70)	15 (0,52)
média	4,7	4,2
P = 0.705		

Tabela 93 - Questão 25 versus Questão 8

Escore	Gerenciamento de Projeto	
Liscore	Ágil	Moderado - Clássico
1	0 (0,00)	2 (0,07)
2	0 (0,00)	1 (0,03)
3	0 (0,00)	1 (0,03)
4	0 (0,00)	8 (0,28)

	Gerenciamento de Projeto	
5	10 (1,00)	17 (0,59)
média	5,0	4,3
P = 0.201		

Tabela 94 - Questão 26 versus Questão 8

Tabela 74 - Questao 20 versus Questao 0		
Escore	Gerenciamento de Projeto	
Liscore	Ágil	Moderado - Clássico
1	2 (0,20)	2 (0,07)
2	1 (0,10)	11 (0,39)
3	0 (0,00)	6 (0,21)
4	6 (0,60)	8 (0,29)
5	1 (0,10)	1 (0,04)
média	3,3	2,8
P = 0,097		

Tabela 95 - Questão 27 versus Questão 8

Tabela 95 - Questao 27 versus Questao 8								
Escore	Gerenciamento de Projeto							
Liscore	Ágil	Moderado - Clássico						
1	0 (0,00)	2 (0,07)						
2	0 (0,00)	4 (0,14)						
3	0 (0,00)	6 (0,21)						
4	3 (0,30)	11 (0,38)						
5	7 (0,70)	6 (0,21)						
média	4,7	3,5						
	P = 0,046							

Tabela 96 - Questão 28 versus Questão 8

Tabeia 96 - Questao 28 versus Questao 8									
Escore	Gerenciamento de Projeto								
Escore	Ágil	Moderado - Clássico							
1	0 (0,00)	5 (0,17)							
2	1 (0,10)	0 (0,00)							
3	0 (0,00)	2 (0,07)							
4	1 (0,10)	9 (0,31)							
5	8 (0,80)	13 (0,45)							
média	4,6	3,9							
	P = 0,081								

Tabela 97 - Questão 29 versus Questão 8

Escore	Gerenciamento de Projeto							
Liscore	Ágil	Moderado - Clássico						
1	0 (0,00)	4 (0,14)						
2	0 (0,00)	4 (0,14)						
3	0 (0,00)	3 (0,10)						
4	3 (0,30)	10 (0,34)						
5	7 (0,70)	8 (0,28)						
média	4,7	3,5						
	P = 0.120							

Tabela 98 - Questão 30 versus Questão 8

Escore	Gerenciamento de Projeto						
Liscore	Ágil	Moderado - Clássico					
1	1 (0,10)	3 (0,10)					
2	0 (0,00)	3 (0,10)					
3	2 (0,20)	6 (0,21)					
4	1 (0,10)	13 (0,45)					
5	6 (0,60)	4 (0,14)					
média	4,1	3,4					
	P = 0.045						

Tabela 99 - Questão 31 versus Questão 8

Tabela 77 - Questao 31 versus Questao o								
Gerenciamento de Projeto								
Ágil	Moderado - Clássico							
2 (0,20)	5 (0,17)							
1 (0,10)	5 (0,17)							
0 (0,00)	5 (0,17)							
3 (0,30)	12 (0,41)							
4 (0,40)	2 (0,07)							
3,6	3,0							
P = 0.109								
	Ágil 2 (0,20) 1 (0,10) 0 (0,00) 3 (0,30) 4 (0,40) 3,6							

Tabela 100 - Questão 32 versus Questão 8

Escore	Gerenciamento de Projeto						
Liscore	Ágil	Moderado - Clássico					
1	0 (0,00)	1 (0,03)					
2	0 (0,00)	4 (0,14)					
3	0 (0,00)	2 (0,07)					
4	4 (0,40)	11 (0,38)					

	Gerenciamen	Gerenciamento de Projeto								
5	6 (0,60)	11 (0,38)								
média	4,6	3,9								
	P = 0,516									

Tabela 101 - Questão 33 versus Questão 8

Tabela 101 - Questao 35 versus Questao o									
Escore	Gerenciamento de Projeto								
Liscore	Ágil	Moderado - Clássico							
1	0 (0,00)	5 (0,17)							
2	1 (0,10)	3 (0,10)							
3	0 (0,00)	6 (0,21)							
4	1 (0,10)	9 (0,31)							
5	8 (0,80)	6 (0,21)							
média	4,6	3,3							
	P = 0,015								

Tabela 102 - Questão 34 versus Questão 8

Escore	Gerenciamento de Projeto						
Liscore	Ágil	Moderado - Clássico					
1	0 (0,00)	3 (0,10)					
2	0 (0,00)	2 (0,07)					
3	0 (0,00)	6 (0,21)					
4	1 (0,10)	12 (0,41)					
5	9 (0,90)	6 (0,21)					
média	4,9	3,6					
	P= 0,004						

#### ANEXO C – TABELAS E GRÁFICOS DA ANÁLISE DISCRIMINANTE

1) Tabelas e gráficos de análise discriminante para a Questão 7:

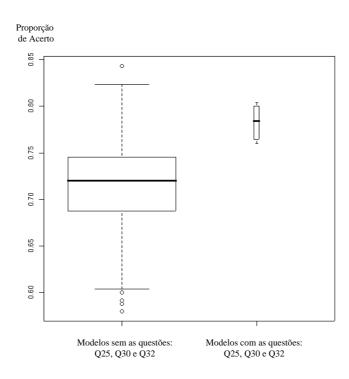


Gráfico 2 - Gráficos de boxplot da proporção de acerto para os 9.108 modelos (Q7).

O Gráfico 2 leva em consideração a presença ou não das questões Q25, Q32 e Q30 nos modelos de predição para a questão Q7. A largura de cada caixa é proporcional ao número de observações utilizadas para construí-la.

A Tabela 105 apresenta aos dez melhores modelos de predição para a Questão 7, dos 9.108 considerados, segundo a proporção de acertos.

Tabela 103 - Dez melhores modelos para a predição da Q7.

	Tabela 103 - Dez memores mouelos para a predição da Q7.													
Q13	Q14	Q16	Q21	Q25	Q28	Q29	Q30	Q31	Q32	Q33	Q34	Prop. acerto	Falsos positivos	n
0	0	0	0	1	1	0	0	1	1	0	0	0.843	6	51
0	1	0	0	1	0	0	1	0	0	0	0	0.824	8	51
1	1	0	0	1	0	0	1	0	0	0	0	0.824	8	51
0	1	0	0	1	1	0	1	0	0	0	0	0.824	8	51
0	1	0	0	1	0	1	1	0	0	0	0	0.824	8	51
0	1	0	0	1	0	0	1	1	0	0	0	0.824	7	51
0	1	0	0	1	0	0	1	0	0	1	0	0.824	8	51
0	1	0	0	1	0	0	0	0	1	0	1	0.824	8	51
0	0	1	0	1	0	0	0	0	1	0	1	0.824	8	51
0	0	0	1	1	0	0	0	1	1	0	0	0.824	7	51
1	7	1	1	10	2	1	6	3	4	1	2			soma

Tabela 104 - Resultados da classificação para Q7

Grupo Original	Grupo Predito					
Grupo Originai	Sucesso Parcial	Sucesso				
Sucesso Parcial	8 (47%)	9 (53%)				
Sucesso	2 (5%)	32 (95%)				

2) Tabelas e gráficos de análise discriminante para a Questão 8:

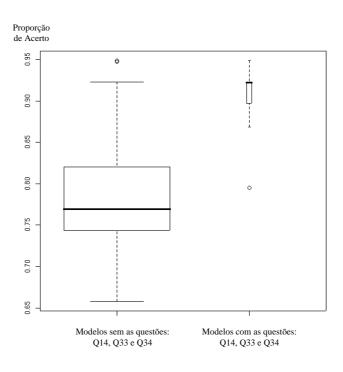


Gráfico 3 - Gráficos de boxplot da proporção de acerto para os 9.108 modelos (Q8)

O Gráfico 3 leva em consideração a presença ou não das questões Q14, Q33 e Q34 nos modelos de predição para a questão Q8. A largura de cada caixa é proporcional ao número de observações utilizadas para construí-la.

A Tabela 107 apresenta aos dez melhores modelos de predição para a Questão 8, dos 9.108 considerados, segundo a proporção de acertos.

Tabela 105 - Dez melhores modelos para a predição da Q8.

Q14	Q16	Q18	Q21	Q24	Q27	Q28	Q31	Q32	Q33	Q34	Prop. acerto	Falsos positivos	n
1	0	0	0	0	0	1	0	0	1	0	0.948	0	39
1	1	0	0	0	1	0	0	0	0	1	0.948	1	39
1	1	0	0	0	0	1	0	0	1	0	0.948	0	39
1	1	0	0	0	0	0	1	0	1	0	0.948	0	39
1	1	0	0	0	0	0	0	1	1	0	0.948	0	39
1	0	1	0	0	0	1	0	0	1	0	0.948	0	39
1	0	1	0	0	0	0	1	0	1	0	0.948	0	39
1	0	1	0	0	0	0	0	1	1	0	0.948	0	39
1	0	0	1	1	0	0	0	0	1	0	0.948	0	39

Q14	Q16	Q18	Q21	Q24	Q27	Q28	Q31	Q32	Q33	Q34	Prop. acerto	Falsos positivos	n
1	0	0	0	1	0	1	0	0	1	0	0.948	0	39
10	4	3	1	2	1	4	2	2	9	1			soma

Tabela 106 - Resultados da classificação para Q8

Cruno Original	Grupo Predito					
Grupo Original	Ágil	Moderado-Clássico				
Ágil	7 (70%)	3 (30%)				
Moderado-Clássico	0 (0%)	29 (100%)				

# ANEXO D – TABELA DA REGRESSÃO LOGÍSTICA

Tabela de regressão logística para a Questão 7:

Tabela 107 - Estimativas dos parâmetros no modelo logístico para Q7

Intercepto	Q25	Q31	Q32
4.338	1.679	0.831	1.159