

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
MESTRADO EM TECNOLOGIA

CECÍLIA KEIKO ADATI TOMOMITSU

IMPACTO DO USO DE MÉTODOS ÁGEIS NO PROCESSO DE ENSINO-APRENDIZAGEM
DE ENGENHARIA DE SOFTWARE

SÃO PAULO
MAIO/2008

CECÍLIA KEIKO ADATI TOMOMITSU

IMPACTO DO USO DE MÉTODOS ÁGEIS NO PROCESSO DE ENSINO-APRENDIZAGEM
DE ENGENHARIA DE SOFTWARE

Dissertação apresentada como exigência parcial para obtenção do Título de Mestre em Tecnologia no Centro Estadual de Educação Tecnológica Paula Souza, no Programa de Mestrado em Tecnologia: Gestão Desenvolvimento e Formação, sob orientação do Prof. Dr. Aristides Novelli Filho.

SÃO PAULO
Maio/2008

CECÍLIA KEIKO ADATI TOMOMITSU

IMPACTO DO USO DE MÉTODOS ÁGEIS NO PROCESSO DE ENSINO-APRENDIZAGEM
DE ENGENHARIA DE SOFTWARE



PROF. DR. ARISTIDES NOVELLI FILHO



PROF. DR. FRANCISCO EUGENIO BARRELLA



PROFª. DRª. MARILIA MACORIN DE AZEVEDO

São Paulo, 16 de maio de 2008.

Tomomitsu, Cecília Keiko Adati
T661i Impacto do uso de métodos ágeis no processo de ensino-aprendizagem de engenharia de software / Cecília Keiko Adati Tomomitsu. -- São Paulo : CEETEPS, 2008.
139 f.

Dissertação (Mestrado) – Centro Estadual de Educação Tecnológica Paula Souza, 2008.

1. Engenharia de software. 2. Ensino – aprendizagem. 3. Métodos ágeis. I. Título.

CDU 681.3.01:37

DEDICATÓRIA

A Deus, por tudo.

Aos meus filhos, Ricardo, Felipe e Henrique, pelo amor e a compreensão da minha ausência em alguns momentos de suas vidas.

Ao meu marido, Shiguelo, pelas contribuições, pelo incentivo, pelo companheirismo e dedicação.

Aos meus Pais, Takasi e Sanae, que sempre me ensinaram a importância de uma boa formação.

AGRADECIMENTOS

A toda a minha família pelo apoio e incentivo.

Ao Prof. Dr. Aristides Novelli Filho pelo grande apoio, incentivo e pela orientação durante a jornada que culmina com esta dissertação.

Ao Prof. Dr. Francisco Eugenio Barrella e a Prof^a. Dr^a Marília Macorin de Azevedo pelas importantes contribuições para a melhoria deste trabalho.

A todos os professores do Programa de Pós-Graduação que contribuíram para a solidificação das bases necessárias para a realização deste projeto de pesquisa.

A Cleonice pela presteza em todos os momentos.

Aos alunos que participaram desta empreitada provendo importantes subsídios para a concretização dos estudos .

A todas as pessoas que ajudaram direta ou indiretamente para a realização deste trabalho.

Muito obrigada a todos.

“Não existem métodos fáceis para resolver
problemas difíceis”

René Descartes

RESUMO

TOMOMITSU, C. K. A. **Impacto do Uso de Métodos Ágeis no processo de Ensino-Aprendizagem de Engenharia de Software**. 2008. 139 f. Dissertação (Mestrado em Tecnologia) - Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2008.

A comunidade envolvida com desenvolvimento de software tem enfrentado grandes dificuldades para entregar software de qualidade, no prazo contratado, com todos os requisitos tendo sido contemplados e de acordo com o orçamento aprovado. Em 1968, surge o termo engenharia de software, que, desde então, busca sistemáticas de trabalho para resolver o caráter crônico dessa atividade, que foi denominada de crise de software.

A busca de novas abordagens tem caracterizado a área de engenharia de software que busca sempre a melhora dos processos e produtos de software que tem no modelo clássico, também conhecido como cascata, uma das abordagens mais difundidas desde a década de 70, quando foi comunicada pela primeira vez. Esse paradigma tem orientado, além da comunidade empresarial, as escolas que têm, nessa abordagem, o referencial para o ensino da engenharia de software, nos seus cursos que envolvem o desenvolvimento de software.

Novas abordagens ou derivadas do modelo cascata têm surgido, mas herdaram o seu rigor formal, entre outras características. No final da década de 90, surgem os métodos denominados ágeis, que tiveram grande impulso na sua difusão a partir de 2001, com a publicação do manifesto ágil. Esses métodos caracterizam-se por valores, princípios e práticas que se orientam por uma perspectiva diferente da abordagem tradicional. Enquanto a abordagem tradicional dá ênfase para o processo e documentação rigorosa, a abordagem ágil foca pessoas, iterações curtas com entrega de produtos e leveza do processo.

Diante dessas novas abordagens, este trabalho teve como objetivo verificar o impacto da adoção dos métodos ágeis, no processo de ensino-aprendizagem de engenharia de software, no âmbito dos cursos de tecnologia em processamento de dados.

Os resultados indicam que a adoção da abordagem ágil resulta em melhora no desempenho do aprendiz e na construção dos produtos de software. Outro fato relevante consiste na importância do pré-conhecimento de outras abordagens, para que se possam obter os ganhos esperados da nova abordagem.

Palavras-chave: Métodos ágeis, Engenharia de Software, Modelagem ágil

ABSTRACT

TOMOMITSU, C. K. A. **Impacto do Uso de Métodos Ágeis no processo de Ensino-Aprendizagem de Engenharia de Software.** 2008. 139 f. Dissertação (Mestrado em Tecnologia) - Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2008

The community involved with software development has been facing great difficulties to deliver quality software, in the agreed term, with all the requirements contemplated and according to the approved budget. In 1968 the term software engineering appeared, which, since then, has searched for work's systematic to solve the chronic character of such activity, denominated software crisis.

The search for new approaches has characterized which always looks for improvements for software process and product which has in the classical model, also known as "waterfall", one of the most known approaches since the 70s, when it was communicated for the first time. That paradigm has been guiding, besides the managerial community, schools which have in this approach a reference for the software engineering teaching in courses that involve software development.

New approaches or derivatives of the waterfall model have appeared, but they inherit its formal rigidity, among other characteristics. At the end of the 90s the so-called agile methods appear, which had a great impulse in its diffusion starting from 2001, with the publication of the agile manifesto. These methods are characterized by values, principles and practices that are guided by a different perspective from the traditional approach. While the traditional approach emphasizes the process and rigorous documentation, the agile approach focus on people, short iterations with the product delivery and the lightness of the process.

Considering these new approaches, the aim of this study is to verify the impact of the adoption of the agile methods in the teaching-learning process of software engineering in courses of data processing technology.

The results indicate that the adoption of the agile approach results in an improvement of the apprentice's performance and in the software products construction. Another important fact consists in the importance of the pre-knowledge of other approaches so that one can obtain the earnings expected from the new approach.

Key words: Agile methods, Software Engineering, Agile modeling.

LISTA DE FIGURAS

Figura 1 – Modelo de Ciclo de Ciclo de Vida Cascata	23
Figura 2 – Ciclo de Vida de um projeto XP	29
Figura 3 – Práticas do XP	30
Figura 4 – Visão Geral do processo SCRUM	33
Figura 5 – Performance dos projetos de TI no período de 1994 a 2004	39
Figura 6 – Atraso médio na entrega de projetos	47
Figura 7 – Projetos que ultrapassaram o valor inicialmente orçado	48
Figura 8 – Fases da MDMS	52
Figura 9 – Níveis de abstração	53
Figura 10 – Nomenclatura adotada nos estudos de caso	59
Figura 11 – Projeto 1 - Funcionalidades implementadas (tradicional)	62
Figura 12 – Projeto 1 - Funcionalidades implementadas (ágil)	62
Figura 13 – Projeto 1 - Comparativo entre as abordagens	63
Figura 14 – Projeto 2 - Funcionalidades implementadas (tradicional)	66
Figura 15 – Projeto 2 - Funcionalidades implementadas (ágil)	66
Figura 16 – Projeto 2 - Comparativo entre as abordagens	67
Figura 17 – Projeto 3 - Comparativo entre as abordagens	71
Figura 18 – Projeto 3 - Implementação sem o uso de software gerador	71
Figura 19 – Projeto 3 - Implementação com o uso de software gerador	72
Figura 20 – Projeto 3 - Implementação ágil (com e sem uso de gerador)	72
Figura 21 – Projeto 3 - Implementação tradicional (com e sem uso de gerador)..	72
Figura 22 – Projeto 3 - Adequação da abordagem em sala de aula	74
Figura 23 – Projeto 3 - Facilidade de aprendizagem	75
Figura 24 – Projeto 3 - Facilidade de execução	75
Figura 25 – Projeto 3 - Eficiência na realização do desenvolvimento	75
Figura 26 – Projeto 4 - Comparativo entre as abordagens – Módulo de compras	78
Figura 27 – Projeto 4 - Comparativo entre as abordagens – Módulo de vendas ..	78
Figura 28 – Projeto 4 - Comparativo entre as abordagens – Módulo Financeiro .	79
Figura 29 – Projeto 4 - Comparativo entre as abordagens – Módulo Produção ...	79
Figura 30 – Projeto 4 - Comparativo entre as abordagens – Geral	79
Figura 31 – Projeto 4 - Adequação da abordagem em sala de aula	83

Figura 32 – Projeto 4 - Facilidade de aprendizagem.....	83
Figura 33 – Projeto 4 - Facilidade de execução	83
Figura 34 – Projeto 4 - Eficiência na realização do desenvolvimento	84

LISTA DE QUADROS

Quadro 1: Comparativo : abordagem ágil x abordagem tradicional	34
Quadro 2 - Citações sobre a problemática de Engenharia de Software	41
Quadro 3: Citações sob a perspectiva das abordagens metodológicas	42
Quadro 4: Citações sob a perspectiva dos stakeholders	44
Quadro 5: Citações sob a perspectiva do tempo	46
Quadro 6: Citações sob a perspectiva do custo	48
Quadro 7: Citações sob a perspectiva das ferramentas	49
Quadro 8: Projeto 1 - Funcionalidades implementadas	61
Quadro 9: Projeto 2 - Funcionalidades implementadas	65
Quadro 10: Projeto 3 - Distribuição dos grupos	69
Quadro 11: Projeto 3 - Funcionalidades implementadas.....	70
Quadro 12: Projeto 4 - Funcionalidades implementadas	80

LISTA DE ABREVIATURAS E SIGLAS

APS – Análise e Projetos de Sistemas

DFD – Diagrama de Fluxo de Dados

FATEC/SP – Faculdade de Tecnologia de São Paulo

KB – Knowledge Base

MDMS – Metodologia de Desenvolvimento e Manutenção de Sistemas

PDCA – Plan, Do, Check, Action

XP – Extreme Programming

SUMÁRIO

Introdução.....	14
1. Engenharia de software.....	21
1.1. A Abordagem Tradicional	22
1.2. A Abordagem Ágil	24
1.3. Comparativo: Abordagem Ágil X Abordagem Tradicional	34
1.4. Conclusão.....	36
2. A problemática do desenvolvimento de software: uma retrospectiva.....	38
2.1. Sob a perspectiva das abordagens metodológicas	41
2.2. Sob a perspectiva dos stakeholders	43
2.3. Sob a perspectiva do tempo	46
2.4. Sob a perspectiva do custo	47
2.5. Sob a perspectiva das ferramentas	48
2.6. Conclusão	49
3. O ambiente de estudo de engenharia de software na FATEC-SP	50
3.1. A adequação do modelo clássico sob a perspectiva da aprendizagem	51
3.2. As estratégias em sala de aula, atividades de laboratório e extra-sala	51
3.3. A abordagem tradicional e o processo de ensino-aprendizagem	55
3.4. Conclusão.....	56
4. Impactos do uso dos métodos ágeis no processo de ensino e aprendizagem de engenharia de software.....	58
4.1. Estudos de Caso.....	58
4.2. Coleta de Dados	59
4.3. Impacto das práticas propostas pelo XP (extreme programming)	60
4.4. Impacto do uso de métodos ágeis associado ao uso de software gerador de aplicações	68
4.5. Impacto do uso de métodos ágeis em relação à gestão do processo.....	76
4.6. Conclusão.....	85
Considerações Finais	86
Referências Bibliográficas	89
Apêndices	93

Introdução

Na economia moderna, é freqüentemente difícil ou impossível prever como um sistema baseado em computador evoluirá com o passar do tempo. Condições de mercado mudam rapidamente, necessidades dos usuários finais evoluem, e novas ameaças de competição emergem sem alerta (Pressman, 2006).

A celeridade é a marca dos nossos dias (Castells, 2000, p. 51). A sociedade impõe e lhe são impostos ritmos cada dia mais alucinantes, rápidos, enfim, céleres para quase tudo que a envolve.

Não basta fazer, tem-se que fazer a cada dia mais rápido. A eficiência, a produtividade, o tempo encurtado constituem a dimensão que expressa o modo de vida corporativo atual.

O prazo de entrega, somente, não atende mais já há algum tempo. Não resolve. O paradigma é o tempo de resposta, ou seja, as entregas das informações de horas ou dias passaram a alguns segundos, em tempo real, visando a tomada de decisões, para suprir as demandas internas e do mercado (Castells, 2006).

Atributos como qualidade não são mais diferenciais, são itens que permitem as organizações habilitarem-se a competir.

As tecnologias, novas e aquelas permanentemente melhoradas, contribuem para que a dimensão tempo tenha a sua unidade de medida reduzida.

O universo do mundo corporativo é conduzido tendo como um dos diferenciais, e que lhe permite um posicionamento competitivo, a capacidade de responder ao mercado o mais rápido possível.

Observa-se que até o ciclo de vida de desenvolvimento dos produtos – qualquer que seja – tem sido significativamente encurtado. Esse “encurtar” dos ciclos de vida reflete, também, nos ciclos de vida dos artefatos e produtos de software, e exige dos desenvolvedores a adoção de um posicionamento compatível frente a esse cenário, ou seja, reduzir os tempos de desenvolvimento.

O “encurtar” o ciclo de vida de desenvolvimento pode ser obtido pela:

- eliminação ou redução de tarefas;
- execução simultânea de tarefas;
- simplificação das tarefas, componentes e do produto final;

- obtenção de produtividade máxima, se ainda não tiver sido alcançada;
- utilização de recursos e ferramentas que contribuam para a maior produtividade;
- adoção de abordagens metodológicas orientadas para essa finalidade.

As coisas céleres permeiam quase tudo que nos rodeia e influenciam a busca por mais velocidade na realização das atividades, com o intuito de reduzir os tempos de desenvolvimento, produção e entrega de produtos. O tempo é crucial para a geração de lucros em todo o sistema (Castells, 2000, p.462).

Objetivo Principal

O objetivo deste projeto de pesquisa é avaliar o impacto, nos processos de ensino-aprendizagem de estudantes de cursos de graduação em Tecnologia de Processamento de Dados, da utilização de métodos ágeis associados à utilização de softwares geradores de código, comparados à utilização de métodos e ambientes de desenvolvimento/programação tradicionais e, nesse contexto, identificar se a adoção dos métodos ágeis contribui para que os alunos possam, sistematicamente, a partir dos conhecimentos e habilidades adquiridas, solucionar problemas que necessitem a aplicação da engenharia de software.

Objetivos Específicos

Identificar indicadores que demonstrem o grau de eficácia e eficiência dos elementos participantes do processo de ensino-aprendizagem de engenharia de software, ou seja:

- dos estudantes, enquanto aprendizes;
- dos métodos tradicional e ágil;
- das ferramentas de apoio ao processo de desenvolvimento de software.

Justificativa

O atendimento aos prazos, custos e funcionalidades estabelecidos pelas partes contratantes do desenvolvimento de um sistema, e dos demais requisitos especificados, constitui-se no grande desafio dos desenvolvedores de sistemas. Nesse contexto, a busca de caminhos que permitam agilizar o desenvolvimento, mantendo a qualidade exigida do produto final, deve ser empreendida pelos desenvolvedores.

Nesse cenário, a área de Engenharia de Software, que tem como principal preocupação a melhoria do processo e produto de software, vem buscando alternativas formais que possam responder aos requisitos especificados, em especial quanto aos prazos acordados. Surgem, nesse contexto, os “Métodos Ágeis” como o Extreme Programming (XP), Scrum e Crystal, entre outros, popularizados recentemente pela divulgação do “manifesto ágil”. O manifesto é definido por quatro simples declarações de valores que definem preferências, não alternativas, encorajando o enfoque de certas áreas, mas sem eliminar outras. Os valores do Manifesto Ágil são: indivíduos e interações valem mais que processos e ferramentas, um software funcionando vale mais que documentação extensa, a colaboração do cliente vale mais que a negociação de contrato, responder a mudanças vale mais que seguir um plano.

Outro recurso importante para a redução do tempo de desenvolvimento de produtos de software, e conseqüente contribuição para a entrega nos prazos acordados, é a utilização de software gerador de código, que vem sendo adotado por algumas organizações, que anseiam conquistar maior velocidade na produção de produtos de software.

Contextualizar, de forma adequada – eficaz e eficientemente – essas questões em sala de aula, constitui-se no desafio que deve ser empreendido para que se possa contribuir para o efetivo aprendizado por parte dos estudantes. Desafio, porque esses aprendizes, na sua maior parte, não têm a experiência e maturidade suficiente, naquele momento, para ter os pontos apresentados como sendo de real importância.

Associe-se a isso o fato das abordagens metodológicas “ensinadas” e “aplicadas” para desenvolvimento de sistemas serem de ciclo longo¹. Ressalte-se que as disciplinas semestrais dos cursos de graduação de tecnologia de processamento de dados, além da ênfase às teorias, destacam-se na ênfase ao saber fazer, ou seja, no desenvolvimento das habilidades dos aprendizes. Isso exige, por parte dos aprendizes, a experimentação de um ciclo de desenvolvimento de sistemas completo, produzindo sistemas de qualidade, na forma, conteúdo e resultados de processos com níveis adequados de formalidade.

Considerando a aprendizagem da engenharia de software sob a ótica do aprendiz, buscou-se, neste projeto de pesquisa, identificar variáveis que impactam no processo de aprendizagem e corroborem para que seja alcançada a assimilação esperada, ou seja, que a aprendizagem seja evidenciada pela explicitação da base conceitual adquirida e, tão importante quanto, as habilidades – o saber fazer – adquirido e desenvolvido por cada um. Isso é alcançado quando o aprendiz é capaz de desenvolver, preparar, modificar e aplicar tudo aquilo que foi aprendido, através dos estudos e práticas exigidos nas disciplinas que compõem a engenharia de software.

Assim sendo, foi contemplado o impacto da adoção do ensino de métodos ágeis no processo de aprendizagem, estando este projeto de pesquisa centrado no impacto desses métodos, sob a perspectiva do aprendiz, com o intuito de identificar o quanto essa mudança de paradigma contribui para isso, o que justifica este projeto de pesquisa.

Delimitação de estudo

Este projeto de pesquisa contempla a variável Metodologia de Desenvolvimento e Manutenção de Sistemas (MD MS) que é ensinada aos alunos de graduação do Curso de Tecnologia em Processamento de Dados, da Faculdade de Tecnologia de São Paulo (FATEC-SP), e utilizada por eles nas atividades práticas exigidas nas disciplinas Análise e Projeto de Sistemas I (APS I), Análise e Projeto de Sistemas II (APS II) e Análise e Projeto de Sistemas III (APS III), que contribuem

¹ O termo ciclo longo é utilizado em oposição a ciclo encurtado, obtido a partir do uso de métodos ágeis.

diretamente para que ocorra a oportunidade da aprendizagem por esses alunos. Embora haja outros assuntos importantes como métodos pedagógicos, ementas e respectivos planos de ensino e seus tópicos, competências e habilidades do professor, entre outros, buscar-se-á isolá-los de maneira a permitir que este projeto de pesquisa alcance os seus resultados, considerando o foco delimitado por seu próprio título: “O impacto do uso de Métodos Ágeis no processo de ensino-aprendizagem de Engenharia de Software”.

Metodologia

Este trabalho, sob o ponto de vista da sua natureza, caracteriza-se por ser uma pesquisa aplicada, por ter o interesse de gerar conhecimentos para aplicação prática e dirigidas à solução de problemas específicos, orientado pelos interesses que envolvem a obtenção de estratégias e melhores abordagens sobre modelos de ciclo de vida de desenvolvimento de sistemas a serem adotados no processo de ensino e aprendizagem.

Do ponto de vista da forma de abordagem do problema, caracteriza-se por ser uma pesquisa qualitativa, uma vez que a fonte de dados foi – essencialmente – o ambiente acadêmico.

Caracteriza-se, também, por ser de caráter descritivo, ou seja, de levantamento, por ter sido conduzida a partir de entrevistas, observações in loco, reuniões de acompanhamento, não deixando de buscar as fundamentações ou referências.

O Estudo de Caso foi a conduta escolhida para a realização deste trabalho.

Estrutura do Trabalho

Esta dissertação está estruturada da seguinte forma:

Introdução

Esta seção apresenta a justificativa do trabalho, os objetivos, a delimitação do estudo e, especialmente, as contribuições da pesquisa do tema proposto para as comunidades docente e discente.

Engenharia de Software

Este capítulo apresenta alguns aspectos do processo de desenvolvimento de software de interesse para este trabalho. Inicia com o modelo clássico de desenvolvimento, denominado tradicional, cujo paradigma constitui-se na base de outras abordagens que dela derivam. Apresenta características que identificam a abordagem ágil e as similaridades representativas encontradas nesses métodos. Ressalta as características de cada abordagem por categoria e conclui sobre a aplicação, o uso das mesmas.

A problemática do desenvolvimento de software: uma retrospectiva

Neste capítulo é apresentada a problemática do desenvolvimento de software, com o objetivo de explicitar o caráter crônico que tem acompanhado o processo de desenvolvimento sob diversas perspectivas, que endereça a abordagem metodológica como uma das perspectivas que tem contribuído para a situação apresentada.

O ambiente de estudo da Engenharia de Software na FATEC-SP

Este capítulo apresenta o modelo de ciclo de vida que orienta as atividades em sala de aula e as atividades práticas realizadas em laboratório, que constituem os padrões ou referências que os alunos seguem no desenvolvimento de sistemas. O modelo clássico e outras abordagens, cujas características são de certa forma herdadas desse ciclo de vida, são aqui referenciadas.

Impactos do uso dos métodos ágeis no processo de ensino e aprendizagem de Engenharia de Software

Os fundamentos metodológicos que nortearam o desenvolvimento do trabalho são descritos nesta seção. São abordados a descrição do universo da pesquisa, os instrumentos de coleta de dados, os projetos acadêmicos, os resultados, a análise dos resultados obtidos e a conclusão do estudo de caso.

Considerações Finais

Apresenta as considerações finais obtidas do trabalho de pesquisa realizado e indicações de estudos e pesquisas futuras que poderão ser empreendidas sobre o tema.

1. Engenharia de Software

John Tukey, em 1958, criou o termo *software* e, dez anos depois, em 1968, o termo *software engineering* foi utilizado na Conferência sobre Engenharia de Software da OTAN (Swebok, 2004).

A engenharia de software surgiu num contexto onde a crise de software se apresentava e, decorrente disso, sistemáticas de trabalho mais consistentes e formais, inspiradas na engenharia, foram concebidas para solucionar os problemas que tendiam a ser, a cada dia, maiores e mais complexos e que acompanhavam a comunidade de desenvolvedores e stakeholders² ao longo dos anos, de forma crônica (Pressman, 1995).

Boehm (2006) apresenta a evolução da engenharia de software desde a década de 50, em que destaca:

- na década de 50, a engenharia de software tendia a ser parecida com a engenharia de hardware;
- na década de 60, o processo de software era considerado arte;
- na década de 70, o desenvolvimento caracterizou-se pelo processo cascata, conhecido também como waterfall, e pela formalidade;
- na década de 80, o processo de software privilegiou a produtividade e a escalabilidade;
- na década de 90, caracterizou-se pelos processos concorrentes, ao invés de apenas processos seqüenciais;
- nos anos 2000, o processo de software orienta-se pela agilidade, os métodos ágeis.

No contexto deste trabalho, a abordagem tradicional, modelo cascata, formalmente empregada desde a década de 70, e a abordagem ágil são a seguir apresentadas e constituem-se no foco deste projeto de pesquisa.

² O termo “stakeholders” foi criado para designar todas as pessoas ou empresas que, de alguma maneira, são influenciadas pelas ações de uma organização.

1.1. A Abordagem Tradicional

O modelo de ciclo de vida clássico ou cascata, descrito pela primeira vez por Winston W. Royce em 1970 (Royce, 1970; Gustafson, 2002), que denominaremos a partir deste ponto como tradicional, caracteriza-se pelo seu caráter preditivo, prescritivo, seqüencial, burocrático, rigoroso, orientado a processos e dados, formal e controlado, que tem o sucesso alcançado desde que esteja em conformidade com o que foi planejado (Pressman, 2006; Paula Filho, 2003). O modelo de ciclo de vida cascata é apresentado na figura 1.

O modelo apresentado por Royce tem servido de base para outras abordagens, como o modelo baseado em prototipação, o modelo incremental e o modelo espiral de Bhoem (Gustafson, 2002) e tem sido, ainda, largamente utilizado (Rozanski, 2005) tanto no desenvolvimento de novos sistemas quanto para a manutenção dos sistemas legados.

Essa metodologia de desenvolvimento e manutenção de sistemas orienta o aprendiz na realização de todos os processos e produção dos respectivos artefatos, servindo-lhe como um guia que o dirige até o final do projeto. Essa característica prescritiva constitui-se num roteiro que permite ao aprendiz uma visibilidade de todas as fases do projeto.

O rigor formal, em especial no que se refere à documentação, é uma das características que permeiam essa abordagem. Privilegia-se a elaboração de uma boa documentação de todos os elementos obtidos e produzidos em todas as fases dos projetos, além dos respectivos artefatos. Tudo aquilo que é documentado como pertencente ao escopo do projeto é considerado objeto do desenvolvimento.

Outro aspecto que sobressai é a rigidez, uma vez que as especificações pré-definidas devem ser seguidas de forma rigorosa.

Nessa abordagem, a modelagem do negócio e sistemas é obtida privilegiando-se a percepção da realidade a partir dos processos. Adota-se, também, a modelagem semântica, criada por Peter Chen, que privilegia a percepção da realidade a partir dos dados (Chen, 1990; Setzer, 2005).

As ações de controle são facilitadas, pois é possível verificar a completude dos artefatos produzidos, uma vez que se tem previamente todos os artefatos que devem ser gerados ao longo do processo de desenvolvimento de sistemas.

Os modelos prescritivos fornecem estabilidade, controle e organização a uma atividade que pode, se deixada sem controle, tornar-se bastante caótica (Pressman, 2006).

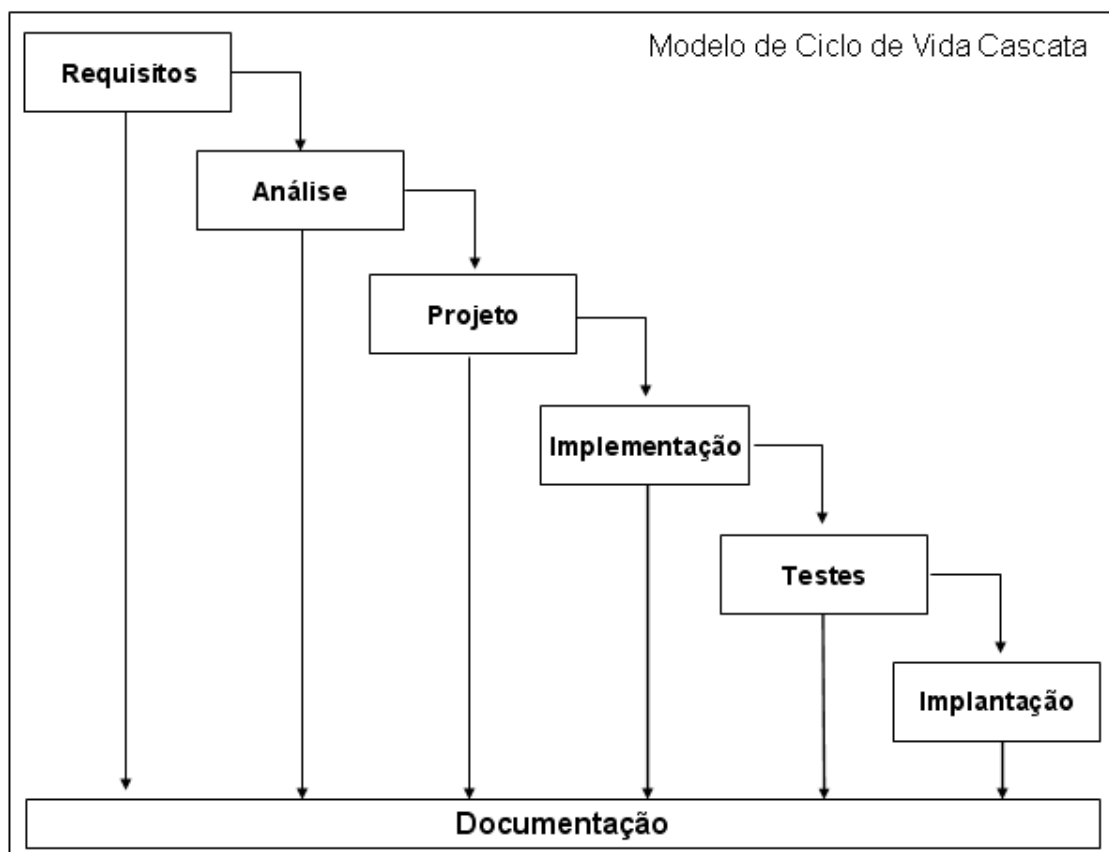


Figura 1. Modelo de Ciclo de Vida Cascata
Fonte: Adaptado de Winston W. Royce

1.2. A Abordagem Ágil

Pressman (2006) ressalta a importância do Software no cenário mundial, uma vez que este afeta praticamente todos os aspectos de nossas vidas e se tornou difundido no comércio e nas nossas atividades do dia-a-dia. Além disso, ressalta o duplo papel do software: o de produto e, ao mesmo tempo, o de veículo para entrega do produto. Como produto, ele disponibiliza o potencial de computação presente no hardware do computador. Como veículo usado para entrega do produto, o software age como base para o controle do computador, o sistema operacional, para a comunicação da informação e para a criação e o controle de outros programas.

Nesse contexto, as soluções inovadoras e criativas que suportam os novos paradigmas são possíveis graças à evolução tecnológica que tem permitido a sua própria presença como parte das soluções. Mais do que parte, estão “fundidas” a essas soluções, de tal modo que ao retirá-las, as mesmas deixam de existir (Poppendieck, 2007).

Surgem as células de produção independente, interconectadas, plenamente integradas, apenas fisicamente distantes, compondo novas formas de organização.

Num contexto onde a tecnologia se funde às soluções (Castells, 2000, p. 87), podemos afirmar que o que se constitui como elemento diferencial está na capacidade de usar eficaz e criativamente esses recursos tecnológicos. Sendo assim, o diferencial está nas pessoas que, com competência e criatividade, constroem soluções inovadoras e diferenciadas.

Pressman, 2006 p. 701, cita:

“Software de computador é uma entre poucas tecnologias importantes que terão um impacto significativo em praticamente todo o aspecto da sociedade moderna... É um mecanismo para automatizar negócios, indústria e governo, um meio para transferir nova tecnologia, um método para captar conhecimento (expertise) valioso para ser usado por outros, um modo de diferenciar os produtos de uma empresa de seus competidores e uma janela para o conhecimento coletivo de uma corporação. O software é central para quase todo aspecto do negócio. E de muitos modos, o software também é uma tecnologia oculta. Encontramos software (freqüentemente sem perceber) quando viajamos em serviço, quando fazemos qualquer compra

no varejo, quando passamos pelo banco, quando damos um telefonema, quando consultamos o médico ou realizamos qualquer uma das centenas de atividades cotidianas que refletem a vida moderna”.

Os recursos tecnológicos são “commodities” (Pisello, 2003, p.6) – produtos que estão disponíveis para venda e compra - que, por si, não fazem mais a diferença, apenas permitem aos competidores competir. São commodities o hardware, o software básico, os utilitários e os “sistemas prontos”. Aqueles recursos tecnológicos que não estão à disposição de qualquer interessado, ou seja, são acessíveis apenas aos seus proprietários e para aqueles que têm a sua posse para uso não são considerados commodities.

Considerando a tecnologia uma commodity, o diferencial estará na capacidade de realizar, com processo otimizado e suportado por diversas tecnologias, produtos únicos para clientes únicos. Criatividade e competências constituem-se nos elementos essenciais que permitirão o diferencial competitivo, num mundo caracterizado pela velocidade.

A velocidade tem sido uma das marcas da sociedade moderna. Os produtos têm seu ciclo de vida encurtado. Esse ritmo acelerado da evolução tecnológica, das organizações e do ambiente de negócios de maneira geral (Boehm, 2006, p. 19) tem incentivado a busca de novas abordagens que dão ênfase à agilidade para um melhor posicionamento competitivo. A competição acirrada tem exigido respostas cada dia mais rápidas por todos os competidores. As tecnologias têm tido aumento significativo nas suas capacidades e rapidez. Nesse contexto, onde a velocidade constitui-se num dos ingredientes para o alcance de uma melhor posição competitiva pelas organizações, é essencial que os processos permitam produzir com a necessária celeridade. Assim, o requisito prazo impõe a adoção de processos ágeis para a obtenção de produtos, com cronogramas cada vez menores, sendo assim condição necessária.

1.2.1. Manifesto Ágil

Em fevereiro de 2001, um grupo de 17 metodologistas formou a Agile Software Development Alliance (www.agilealliance.org), freqüentemente citada

apenas como Agile Alliance, para enfrentar os diversos desafios encontrados pelos desenvolvedores de software. Este grupo de pessoas definiu um manifesto para encorajar melhores meios de desenvolver software e, com base nesse manifesto, formulou um conjunto de princípios que definem critérios para os processos de desenvolvimento ágil de software.

Pressman (2006, p.58) define a engenharia de software ágil como uma combinação de filosofia e um conjunto de diretrizes de desenvolvimento. A filosofia encoraja a satisfação do cliente e a entrega incremental do software logo de início; equipes de projeto pequenas, altamente motivadas; métodos informais; produtos de trabalho de engenharia de software mínimos e simplicidade global do desenvolvimento. As diretrizes de desenvolvimento enfatizam a entrega, em contraposição à análise e ao projeto, e a comunicação ativa e contínua entre desenvolvedores e clientes.

1.2.2. Processos Ágeis de Desenvolvimento

Os métodos ágeis caracterizam-se pelo seu caráter adaptativo e orientado para pessoas. São várias as metodologias que são classificadas como ágeis, em que se destacam o XP (Extreme Programming) e Scrum, que são os processos utilizados neste projeto de pesquisa.

Segundo Teles (2004), os Processos Ágeis de Desenvolvimento compartilham a premissa de que o cliente aprende sobre suas necessidades, na medida em que é capaz de manipular o sistema que está sendo produzido e, com base no feedback do sistema, ele reavalia as suas necessidades e prioridades, gerando mudanças que devem ser incorporadas ao software. O aprendizado é importante, porque permite que o cliente direcione o desenvolvimento de modo que a equipe produza sempre aquilo que tem o maior valor para o seu negócio.

As abordagens ágeis compartilham, na sua essência, o processo de desenvolvimento centrado nas pessoas, orientado para a obtenção de artefatos a partir de iterações, o que, conseqüentemente, impõe o caráter adaptativo durante todo o ciclo de desenvolvimento.

Tate (2006, p. 29-31) apresenta duas perspectivas sobre o desenvolvimento ágil de software: o desenvolvimento orientado a visão e o desenvolvimento orientado

a princípios.

Tate destaca os seguintes aspectos orientados a visão:

- um usuário precisa ver o que está sendo construído;
- refinamentos devem ser rápidos para atendimento às mudanças;
- um relacionamento próximo com os usuários para a obtenção de um “feedback” rápido;
- o aprendizado deve ser contínuo para refinar o produto, para atender as necessidades efetivas dos usuários e evitar o que for desnecessário.

Sob a perspectiva dos princípios destaca:

- refinamento contínuo do produto e das práticas de projeto;
- artefatos/produtos devem ser liberados continuamente, ou seja, todo o tempo;
- investimentos contínuos no projeto do software;
- valorização na prevenção de defeitos ao invés de privilegiar a detecção.

Para melhor compreensão do que seja o desenvolvimento ágil, os membros da Agile Alliance (<http://www.agilemanifesto.org/principles.html>) apresentaram à comunidade os seguintes princípios:

- a maior prioridade é satisfazer o cliente, a partir de entregas de produtos de software de efetivo valor em tempo hábil e continuamente;
- acatar as necessidades de mudanças em qualquer estágio do processo de desenvolvimento, pois o software deve prover efetiva vantagem competitiva ao cliente;
- os artefatos ou produtos de software devem ser entregues no menor tempo possível e funcionando;
- todos os envolvidos devem trabalhar efetivamente juntos, diariamente, durante o projeto;
- a confiança deve nortear o processo de desenvolvimento e a construção dos produtos de software deve estar alinhada aos indivíduos mais motivados;
- privilegiar a comunicação direta, cara a cara;

- a percepção da evolução do projeto se dá através de produtos de software entregues e funcionando;
- os stakeholders devem ser capazes de manter o ritmo necessário durante o processo de desenvolvimento;
- a agilidade é conquistada pelo constante aprimoramento técnico e a consequente qualidade do projeto;
- a simplicidade deve ser a característica marcante dos projetos;
- decorrem de equipes organizadas as melhores práticas;
- avaliações regulares devem ser empreendidas para tomadas de decisões para o ajustamento ou adaptações necessárias.

1.2.2.1. Extreme Programming (XP)

Extreme Programming, ou XP, criado por Kent Beck em 1997, é um processo de desenvolvimento de software classificado como método ágil. É o mais popular (Cecil Martin, 2003, p.11), e o seu uso é recomendado em projetos que tenham requisitos vagos e instáveis (Teles, 2004, p. 21).

O XP também se caracteriza por ser um processo de desenvolvimento que busca assegurar que o cliente receba, o mais rapidamente possível, e frequentemente, os produtos de software produzidos pela equipe de desenvolvimento, permitindo o uso e consequente aprendizado, favorecendo o rápido feedback e reorientação do que for necessário para atendimento às necessidades e prioridade estabelecida. O ciclo de vida do projeto XP é mostrado na figura 2.

Essa abordagem dá ênfase ao trabalho em equipe (Rabello, 2006, p.61) e aos papéis que devem ser exercidos pelos participantes do grupo de trabalho. Os papéis de cliente, desenvolvedor, treinador, redator técnico, analista de testes e de gerente de projeto podem ser exercidos por uma mesma pessoa (Teles, 2004, p.28).

A proximidade das pessoas que exercem os diversos papéis é essencial, para que possam trocar idéias e informações rapidamente. Quanto mais ágil a comunicação entre os membros da equipe, mais rápido o feedback e as respectivas ações.

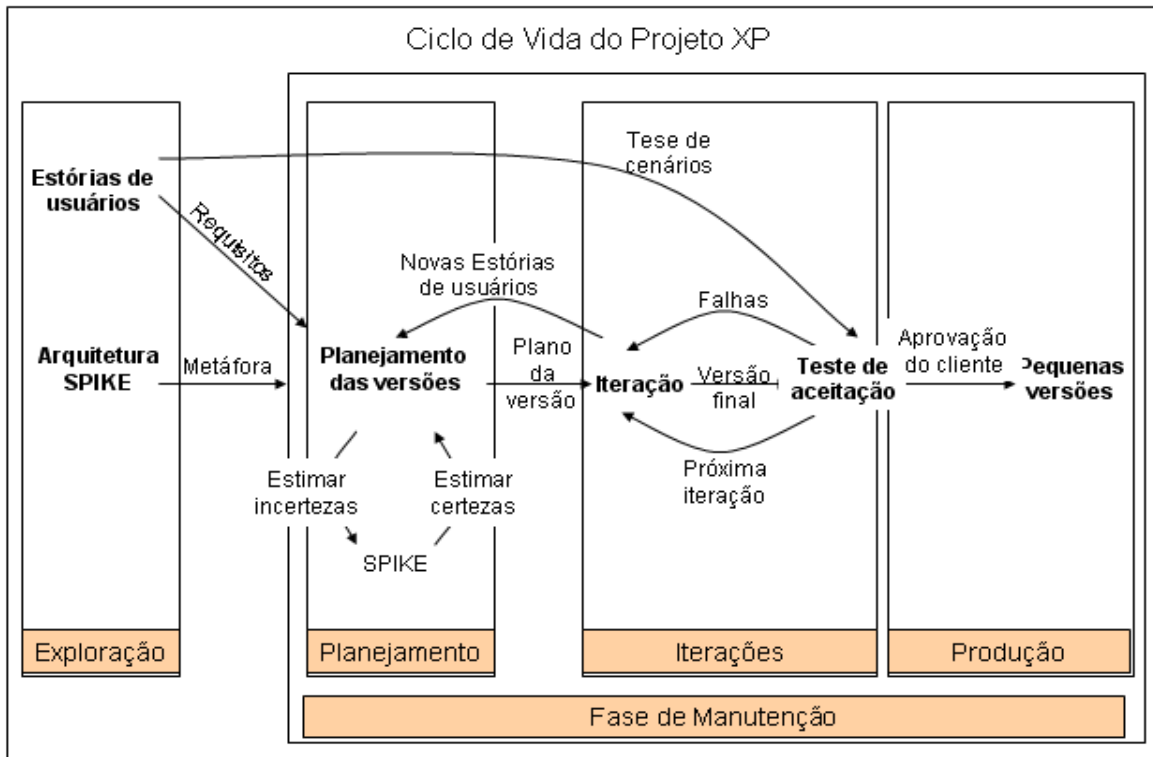


Figura 2: Ciclo de Vida de um projeto XP

Fonte: www.agilemodeling.com/essays/agileModelingXPLifecycle.htm

O XP se baseia em quatro valores fundamentais: Feedback, Comunicação, Simplicidade e Coragem (Teles, 2004, p. 44). Esses valores evidenciam a importância do Ser Humano, das pessoas, na execução dos projetos ao adotarem esse método. Esses valores são dependentes diretamente das iniciativas das pessoas que participam dos projetos.

O feedback que o cliente oferece aos desenvolvedores e vice-versa é altamente incentivado e necessário e só acontece se as pessoas se mobilizarem para essa finalidade. Quanto mais rapidamente for oferecido feedback à outra parte, mais ágil será a entrega dos produtos de software.

Da mesma forma, a comunicação deve ser exercida de forma intensa, visando propiciar a todos os envolvidos obterem informações e/ou transmiti-las de forma adequada, de tal modo que não ocorram desvios ou ruídos. Privilegia-se a comunicação face a face, o que justifica a indicação de equipes reduzidas para a execução dos projetos.

A orientação para a obtenção de soluções simples constitui-se num grande desafio, uma vez que realizar o simples não significa fazer algo com menor

complexidade. A premissa que orienta o alcance da simplicidade está no fato de se procurar atender essencialmente aquilo que é necessário acompanhado de controle corrente. No caso do XP, através de programação em par, busca-se prevenir possíveis erros. A execução dos trabalhos em duplas incentiva a troca de experiências e, conseqüentemente, maior aculturação dos participantes em relação ao projeto que desenvolvem (Rabello, 2006, p. 67; Teles, 2004, p. 48).

Para que seja empreendida a realização do projeto, a partir de novos paradigmas metodológicos e de conduta, quando comparado aos processos tradicionais, é necessário que os empreendedores XP demonstrem coragem na execução dos trabalhos, a partir dessa nova abordagem. Simplificar um código que está funcionando, fazer do usuário ou cliente um efetivo parceiro no processo de desenvolvimento, buscar sempre otimizar a comunicação com o cliente e usuários, não omitir alguma especificidade, desenvolver o sistema voltado aos testes, privilegiar a troca efetiva de informações e conhecimento, e romper com as exigências burocráticas exige uma mudança significativa do indivíduo, exige coragem (Ambler, 2004, p. 38).

A condução dos projetos baseados em XP recomenda a presença efetiva do cliente, o planejamento interativo, as reuniões breves e diárias, a programação em duplas, o desenvolvimento orientado para testes, a refatoração, o código coletivo e padronizado, a simplicidade do design, as metáforas, a integração contínua, as pequenas liberações e o ritmo constante. Essas práticas são mostradas na figura 3.

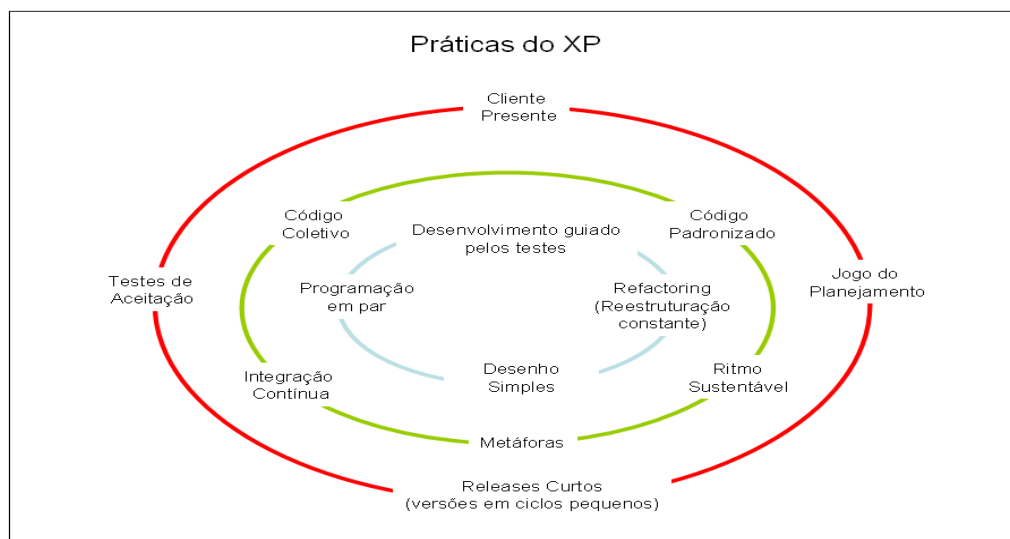


Figura 3: Práticas do XP
Fonte: www.xprogramming.com

1.2.2.2. SCRUM

O Scrum foi criado por Jeff Sutherland e Ken Schwaber, na década de 90 (Rabello, 2006, p.68); o termo foi inspirado no jogo de rugby.

A imprevisibilidade dos eventos constitui-se numa das premissas que o método Scrum considera e que o fundamenta. A imprevisibilidade endereça uma abordagem que trata adequadamente situações em que não se tem condição de obter requisitos definidos e estáveis. Uma abordagem de caráter empírico, como o Scrum, atende essa característica (Schwaber, 2004, p. 2).

O controle dos processos de desenvolvimento de software não são triviais; ao contrário. Por ser uma atividade de difícil controle, que não se equipara a uma fábrica de automóveis, por exemplo, exige uma abordagem diferenciada para a sua realização e que seja adaptável aos estímulos que ocorrem ao longo de todo o projeto. A adaptabilidade constitui-se num importante atributo dessa abordagem, para tratar adequadamente a ocorrência imprevisível desses estímulos. Em relação ao controle é recomendada a determinação do mínimo suficiente.

O Scrum orienta-se por três princípios: a visibilidade, a inspeção e a adaptabilidade (Schwaber, 2004, p. 3). As coisas devem estar visíveis a todos os envolvidos no desenvolvimento, a inspeção deve ser uma ação corrente e, conseqüentemente, as ações para adaptação do produto de software devem ser realizadas.

Vale salientar que essa abordagem é semelhante ao ciclo PDCA (Plan, Do, Check, Action), evidenciando que esse método fundamenta-se em valores já conhecidos e considerados válidos pela comunidade.

O ciclo PDCA é um método de gerenciamento de processos ou de sistemas, que assume uma forma cíclica, com a repetição de um conjunto de atividades, e objetiva melhorar o processo. (Campos, 2006).

São consideradas iniciativas essenciais do Scrum a comunicação, o trabalho em equipe, a flexibilidade, as entregas de produtos de software funcionando, sendo que essas entregas caracterizam-se por serem entregas que, a cada versão, possuem novas funcionalidades ou alguma melhoria que tenha sido introduzida pela equipe (Pressman, 2006, p. 69).

A equipe Scrum é composta pelo dono do projeto (product owner), pelo time (Team Scrum) e pelo líder (Scrum Master). Cabe ao dono do projeto estabelecer o que se pretende com o projeto e determinar a prioridade, o Product Backlog.

O Product Backlog constitui-se numa lista dos requisitos funcionais e não-funcionais com a atribuição do grau de importância, ou seja, essa lista de requisitos é avaliada, e é atribuída a precedência com a qual cada item deve ser realizado, determinando dessa maneira a precedência e a prioridade.

Ao time cabe a execução do que foi definido pelo dono do projeto e ao Scrum Master fazer com que o projeto flua, removendo obstáculos ou dificuldades, administrando as expectativas e provendo meios, para que o time possa realizar o projeto sem impedimentos. Cabe também ao Scrum Master fazer com que o processo de desenvolvimento esteja aderente ou em conformidade com o que o método Scrum estabelece. O Scrum Master é o conciliador dos que exercem os dois outros papéis e tem a incumbência de fazer valer a segregação de funções imposta por esses papéis e ajudar a resolver os conflitos, de maneira a garantir, por exemplo, que não ocorram sucessivas mudanças de prioridade pelo dono do projeto. Busca, também, controlar o excesso de euforia e otimismo que podem surgir durante a realização do projeto.

O time, responsável pela execução do projeto, deve ser composto por especialistas, e a responsabilidade é compartilhada por todos. O time deve ser composto por pessoas que detenham as competências e habilidades necessárias para a execução do projeto.

O processo Scrum é caracterizado pela realização de iterações, os sprints. A cada sprint é determinado um objetivo pelo dono do projeto. Para que seja alcançado o objetivo estabelecido, o time identifica as tarefas necessárias. Cada sprint deve ter uma duração determinada pela equipe e decorre de algumas variáveis, como a competência e a habilidade dos integrantes, a estabilidade da prioridade definida pelo dono do projeto e alguma outra variável que seja importante para a determinação desse tempo.

A duração de cada sprint, entre duas a quatro semanas, deve ser constante na realização do projeto. Essa constância na realização dos sprints determina o ritmo, a cadência do projeto, atributo que contribui para a agilidade esperada. Se o time é capaz, a cada sprint, de realizar as entregas de produtos de software

funcionando e, a cada novo sprint, realizar e entregar os incrementos solicitados, demonstra que a equipe está conseguindo ser eficaz e eficiente.

Outra característica importante dessa abordagem refere-se à capacidade do time se auto-organizar, planejar e, buscar o consenso para a solução de problemas e determinação de prioridade.

Durante as iterações, são realizadas reuniões curtas diárias (Daily Scrum Meeting), para que cada integrante da equipe Scrum apresente o que foi realizado e, se há alguma dificuldade, para o Scrum Master removê-la, se for o caso. É uma reunião que serve para que todos tenham oportunidade de constatar o progresso dos trabalhos.

O que determina o sucesso do sprint é a entrega. Sendo um software, deve estar funcionando e em conformidade ao que foi solicitado pelo dono do projeto.

Uma característica importante e que contribui para a estabilidade dos requisitos (product backlog) consiste na não introdução de mudanças durante essa iteração. Ao final de cada sprint é realizada uma reunião, para que aconteça uma revisão do que foi realizado (Sprint Review Meeting).

A figura 4 mostra a visão geral do processo SCRUM.

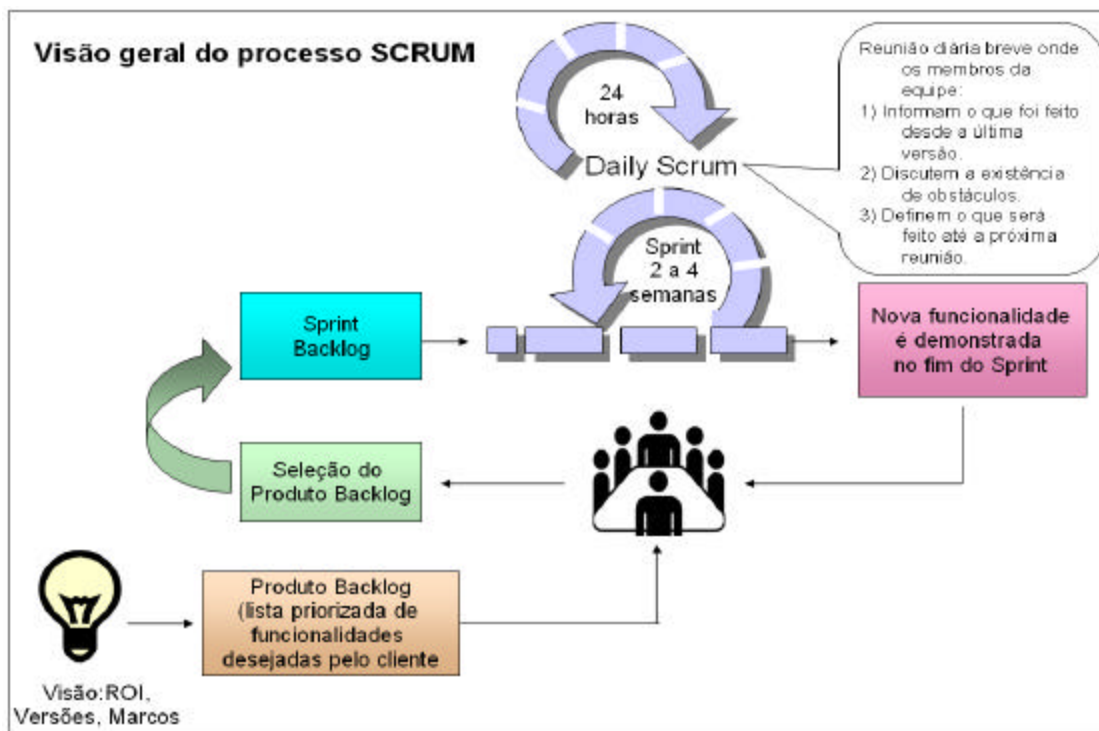


Figura 4: Visão Geral do processo SCRUM

Fonte: Schawaber, 2004, p. 9

1.3. Comparativo: abordagem ágil x abordagem tradicional

Destacam-se, neste capítulo, as características importantes da abordagem ágil, em contraponto à abordagem tradicional, uma vez que o ágil surgiu num contexto de busca de alternativas para a redução do caráter crônico do processo de software que é apresentado no capítulo 2.

As abordagens tradicional e ágil distinguem-se pela maneira como orientam o desenvolvimento de software. A abordagem tradicional privilegia fortemente os processos e a sua documentação, enquanto as abordagens ágeis privilegiam a adoção de poucas regras e práticas, e são fortemente orientadas para pessoas (Fowler, 2005).

O quadro 1 resume as diferenças importantes entre as abordagens.

QUADRO 1: Comparativo : abordagem ágil x abordagem tradicional

Categoria	Abordagem tradicional (Cascata)	Abordagem Ágil (XP/Scrum)
Especificação de Requisitos	Tem o caráter preditivo e prescritivo e, em consequência disso, exige que seja obtida uma especificação de requisitos completa, integral.	Tem o caráter adaptativo e, em consequência disso, os requisitos podem ser introduzidos durante o processo de desenvolvimento.
Aderência ao Processo	A equipe deve realizar as atividades em conformidade com o que é pré-estabelecido, o que a caracteriza pela rigidez do processo.	A equipe é incentivada a buscar, sempre, as melhores alternativas, para que seja alcançado o objetivo traçado para determinada iteração, o que a caracteriza pela flexibilidade do processo.
Posicionamento do controle	É uma abordagem orientada ao pós-controle. Após o final das fases ou etapas, são realizadas as verificações.	Privilegia-se o controle corrente, aquele que é realizado durante o processo.
Formalidade	Essencialmente burocrática em função das exigências impostas, que obrigam a produção de muita documentação, o que a torna produtora, também, de documentos.	Privilegia-se o mínimo de documentação, apenas o necessário.

QUADRO 1: Comparativo : abordagem ágil x abordagem tradicional (Cont.)

Categoria	Abordagem tradicional (Cascata)	Abordagem Ágil (XP/Scrum)
Orientação	Orientada para processos. A aderência ao processo deve garantir a qualidade dos produtos de software e a produtividade desejada.	Orientada para pessoas. O comprometimento, motivação e habilidades das pessoas são garantidoras da qualidade e produtividade desejadas.
Participação dos stakeholders	O envolvimento acontece em momentos pontuais e pré-determinados, normalmente descrevendo o problema.	Busca-se o comprometimento efetivo dos envolvidos, tornando-os integrantes da equipe de desenvolvimento, contribuindo para a solução do problema.
Perfil dos desenvolvedores	Privilegiam-se desenvolvedores especialistas que sejam fiéis às especificações.	Privilegiam-se desenvolvedores colaborativos e adaptados a entregas ágeis.
Custo das Mudanças	Quanto mais tarde ocorrer a necessidade de mudanças que exijam retrabalho ou modificações que impliquem numa reestruturação da arquitetura será mais custosa e, em consequência disso, em alguns casos impeditiva. Fato que justifica que as mudanças sejam evitadas.	A abordagem é orientada para mudanças, tornando-as menos custosas, fato que facilita a mudança.
Planejamento	Os produtos de software são obtidos em decorrência do planejamento inicialmente realizado. É orientado para a atividade.	Os produtos de software obtidos e entregues contribuem para o planejamento. É orientado para a entrega e mudanças.
Foco	O foco está no processo definido, rigoroso e formal.	O foco está nos valores, nas atitudes e nos princípios.
Indicadores de progresso	Obtidos a partir das medições das atividades realizadas.	Obtidos a partir das funcionalidades entregues
Gestão	Esse papel é exercido por um gerente de projeto.	Esse papel é exercido pela equipe.

QUADRO 1: Comparativo : abordagem ágil x abordagem tradicional (Cont.)

Categoria	Abordagem tradicional (Cascata)	Abordagem Ágil (XP/Scrum)
Codificação	Executado por um programador. Orientado para novas funcionalidades e para a realização de manutenções corretivas.	Executado por um ou dois desenvolvedores (XP). Orientado para a implementação de melhorias contínuas.
Projeto	Realizado por um analista projetista.	Realizado pela equipe.
Testes	Manuais. Orientados para encontrar defeitos.	Automatizados. Orientados para evitar defeitos.

1.4. Conclusão

A engenharia de software surgiu com o intuito de solucionar problemas que, pela sua magnitude, foram denominados de crise de software. Independentemente de ser uma crise ou problema crônico, o fato é que as dificuldades persistem (Pressman, 2006), e o desenvolvimento de software constitui-se numa tarefa muitas vezes árdua e que tem levado a muitos ao insucesso. Esses fatos justificam o grande interesse de todos os stakeholders, particularmente a comunidade de desenvolvedores, em buscar abordagens metodológicas que possam ajudar a eliminar ou minimizar significativamente as dificuldades enfrentadas a cada projeto. Essa problemática é apresentada no capítulo 2.

Sommerville (2003) relata que as características de determinada abordagem metodológica constituem-se de grande valor para a solução de problemas em determinados contextos. Dessa forma, tanto a abordagem tradicional quanto a ágil, destacadas neste capítulo, têm a sua validade.

Assim, características como formalidade, ênfase na documentação, cumprimento rigoroso das fases estabelecidas e produtos definidos, características da abordagem tradicional, não concorrem com a abordagem ágil, cujo conjunto de valores e princípios serve para resolver problemas cujas especificidades exigem uma ou outra abordagem.

Deve ser levado em consideração o fato das abordagens ágeis serem fortemente centradas nas pessoas, pois, como destaca Cockburn (2001), fatores

como talento, habilidades, conhecimentos, experiências, amizade e comunicação podem ser aspectos-chave na execução dos projetos. Por outro lado, o fato de se caracterizarem pela adaptabilidade, e focadas em resultados, as torna alternativas que devem ser consideradas na escolha de uma abordagem.

2. A problemática do desenvolvimento de software: uma retrospectiva

Há muito se tem dito que o desenvolvimento de sistemas passa por uma crise denominada crise do software (Pressman, 2006). Classificar um problema tão duradouro como sendo uma crise parece ser inadequado. De fato ocorre um problema crônico sem que se tenha uma solução definitiva para as suas causas. Esse cenário é evidenciado a seguir sob diversas perspectivas, justificando o interesse de pesquisadores no estudo e pesquisa do tema desenvolvimento de software.

O software, parte essencial de um sistema informatizado ou automatizado, determina as ações de execução e controle dos dispositivos computadorizados, independentemente do tipo, ou seja, software básico, de apoio e de aplicação, e é construído a partir de um processo de desenvolvimento. Esse processo de desenvolvimento e o seu produto – o software – têm sido o grande desafio ao longo dos anos, desde o surgimento dos recursos computacionais. Desenvolver software, atendendo as diversas perspectivas, no prazo e em conformidade com o orçamento e as funcionalidades necessárias, tem sido o grande desafio dos desenvolvedores de software.

Diversos fatos evidenciam que o desenvolvimento de software constitui-se, como já citado, numa empreitada de sucesso duvidoso.

O Standish Group, uma organização de pesquisa e análise independente da performance de projetos de Tecnologia da Informação, analisa diversos projetos nos Estados Unidos e Canadá. Seu primeiro relatório analisou 365 organizações de desenvolvimento, abrangendo 3682 sistemas ou projetos. Os dados publicados no primeiro relatório mostram que 31% dos projetos de software nunca são completados e 53% custam quase 200% da sua estimativa inicial. Calcula-se que as empresas norte-americanas e as agências governamentais gastaram 81 bilhões de dólares em 1995 com projetos de software cancelados (Larman, 2000, p. 416).

A figura 5 mostra a evolução da performance dos projetos de TI no período de 1994 a 2004, segundo o Standish Group. Notam-se melhoras significativas na taxa de sucesso; no entanto, a taxa de projetos que são finalizados ultrapassando o orçamento e prazos, além de disponibilizarem menos funcionalidades, continuam altos.

Teles (2004) ressalta os avanços apresentados pelo Standish Group no relatório de 2002. A taxa de sucesso saltou para 34%, ou seja, um aumento de 100% em 8 anos. Os projetos fracassados foram reduzidos para 15%, enquanto aqueles finalizados, ultrapassando o orçamento e prazos, além de disponibilizarem menos funcionalidades, se mantiveram praticamente estáveis na casa dos 51 %.

Jim Johnson, fundador e presidente do Standish Group, cita que a melhora no desempenho geral dos projetos foi devida a três fatores: melhor gerenciamento de projetos, aplicação de desenvolvimento iterativo e melhor infra-estrutura da WEB.

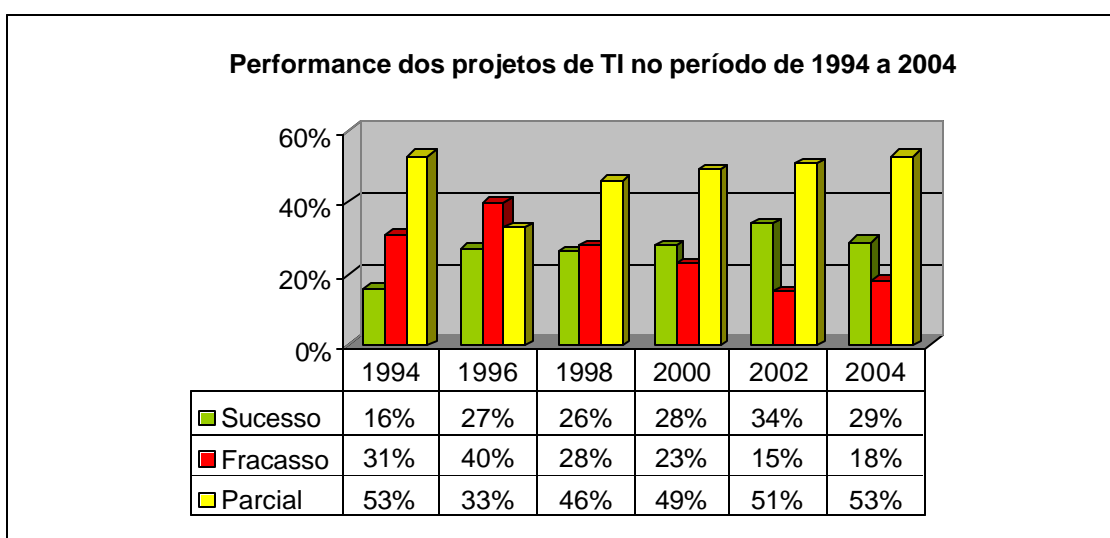


FIGURA 5: Performance dos projetos em TI no período de 1994 a 2004.

Fonte: CHAOS Database. <http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS-2004>.

O processo de desenvolvimento de software, apoiado pelas disciplinas que compõem a engenharia de software, tem amadurecido ao longo do tempo, mas os problemas persistem, não sendo capaz de atender plenamente quem adquire o produto de software e aqueles que o usam.

Esse cenário de entregas de produtos de software que não refletem a necessidade do usuário em termos de funcionalidades, qualidade, prazo e custos, é frequentemente observado, apesar de toda a evolução tecnológica que promove ambientes mais produtivos, processos mais bem definidos, profissionais mais bem-formados e a disponibilidade e utilização de ferramentas e equipamentos que facilitam significativamente o trabalho dos desenvolvedores. Em 1996, Braga (1996) já constatava essa situação.

Assim, a problemática do desenvolvimento de software resulta dos stakeholders, dos métodos empregados, da cultura organizacional e suas estratégias, dos recursos financeiros, materiais e tecnológicos disponíveis, entre outras perspectivas.

A problemática já inicia na fase de análise. Gane (1984) já considerava essa problemática e relacionava alguns aspectos que tornam a atividade de análise de sistemas tão difícil e exigente. São eles: a combinação das dificuldades técnicas do trabalho, as questões políticas que muitas vezes estão envolvidas e os problemas de comunicação que surgem, quando estão envolvidas pessoas com experiências diferentes e com diferentes pontos de vista.

Parece que aspectos relacionados a eficácia, eficiência e segurança – entre outros atributos – fazem parte do grande desafio dos desenvolvedores de software. Essa preocupação não é recente; Melendez Filho (1990, p. 36) lembra que não é raro vermos as seguintes situações que ocorrem após a implantação de sistemas:

- os usuários não utilizam todas as funções do sistema;
- não “abrem mão” de seus arquivos manuais;
- não confiam nos resultados eletrônicos do computador;
- boicotam dados para inviabilizar ou estremecer a credibilidade do sistema.

Da mesma forma, a preocupação com a qualidade do software sempre esteve presente, mas muitas vezes não se traduz em realidade.

Sobre as dificuldades de construir um sistema de informações, Shlaer (1990, p. 1) questiona:

Por que é tão difícil construir um sistema de processamento de informação de grande porte: conhecer exatamente o que estamos fazendo, enquanto progredimos através de vários passos do desenvolvimento do software, e conseguir no final (em tempo e dentro do orçamento, naturalmente) um sistema que satisfaça os requisitos e as expectativas do usuário e que seja fácil de se manter, modificar e compreender?

O quadro 2 mostra citações de outros autores sobre a problemática da engenharia de software, o que enfatiza o seu caráter crônico.

QUADRO 2: Citações sobre a problemática de Engenharia de Software

Autor	Citação
Silva (2007)	Ressalta que muitos projetos de software ainda fracassam.
Ambler (2004, p. 21)	Considera que a situação do desenvolvimento de software se encontra aquém do ideal.
Martin (1994, p. 376)	Lembra que o software é vendido, não quando está livre de erros, mas quando estes ocorrem com uma frequência razoavelmente baixa.
Martin (1991a, p. 6)	Cita que a construção de sistemas toma muito tempo, e o custo é muito alto.
McMenamin (1991, p. 502)	Alerta que o maior tempo é perdido na execução de atividades que não ajudam a atingir os objetivos do projeto.
Keller (1990, p. 5)	Relata que terminar um sistema atrasado, elaborar sistemas que não fazem o que o usuário espera, ou superar a previsão orçamentária são algumas das falhas típicas do desenvolvimento de sistemas da forma tradicional.
Chorafas (1988, p. 217)	Faz crítica aos profissionais da área de desenvolvimento, que classificava como sendo “pessoal de computadores”, que após mais de 30 anos não conseguiam comunicar seu conhecimento aos usuários finais de forma não-técnica.
Staa (1987, p. 5)	Menciona que é comum reclamar-se da baixa qualidade do software e de seu elevado custo de desenvolvimento.
Bingham (1977, p. 151)	Cita que muitos sistemas tornam-se obsoletos no momento em que entram em operação. Associado à questão da obsolescência prematura, observa-se que a insatisfação com os produtos de software está, frequentemente, presente.

2.1. A problemática do desenvolvimento de software sob a perspectiva das abordagens metodológicas.

É corriqueiro obter manifestações, relatos e citações de queixas decorrentes do uso de métodos de desenvolvimento de software.

As dificuldades, que se traduzem em erros, vêm sendo observadas ao longo dos anos e, até hoje, não houve uma solução adequada, e acontecem ao longo do ciclo de vida do software.

De fato, muitos problemas surgem nas primeiras fases do processo de desenvolvimento e, mesmo utilizando as melhores ferramentas, ainda persistem. Por exemplo, não existe ferramenta analítica que possibilite ao analista saber as necessidades ditas implícitas, ou seja, o usuário pensa, mas não diz.

Observa-se que a adoção de uma determinada metodologia, focada em uma determinada abordagem – orientada a processos, dados ou objetos – mostra-se problemática.

O fato evidente é que a parte que é atingida por esses problemas são os usuários e, diante dos sucessivos problemas, faz-se necessário um repensar de como os sistemas de informações são projetados e construídos.

O quadro 3 mostra a preocupação dos diversos autores ao longo dos anos.

QUADRO 3: Citações sob a perspectiva das abordagens metodológicas

Autor	Citação
Sommerville(2003)	Enfatiza que não existe método ideal e que diferentes métodos têm aplicação diversificada.
Molinari (2003, p. 36)	Ressalta que independente do processo de desenvolvimento de sistemas utilizado (seja moderno ou tradicional), se percebe que o usuário final sai sempre perdendo no que tange a não ter em mãos produtos com uma quantidade de defeitos aceitável.
Rocha (2001, p. 2),	Enfatiza a importância dos métodos usados no desenvolvimento de software.
Furlan (1998, p. 3)	Observa que tudo nos leva a crer que o problema está deixando de ser o software e o hardware – cada dia mais poderosos do que no dia anterior – para se tornar o método de trabalho que se emprega no uso dessa tecnologia, bem como seu alvo de utilização.
Coleman (1996, p.2).	Independente da abordagem metodológica empregada, verifica-se a baixa capacidade de previsão, duplicação de esforços, programas de baixa qualidade e altos custos de manutenção.
Keller (1990, p. 6)	Enfatiza que o problema com o método tradicional de desenvolvimento reside no fato de que as especificações são geralmente incompletas, ambíguas e redundantes e, portanto, difíceis de ser mudadas.
Fisher (1990, p. 21)	Cita que talvez piores que as especificações de projeto incompletas sejam as especificações imprecisas.
Rocha (1990, p.1)	Mesmo realizando refinamentos sucessivos, como nas abordagens incremental ou espiral, a produção de artefatos dificilmente se dá com perfeição.

QUADRO 3: Citações sob a perspectiva das abordagens metodológicas (Cont.)

Autor	Citação
Carvalho (1988, p. 177).	Declara que é comum constatar que muitos desenvolvedores se guiam pelas suas próprias inclinações e utilizam recursos que consideram adequados, independentemente dos demais.
Weiss (1988, p. 106)	Diz que a parte mais difícil do desenvolvimento de um sistema é o esforço de projeto, em seus primeiros estágios. É justamente nos estágios iniciais que se busca obter as especificações do projeto e é, nesse ponto, que se verifica que os documentos de especificação produzidos são incompletos.
Amaral Filho (1988, p. 22)	Alguns analistas tinham conhecimento de métodos de desenvolvimento de sistemas, obtidos de cursos preparatórios por empresas fornecedoras de equipamentos. Essas metodologias eram muito fracas, com documentos em excesso e de pouca valia para o projeto em si. Eram basicamente documentacionais e extremamente carregadas, o que dificultava ainda mais o já complexo projeto do sistema.
Teixeira (1979, p. 20)	Afirma que a maioria dos erros têm origem conceitual e são cometidos nas atividades de Análise e Projeto, dentro da etapa de Desenvolvimento e, nesse contexto, verifica que durante o desenvolvimento a produtividade é baixa, porque uma grande parte dos esforços são dedicados à correção dos erros introduzidos no processo.

2.2. A problemática do desenvolvimento de software sob a perspectiva dos stakeholders

O fato é que o desenvolvimento de *software* é difícil e complexo, para que um único indivíduo possua todo o conhecimento e as habilidades necessárias para desenvolver e distribuir com sucesso um sistema. As pessoas precisam trabalhar juntas, para que seus pontos fortes formem um todo balanceado que permita vencer o grande desafio, que é o desenvolvimento de sistemas (Pressman, 2006).

A participação dos interessados no desenvolvimento de software constitui-se numa máxima relatada com frequência, e muitos autores já demonstravam essa preocupação como mostrado no quadro 4.

QUADRO 4: Citações sob a perspectiva dos stakeholders

Autor	Citação
Pressman (2006, p.3)	Cita que o programador solitário de antigamente foi substituído por uma equipe de especialistas em software.
Pressman (2006, p. 16)	Considera que a engenharia de software é realizada por pessoas criativas, com amplos conhecimentos, que devem adaptar um processo de software maduro, apropriado para os produtos que elas constroem e para as demandas do seu mercado.
Rocha (2001, p. 2)	Destaca que desenvolver software é um esforço coletivo (envolve muitas pessoas), complexo e criativo, e que a qualidade do software depende das pessoas.
Larman (2000, p. 41)	Destaca que mais importante que seguir um processo ou método oficial é um desenvolvedor adquirir a habilidade de criar um bom projeto, e que organizações sustentem o desenvolvimento desse tipo de habilidade. Ela surge do domínio de um conjunto de princípios e de heurísticas relacionadas com a identificação e abstração de objetos adequados e da atribuição de responsabilidades aos mesmos.
Martin (1994, p. 4)	Observa que uma das preocupações mais urgentes da indústria da informática, hoje, é a necessidade de se criarem softwares e sistemas corporativos muito mais velozes a um custo mais baixo. Martin (1994, p. 416) diz que uma grande parte do problema é que os humanos são lentos.
Maffeo (1992, p. 34).	Diante das exigências cada dia mais sofisticadas e céleres, constata-se que o perfil dos recursos humanos adequados para o desenvolvimento de software vem sofrendo pressão de adaptação para essas condições.
Martin (1991a, p. 96),	A problemática que é evidenciada, quando se consideram os stakeholders no contexto do desenvolvimento de sistemas, está associada às dificuldades de se comunicarem claramente as idéias uns aos outros; e, além disso, na medida em que o desenvolvimento progride, a compreensão de seus clientes sobre o domínio do problema muda, suas prioridades mudam, e sua visão da solução também muda, conforme eles acompanham o desenvolvimento do sistema.
Keller (1990, p. 91).	O papel do usuário não se resume a mero receptor dos artefatos para uso, é co-autor e também orientador do projeto.
Melendez filho (1990, p. 53).	A participação dos stakeholders, em especial dos usuários, torna-se cada vez mais um fator básico para a construção de sistemas estáveis, de fácil operacionalização e de bom desempenho funcional

QUADRO 4: Citações sob a perspectiva dos stakeholders (Cont.)

Autor	Citação
<p>Donaldo S. Dias, apud Melendez Filho (1990, p. 37)</p>	<p>Aponta quatro argumentos para a participação do usuário na especificação de sistemas:</p> <p>“As pessoas têm o direito moral de controlar seus próprios destinos, e isto se aplica às situações de trabalho”;</p> <p>As pessoas são efetivamente as responsáveis finais pela execução e controle das atividades da empresa. “As pessoas não consultadas sobre suas atividades podem decidir anular decisões tomadas por outros, tão logo estes saiam de cena”;</p> <p>“O usuário final é, em última análise, um especialista na sua área de trabalho, no que se refere a aspectos operacionais e também informacionais”;</p> <p>“A participação do usuário serve como um fato gerador de motivação do projeto, obtendo-se disto uma maior produtividade e eficiência”.</p>
<p>DeMarco (1989, p.6)</p>	<p>Apesar de parecer óbvio, constata-se que a comunicação entre os stakeholders muitas vezes não acontece ou, quando acontece, não é suficiente, fato que leva ao insucesso do projeto de desenvolvimento de sistemas. DeMarco já considerava que a experiência dos usuários na área de trabalho deve ser um ingrediente-chave para o desenvolvimento dos sistemas.</p>
<p>Chorafas (1988, p. 118)</p>	<p>É fato que, se os colaboradores envolvidos diretamente na especificação do problema tiverem plena compreensão do domínio do problema, haverá menos espaço para a interpretação errônea daquilo que for objeto da análise. Chorafas ressalta que os administradores e profissionais falam muito mais livremente sobre o que fazem, caso a discussão ocorra com alguém que compreende o que dizem.</p>
<p>Staa (1987, p.6)</p>	<p>A participação efetiva dos stakeholders, constituída de profissionais talentosos, é considerada tão relevante, a tal ponto de ser possível obter-se software de boa qualidade, mesmo trabalhando de forma pouco sistemática, embora não seja usual.</p>
<p>Bingham (1977, p. 152)</p>	<p>“É essencial para o sucesso da análise e do projeto de sistemas que o pessoal que dele se utiliza seja envolvido desde o princípio em seu projeto”.</p>

2.3. A problemática do desenvolvimento de software sob a perspectiva do tempo (celeridade)

Explicita-se a dimensão tempo com o intuito de reforçar a importância do atendimento célere exigido pelas organizações (Boehm, 2006).

Essa celeridade constitui-se num dos atributos da qualidade que é imposto aos produtos ou artefatos de software. É necessário ter agilidade em todas as fases do ciclo de vida do software, para que seja possível oferecer alguma vantagem sob a perspectiva das estratégias de negócio.

As citações do quadro 5 colaboram para reforçar esse tema.

QUADRO 5: Citações sob a perspectiva do tempo

Autor	Citação
Pressman (2006)	Afirma que ainda temos problemas na construção de software de alta qualidade, no prazo e dentro do orçamento.
Begossi (2002)	Confirma a existência de sérios problemas quanto ao tempo, custo e qualidade de desenvolvimento de muitos produtos de software. Enfatiza ainda que é quase norma que esses problemas aconteçam nos projetos de software.
Bartié (2002, p. 4)	Constata que estamos percebendo que, de forma rápida e constante, as organizações estão aumentando sua dependência tecnológica, e isso significa que suas operações internas estão sendo conduzidas e direcionadas por um conjunto cada vez maior de sistemas informatizados. Estão sofisticando seus sistemas, para tomar decisões cada vez mais complexas, com a intenção de ganhar eficiência e controle.
Taylor (1995, p. 3)	Afirma que, para sobreviver em um ambiente de negócios cada vez mais competitivo, as empresas devem estar dispostas a se reinventarem constantemente. Elas só podem fazer isto, se seus sistemas de software forem capazes de rápidas mudanças. Entretanto, o que se observa são – com certa frequência – fatos que não contribuem para a obtenção célere de produtos de software.
Martin (1995, p. 75)	Alerta que as corporações de movimento lento estão superadas
Martin (1994, p. 416)	Diz que uma grande parte do problema é que os humanos são lentos.

QUADRO 5: Citações sob a perspectiva do tempo (Cont.)

Autor Martin (1991a, p. 6)	Citação Declara que a construção de sistemas toma muito tempo, e o custo é muito alto.
McMenamin (1991, p. 502)	Ressalta que se gasta tempo de projeto para conversar com usuários, para rever a documentação existente, para desenhar, para conduzir revisões, para redesenhar e, especialmente, para criar e manter definições de dicionário de dados e mini-especificações.

A figura 6 mostra os atrasos no período de 1994 a 2004, nos Estados Unidos e Canadá. Os dados evidenciam uma melhora importante, mas o percentual continua elevado.

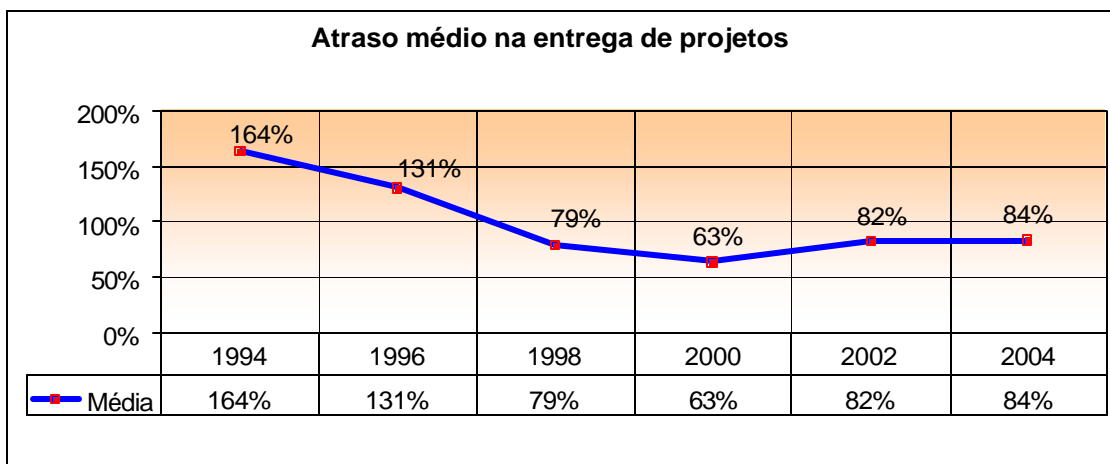


FIGURA 6: Atraso médio na entrega de projetos

Fonte: CHAO S Database. <http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS-2004>.

2.4. A problemática do desenvolvimento de software sob a perspectiva do custo

As questões associadas ao custo do software referem-se às dificuldades de estimá-lo para o seu desenvolvimento. A dificuldade de estimar os custos contribui para alguns fracassos, uma vez que contribuem para a frustração dos usuários e clientes.

O quadro 6 mostra a preocupação dos autores sobre essa perspectiva, e a figura 7 mostra que, no período de 1994 a 2004, houve uma melhora nas

estimativas. Segundo Jim Johnson, isso se deve a melhores práticas de gerenciamento.

QUADRO 6: Citações sob a perspectiva do custo

Autor	Citação
Presman (2006, p.4)	Salienta que os custos do software são concentrados na engenharia e acontecem em todo o processo de software. Independentemente do modelo de processo de software adotado, o objetivo é a obtenção de software a um custo mais baixo possível, acompanhado da qualidade exigida e nos prazos estabelecidos.
Mendes (2002, p. 3)	Diz que a idéia central da engenharia de software é fazer uso de princípios de engenharia, a fim de produzir, a baixo custo, softwares que operem corretamente e com eficiência, em equipamentos onde sejam instalados.
Martin (1991a, p. 6)	Declara que a construção de sistemas toma muito tempo, e o custo é muito alto.

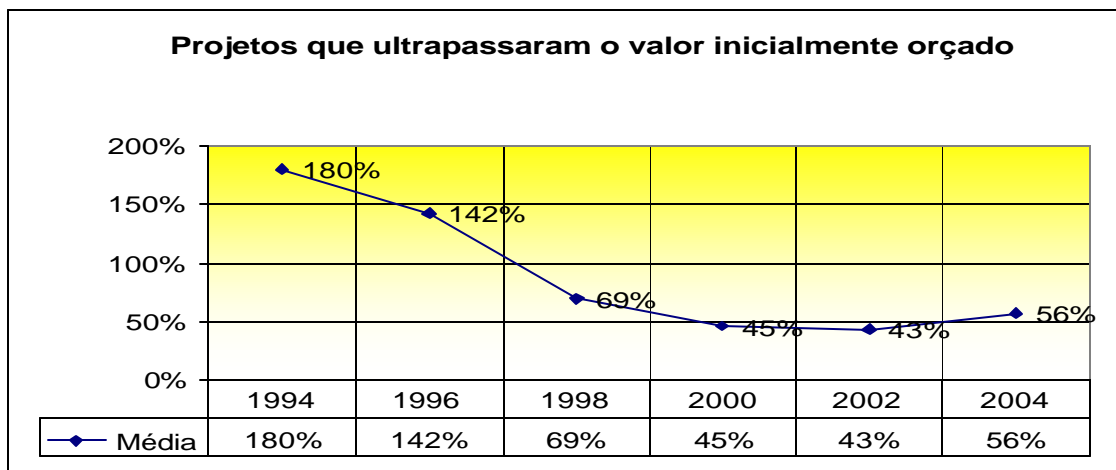


FIGURA 7: Projetos que ultrapassaram o valor inicialmente orçado.

Fonte: CHAOS Database. <http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS-2004>.

2.5. A problemática do desenvolvimento de software sob a perspectiva das ferramentas

A adequação e suficiência das ferramentas utilizadas durante o processo de desenvolvimento de software contribuem para o alcance do sucesso na realização de projetos.

Outro aspecto relevante está relacionado à eficiência na execução das atividades executadas manualmente, comparada à utilização, por exemplo, de softwares geradores de código. Quanto à eficácia, não se pode atribuir às ferramentas responsabilidade pelos casos de insucesso.

O quadro 7 apresenta o pensamento dos autores sobre o assunto.

QUADRO 7: Citações sob a perspectiva das ferramentas

Autor	Citação
Norbert Wiener, apud Martin (1994, p. 415)	Diz que tarefas que são desumanas, porque são enfadonhas, deveriam ser feitas por máquinas, e não por pessoas. Martin complementa dizendo que o trabalho do programador é desumano, porque os programadores têm que escrever grandes quantidades de código complexo sem erro.
Martin (1994, p. 255)	Recomenda evitar a programação manual, onde possível, para construir sistemas tão rapidamente quanto possível.
McMenamin (1991, p. 502)	Os desenvolvedores que se utilizam de ferramentas erradas podem perder tempo, independentemente de estarem fazendo as coisas certas, e podem impor dificuldades decorrentes da inadequação das ferramentas utilizadas.

2.6. Conclusão

Enfatizaram-se as dificuldades que, ao longo do tempo, têm acompanhado o processo de desenvolvimento de sistemas, sob as perspectivas das abordagens metodológicas, stakeholders, tempo, custos e ferramentas, evidenciando o seu caráter crônico, fato que incentiva a busca de abordagens que minimizem ou eliminem os problemas relatados.

É nesse cenário, onde ainda se verifica o caráter crônico do processo de desenvolvimento, que se insere o meio acadêmico que tem por objetivo oferecer à comunidade profissionais habilitados e capacitados a solucionar problemas e minimizem ou eliminem os fatos citados. Para tal, deve buscar caminhos seguros que possam contribuir nesse sentido. Com esse intuito é apresentada, no capítulo seguinte, a abordagem atualmente utilizada no ensino do processo de software, na FATEC/SP, e, no capítulo 4, o estudo de caso que procurou observar os impactos dessa nova abordagem no processo de aprendizagem.

3. O ambiente de estudo de engenharia de software na FATEC-SP

Este capítulo apresenta as disciplinas de análise e projeto de sistemas, no âmbito do curso de tecnologia em processamento de dados da FATEC-SP, a abordagem metodológica que orienta a condução dessas disciplinas, e apresenta as estratégias, em sala de aula, e outras atividades associadas ao processo de ensino e aprendizagem de engenharia de software.

O processo de ensino e aprendizagem se dá no decorrer de três semestres, no curso de graduação em tecnologia de processamento de dados. Tem a finalidade de formar graduados aptos a desenvolver a função de analista de sistemas e ser capazes de desenvolver software de qualidade, atendendo os requisitos estabelecidos como prazos, custos e funcionalidades, entre outros. A capacitação pretendida visa contribuir para minimizar a problemática do desenvolvimento de software, apresentado no capítulo 2, que, como foi descrito, constitui-se num problema crônico que acompanha os empreendimentos de desenvolvimento de sistemas.

Nesse contexto, o interesse do ensino, no âmbito da engenharia de software, é, em suma, contribuir para que os egressos do curso estejam adequadamente preparados para o desenvolvimento de software de qualidade.

O ensino das disciplinas de Análise e Projeto de Sistemas está fundamentado no modelo de ciclo de vida clássico, cujas fases são estudadas ao longo de três semestres. Essa abordagem privilegia a condução das atividades a partir de fases bem definidas, caracterizadas pelo rigor formal e pela produção de artefatos definidos em cada uma das fases dos projetos de desenvolvimento de sistemas (apêndice A).

Além da preocupação com a fundamentação teórica, exige-se dos alunos o desenvolvimento das habilidades, ou seja, incentiva-se o “saber fazer”, que se constitui num importante orientador da conduta docente e discente, em sala de aula e nas atividades desenvolvidas em laboratório.

As estratégias adotadas visam obter condições de aprendizagem que permitam aos alunos experimentar situações problema que possam permitir-lhes aprender a aprender, pois seria utópico oferecer experimentações caracterizadas pela completude dos desafios, espelhando, de fato, o mundo real (Begosso, 2002).

3.1. A adequação do modelo clássico sob a perspectiva da aprendizagem

A adoção do modelo de ciclo de vida clássico permite aos aprendizes ter claramente definido o que deve ser obtido fase a fase. Essa abordagem considera a estabilidade do ambiente do sistema que será informatizado, além de exigir o conhecimento prévio dos seus requisitos.

Ao aprendiz cabe atender os objetivos estabelecidos quanto aos prazos, custos, funcionalidades e outros requisitos explicitados ou não, no desenvolvimento de sistemas baseado nessa abordagem, que é introduzido a partir de uma metodologia de desenvolvimento e manutenção de sistemas (MDMS). A produção da maioria dos artefatos, antes da fase de programação, constitui-se essencialmente de documentos que são gerados ou coletados ao longo do projeto. Daí a percepção da ênfase na produção de documentos que é exigida por essa abordagem metodológica.

3.2. As estratégias em sala de aula, atividades em laboratório e extra-classe

A condução das aulas e atividades em laboratório buscam oferecer os fundamentos teóricos necessários ao aprendizado do processo de software, e são complementadas por atividades práticas que exigem dos alunos o desenvolvimento de um sistema de informações, ao longo das disciplinas de APS (APS I, APS II e APS III).

As iniciativas em sala de aula visam, a partir de aulas expositivas, a apresentação dos tópicos e respectivos conteúdos, conforme os planos de ensino. São apresentados aos alunos de maneira que possam aprender o processo de software e tenham condições de iniciar as atividades práticas, com o intuito de incentivar o trabalho em grupos, aproximá-los às práticas de negócios, mesmo que parcialmente, e corroborar para a construção das suas experiências.

As aulas expositivas privilegiam os conceitos que suportam o processo de desenvolvimento de sistemas, e as atividades práticas reforçam esse paradigma. Essa estratégia visa contribuir para a fixação, pelos aprendizes, dos fundamentos e

passos que são exigidos, no momento da execução de atividades de desenvolvimento de sistemas.

Além dos tópicos abordados a partir do que é exigido nos respectivos planos de ensino, são utilizadas, complementarmente, as normas NBR/ISO IEC 12207 e NBR/ISO IEC 9126, que contribuem para a melhor percepção do processo de software e dos atributos da qualidade pelos aprendizes.

A adoção da norma NBR ISO/IEC 12207 orienta o desenvolvedor na estruturação e gerenciamento do ciclo de desenvolvimento, e a norma NBR ISO/IEC 9126 descreve um modelo de qualidade do produto de software a partir da definição dos atributos da qualidade.

As aulas são conduzidas numa seqüência que coincide com as atividades exigidas pela MDMS, o que permite solicitar as produções em seguida à apresentação dos tópicos.

Essa MDMS orienta a execução das fases, conforme podemos verificar na figura 8.

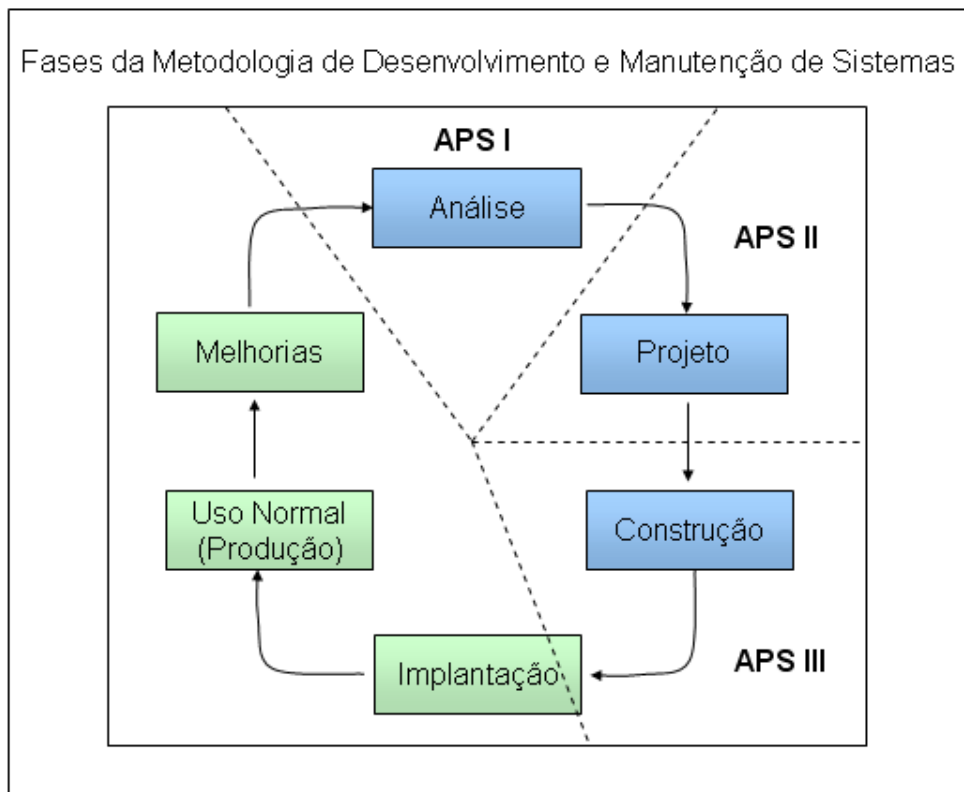


FIGURA 8: Fases da MDMS

Fonte: Adaptação do livro Best practice for Application Management. Inglaterra: 2002, OGC.

Na fase de Análise, onde são levantadas a situação atual e as necessidades das áreas, a partir de questionários, entrevistas, reuniões, verificações in loco, entre outras técnicas, pretende-se que os aprendizes tenham sido capazes de obter, a partir do levantamento de dados realizado, as necessidades e formalizar as especificações de requisitos.

A representação da realidade se dá de cima para baixo, de forma “top down”, característica marcante da análise estruturada. A decomposição dos sistemas se dá a partir dos níveis de abstração, mostrados na figura 9.

O levantamento de dados, a partir desse critério de decomposição, é realizado e documentado através do Diagrama de Fluxo de Dados (DFD) e do dicionário de dados, que completa essa documentação.

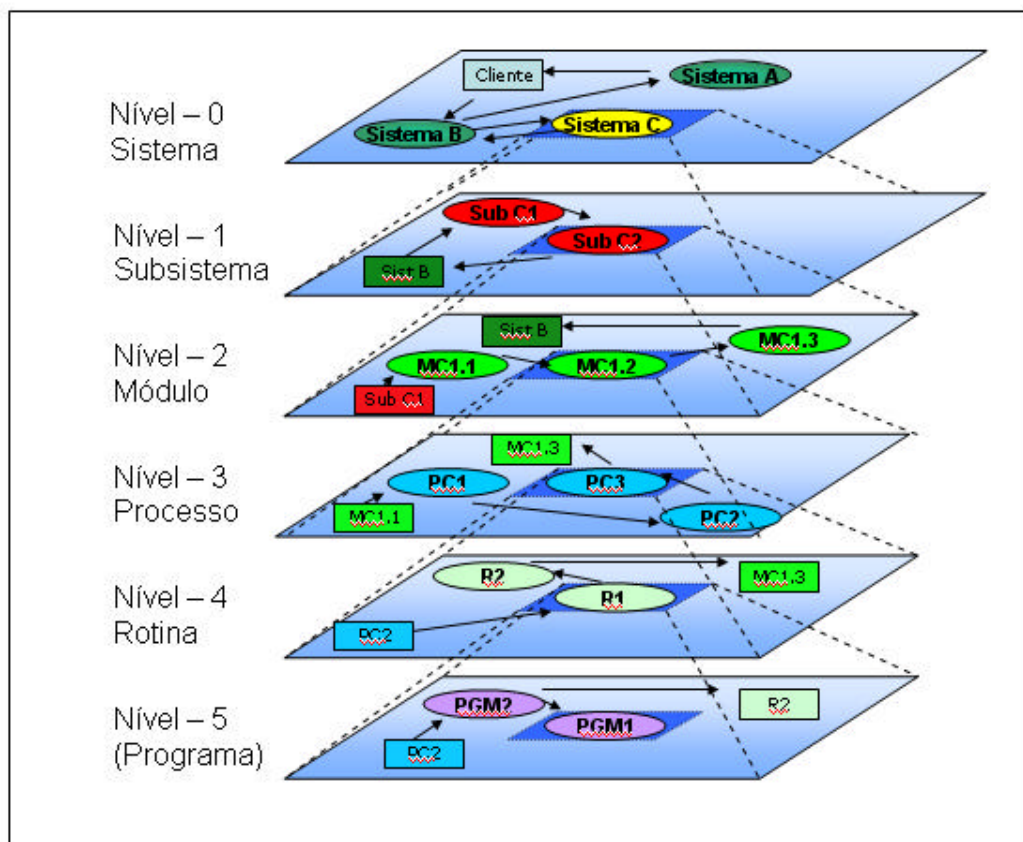


FIGURA 9: Níveis de Abstração

Fonte: Baseada no conceito de construção em níveis proposto por Tom DeMarco.

A partir da análise do que foi obtido no levantamento da situação atual, é produzida a proposta do que se pretende informatizar. Dessa proposta são elaborados os grafos estruturais que alicerçam a fase de projeto do sistema.

Na fase de projeto, são contemplados e documentados as interfaces, as estruturas de dados, as especificações de programas, o fluxo das rotinas e os demais níveis.

Na fase de construção, os programas especificados são codificados. Nessa fase, ocorrem, também, os testes unitários, de integração e de sistemas. É estabelecida como meta a execução do sistema com zero erro.

Recomenda-se, aos alunos, utilizarem ferramentas de software distribuídas gratuitamente para a elaboração dos diagramas de fluxo de dados, uma vez que essas ferramentas facilitam a decomposição em “n” níveis de abstração e em conformidade com a notação de Tom DeMarco (DeMarco, 1989; Chiossi, 2006).

Para as atividades práticas, os alunos seguem um roteiro de desenvolvimento que orienta os alunos na execução das atividades (apêndice A).

3.2.1. Ferramentas geradoras de código

As disciplinas de APS adotam o software gerador de aplicações Genexus como ferramenta de apoio ao desenvolvimento de aplicações, sendo utilizado para prototipação e desenvolvimento incremental, com o objetivo de agilizar o processo de desenvolvimento de software, a partir dos benefícios oferecidos por esse software, que contempla várias atividades do ciclo de vida de desenvolvimento.

Pressman (2002) ressalta que a Engenharia de Software é difícil, e que as ferramentas que reduzem a quantidade de esforço necessário, para produzir um trabalho ou atingir algum marco de projeto, têm benefícios substanciais.

Sommerville (2003) também manifesta a importância das ferramentas de construção de sistemas que automatizam o processo de construção, a fim de reduzir a possibilidade de erros humanos e, quando possível, minimizar o tempo necessário para a construção de um sistema.

3.2.1.1. Genexus

Genexus é uma ferramenta multi-plataforma de desenvolvimento (www.genexus.com). Gera 100% da aplicação, o código fonte, baseado nos

requisitos do usuário, não exige programadores para a codificação do programa e se encarrega da manutenção do banco de dados. É orientado para a prototipação e desenvolvimento incremental, o que permite agilizar o processo de desenvolvimento. Trabalha com o conceito de base de conhecimento (knowledge base, KB); nessa KB são armazenadas todas as informações do projeto; e possui 3 (três) ambientes, denominados de modelos: design (Design Model), prototipação (Prototype Model) e produção (Production Model).

O Design Model é baseado na descrição dos objetos, e as estruturas de dados (tabelas físicas) são projetadas, de acordo com a teoria de banco de dados relacional. Essas tabelas são geradas pelo GeneXus, que se encarrega de proceder a normalização até a terceira forma normal (3FN).

O Prototype Model permite que a aplicação seja totalmente gerada e testada, antes de ser disponibilizada para implantação no ambiente de produção. Isso permite a qualquer envolvido no projeto testar o modelo implementado, as interfaces de comunicação, as funcionalidades primitivas, as tabelas e as regras de negócios.

No ambiente de prototipação é permitido:

- administrar o banco de dados, e
- executar a aplicação para avaliação.

O Production Model permite gerar a aplicação final, aprovada pelo usuário no ambiente de protótipo, e liberá-la para o ambiente de produção. As funções desse ambiente são:

- administrar o banco de dados associado ao Modelo de Produção;
- gerar programas que serão executados no ambiente de Produção;
- executar a aplicação para processar dados reais.

Os ambientes de produção e protótipo são similares em termos de funcionalidade, diferenciando-se apenas no seu propósito, ou seja, o ambiente de prototipação e o de design servem para o desenvolvimento e testes de software, enquanto o ambiente de produção serve para a distribuição e implantação do sistema gerado.

3.3. A abordagem tradicional e o processo de ensino/aprendizagem

A abordagem tradicional, pelo seu caráter prescritivo, formal e ênfase no

controle, permite que professores e alunos se orientem facilmente no processo de desenvolvimento, pois todos sabem quais artefatos devem ter sido produzidos até o final de cada fase.

Essa abordagem, ao definir claramente as fases, etapas, atividades e produtos que devem ser gerados, serve ao aluno como referencial importante, permitindo-lhe, antecipadamente, saber tudo que deve produzir.

Acontece que a carga horária das disciplinas, de 72 horas por semestre, num total de 3 (três), deve ser suficiente para empreender a realização de um projeto, concorrendo com as outras demandas da disciplina, no cumprimento do que estabelecem os respectivos planos de ensino.

Busca-se adequar a carga horária definida, de maneira que o aluno possa ter condições de obter todo o arcabouço teórico exigido, acrescido das práticas que contribuem para a consolidação do conhecimento e o desenvolvimento das habilidades exigidas ao longo dos semestres.

3.4 Conclusão

Como foi destacado no capítulo 2, o processo de desenvolvimento de sistemas tem problemas crônicos que se verificam há muito tempo e até hoje não estão adequadamente resolvidos. Esses problemas são causados, entre outros motivos, pelo excesso de formalidade dos modelos de processo propostos nos últimos 30 anos (Fowler, 2005). Fowler destaca ainda que há a necessidade de desenvolver softwares, de forma mais rápida, não deixando de produzi-los com qualidade.

Apesar do caráter crônico do processo de desenvolvimento de software, cujo modelo é base para muitas das metodologias utilizadas, tem-se nessa abordagem o referencial para o processo de ensino/aprendizagem no curso.

Em uma realidade que tem privilegiado a celeridade, além da qualidade e de outros requisitos importantes, impõe-se aos formadores, no mínimo, o papel de levar às novas gerações de formandos as abordagens que possam contribuir para que o caráter crônico apresentado seja pelo menos minimizado, sendo que se busca – idealmente – a eliminação desse problema.

A oportunidade de estudo, pesquisa e adoção, nesse contexto, de outra

abordagem que possa impor mais agilidade é recomendável, desde que a qualidade no processo de ensino e aprendizagem da engenharia de software não seja comprometida.

A partir de 2001, quando foi publicado o “Manifesto Ágil”, ocorreu de forma mais rápida a popularização dos métodos ágeis, cujos princípios, que regem a adoção dessas abordagens, privilegiam o desenvolvimento centrado nas pessoas, no resultado sendo entregue e funcionando, na adesão do cliente ao time de desenvolvedores e em uma pré-disposição para atender as mudanças exigidas de forma rápida, não significando que seja rejeitado o que recomendam as metodologias que se enquadram na abordagem tradicional.

Esse cenário incentiva a avaliação do impacto dessas novas abordagens no processo de ensino e aprendizagem de engenharia de software, de maneira a concluir sobre a sua eficácia e contribuição para minimizar os problemas crônicos que foram relatados, cujos resultados são apresentados no capítulo 4.

4. Impacto do uso de métodos ágeis no processo de ensino e aprendizagem de engenharia de software

Este capítulo apresenta os estudos de caso aplicados aos alunos do curso de Tecnologia em Processamento de Dados da FATEC-SP, com o objetivo de verificar o impacto do uso de métodos ágeis no processo de ensino e aprendizagem de engenharia de software. Essa verificação contempla como os valores, princípios e práticas ágeis contribuem para o aprendizado do processo de software. As observações e o testemunho dos participantes constituem-se em fonte de informação para o objetivo do trabalho, que está orientado para a aprendizagem do processo de software.

4.1 Estudo de Caso

Os alunos do Curso de Processamento de Dados da FATEC-SP, dos turnos matutino e vespertino, nos anos 2006 e 2007, foram convidados a participar deste projeto de pesquisa. As variáveis observadas consideraram a “maturidade acadêmica” de cada aluno e, principalmente, a sua percepção dos elementos facilitadores e dificultadores focada nos aspectos relacionados ao processo de software estudado e aplicado no desenvolvimento dos sistemas em laboratório.

Os recursos de infra-estrutura que influem na aprendizagem da engenharia de software, considerados sob a ótica do aluno, são neutralizados pela adoção do mesmo ambiente para todos os participantes desta pesquisa.

As aplicações dos métodos, tradicional e ágil, foram observadas “in loco” e através de relatórios de progresso elaborados pelos alunos e professores, onde os requisitos funcionais subsidiaram a obtenção de indicadores relacionados aos métodos aplicados.

Os indicadores contemplam o processo e o produto do desenvolvimento de sistemas e evidenciam o quanto impacta a adoção de métodos ágeis, na aprendizagem da engenharia de software, a partir das funcionalidades implementadas.

Os projetos descritos neste capítulo, os estudos de caso, foram executados por times compostos com experiência acadêmica na abordagem tradicional, uma vez

que estavam cursando a disciplina de Análise e Projetos de Sistemas III e sem experiência anterior em métodos ágeis. Para tanto, foi dada uma visão geral dessa nova abordagem, enfatizando os aspectos relevantes para o propósito do trabalho.

A figura 10 mostra como os grupos são identificados nos projetos desenvolvidos que são descritos no tópico 4.3.

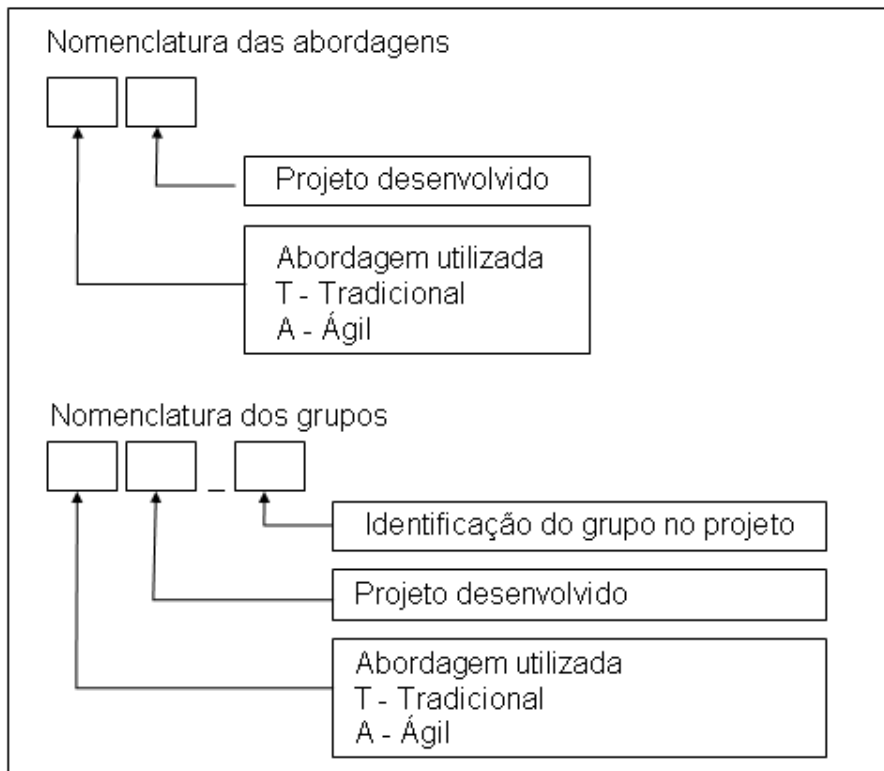


FIGURA 10: Nomenclatura adotada nos estudos de caso

4.2. Coleta de Dados

A coleta de dados foi realizada utilizando-se questionários, relatórios de acompanhamento e verificações “in loco”. Os questionários a seguir foram utilizados:

- Questionário de Identificação do perfil do grupo (Apêndice B): Foi utilizado um questionário para coletar informações sobre o nível educacional e de experiência profissional dos alunos, o conhecimento da abordagem tradicional e de métodos ágeis. O questionário foi

aplicado aos alunos do primeiro semestre de 2007 e do segundo semestre de 2007.

- Questionário de avaliação I (Apêndice C): O questionário de avaliação I foi utilizado para coletar informações sobre a eficácia da aprendizagem, utilizando a abordagem tradicional e o método ágil. O questionário foi aplicado aos alunos do primeiro semestre de 2007.
- Questionário de avaliação II (Apêndice D): O questionário de avaliação II contém as questões do questionário I acrescidas de questões sobre gerência de projetos, como o auto-gerenciamento e reuniões rápidas propostas pelo SCRUM. O questionário foi aplicado aos alunos do segundo semestre de 2007.

4.3. Impacto das práticas propostas pelo XP (extreme programming)

O objetivo dos estudos de caso foi verificar o quanto as práticas propostas pelo XP (extreme programming) contribuem para uma implementação mais simples, eficaz e eficiente, e o quanto contribuem para o processo de aprendizado do processo de software.

4.3.1. Projeto 1 – Cenário: Planejamento da produção

Um abatedouro de frangos deseja informatizar um dos processos de sua área de planejamento e controle de produção. Esse planejamento é originado pela informação dos produtos que a área comercial estima vender no dia seguinte. O produto acabado é obtido a partir da decomposição do frango vivo. Esse módulo deverá permitir aos gestores obter a quantidade de frangos vivos necessários para a produção do que foi estimado.

4.3.1.1. Variáveis do estudo de caso









O tempo estipulado para o projeto foi de 15 horas, sendo 3 horas por semana, na presença da autora do texto, para que todos os alunos tivessem a mesma orientação e o mesmo tempo de programação. O desenvolvimento foi feito no laboratório da FATEC/SP, sendo que a autora atuou como usuária do sistema. Participaram deste estudo 16 alunos do curso de Tecnologia em Processamento de Dados da Fatec-SP, sendo que 4 (quatro) deles programaram individualmente, e os demais, em pares. Foram definidas 4 (quatro) funcionalidades para este projeto. A linguagem de programação foi escolhida pelos alunos, de acordo com o conhecimento adquirido em disciplinas de linguagem cursadas anteriormente.

4.3.1.2. Apresentação dos resultados

Foram definidas quatro funcionalidades que deveriam ser entregues ao final do prazo estabelecido, tanto para aqueles que adotaram a abordagem ágil, quanto para aqueles que adotaram a tradicional.

O quadro 8 mostra as funcionalidades implementadas no projeto.

QUADRO 8: Funcionalidades implementadas no projeto 1

Projeto 1											
Funcionalidades	A1 _A	T1 _A	A1 _B	T1 _B	A1 _C	T1 _C	T1 _D	A1 _D	A1 _E	A1 _F	
Cadastrar componentes	NA	AT	AT	AP	NA	AP	AT	AT	NA	NA	
Cadastrar estrutura do produto	NA	NA	AT	NA	NA	NA	AT	AT	AP	NA	
Cadastrar planejamento	NA	NA	AT	NA	AT	NA	AT	AT	AP	NA	
Calcular necessidades	NA	NA	AP	NA	AT	NA	AT	AT	NA	NA	
Legenda		Abordagem Ágil				Delphi					
		Abordagem Tradicional				JAVA					
		AT - Atendida				Visual Basic 6.0					
		AP - Atendida parcialmente									
		NA - Não atendida									

A figura 11 mostra as funcionalidades implementadas pelos alunos que desenvolveram com base na abordagem tradicional.

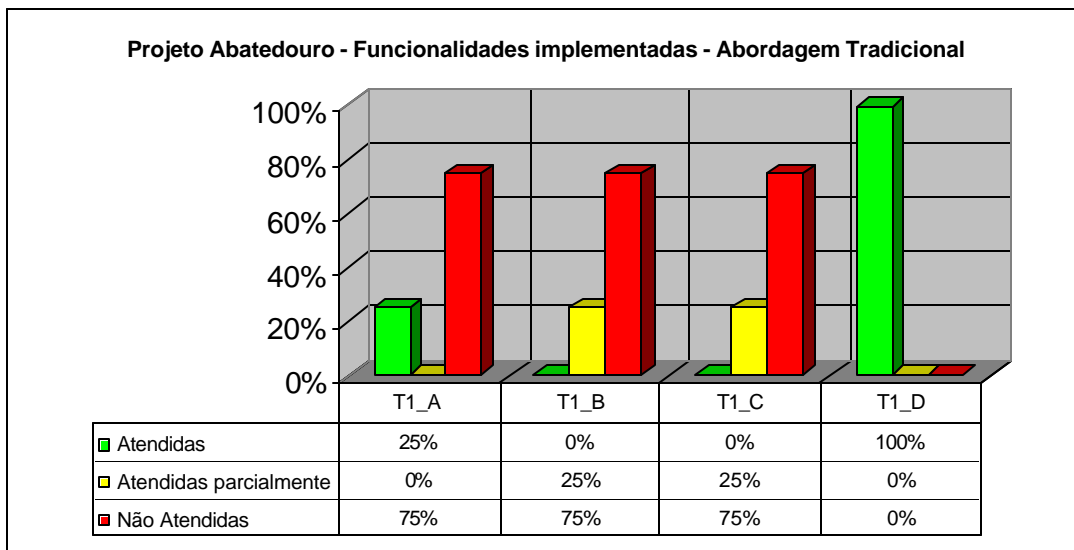


FIGURA 11: Projeto 1 - Funcionalidades implementadas (Tradicional)

A figura 12 mostra as funcionalidades implementadas pelos alunos que desenvolveram pela abordagem ágil.

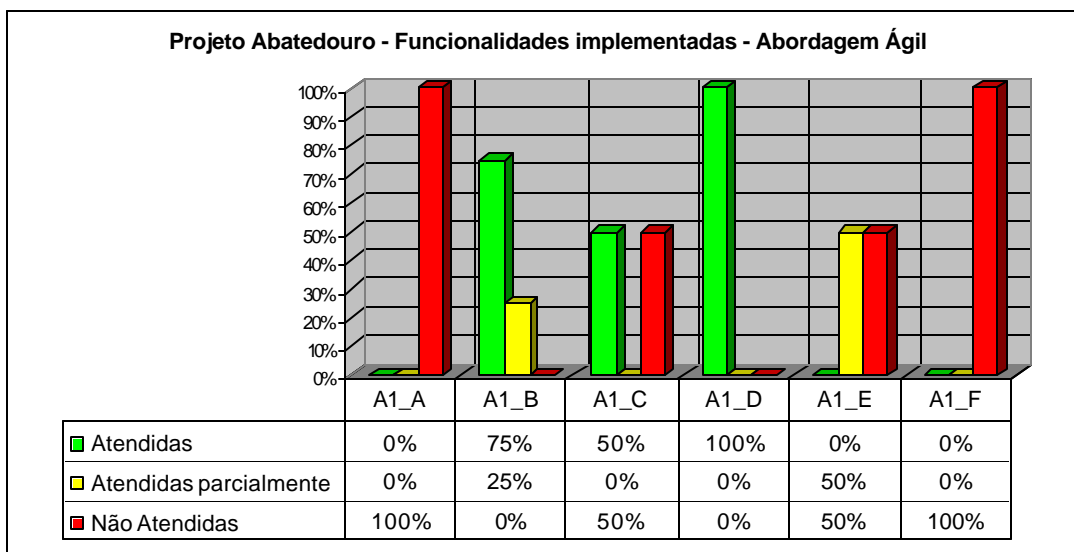


FIGURA 12: Projeto 1 - Funcionalidades implementadas (Ágil)

A figura 13 faz um comparativo entre as abordagens para o projeto 1.

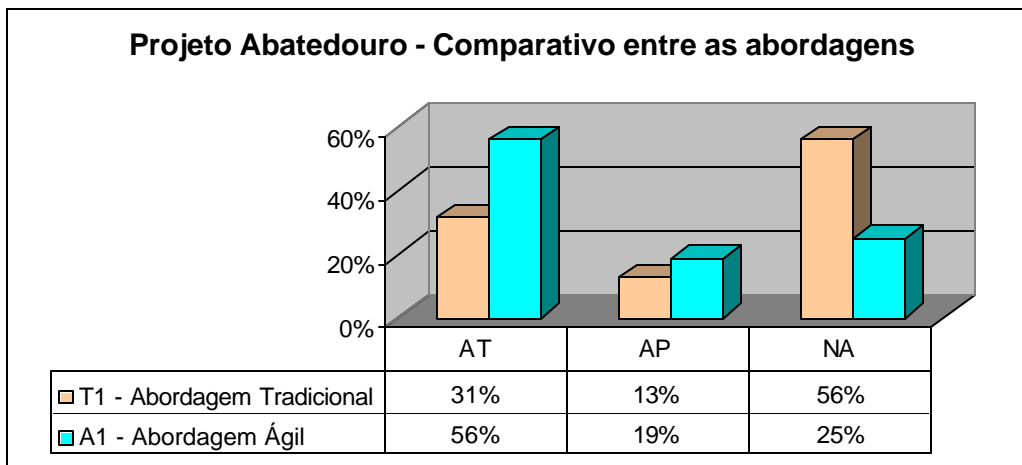


FIGURA 13: Projeto 1 - Comparativo entre as abordagens

4.3.1.3. Observações realizadas durante o estudo de caso

Os alunos que adotaram a abordagem ágil ressaltaram que um dos benefícios importantes foi a possibilidade de troca de conhecimentos, que muitas vezes se complementam, facilitando a execução do trabalho, e também contribuem para o aprendizado daqueles com menos conhecimento e experiência.

4.3.1.4. Análise dos resultados

Para a análise dos resultados deste projeto não foram considerados os dados dos grupos que não conseguiram realizar pelo menos uma implementação. Essa decisão foi tomada com o intuito de eliminar possíveis desvios que pudessem estar associados a problemas de aprendizado de outras disciplinas, como o domínio de linguagens de programação, que impedem o êxito das tarefas solicitadas.

Os dados obtidos demonstram que a produtividade alcançada pelo grupo A1, que utilizou a abordagem ágil (XP), foi significativamente superior, conforme a figura 12, superando em 79,9% as funcionalidades implementadas pelo grupo T1, que adotou a abordagem tradicional. Verifica-se, também, que a taxa de funcionalidades não implementadas pelo grupo A1 foi de 25% contra 56% do grupo T1.

Em relação às funcionalidades implementadas parcialmente, constata-se que o grupo A1 entregou 19% contra 13% do grupo T1.

Ao refinar a análise, considerando o desempenho dos times que pertencem ao mesmo grupo, sobressaem os seguintes fatos:

- um time que adotou a abordagem tradicional contribuiu com 4 das 5 funcionalidades entregues, o que representa 80% do total;
- um time que adotou a abordagem ágil contribuiu com 4 das 9 funcionalidades entregues, o que representa 44% do total.

Esses participantes podem ser considerados “premium people”, termo que Boehm (2002) utiliza para destacar aqueles colaboradores que se sobressaem em relação aos demais.

Conclui-se, a partir dos resultados obtidos, que:

- a abordagem ágil contribui para a obtenção de maior produtividade;
- tanto na abordagem ágil quanto na tradicional sobressaem as competências individuais.

4.3.2 Projeto 2 – Cenário: Formulação de Tintas

Uma indústria de tintas deseja informatizar o processo de formulação de tintas. Essa atividade consiste na formulação das tintas que devem atender às especificações fornecidas pelos clientes. O processo de formulação contempla duas atividades: a determinação da composição da tinta desenvolvida e os processos que devem ser executados para a sua fabricação.

4.3.2.1. Variáveis do estudo de caso

Foi estimado o total de 15 horas de trabalho para a execução desse estudo de caso, sendo 3 horas semanais na presença da autora do texto, para que todos os alunos tivessem a mesma orientação e o mesmo tempo de execução. As atividades foram realizadas no laboratório da FATEC/SP, sendo que a autora atuou como

usuária do sistema. Participaram deste estudo 15 alunos do curso de Tecnologia em Processamento de Dados da Fatec-SP, sendo que 7 deles programaram individualmente e os demais, em pares. Foram definidas 7 (sete) funcionalidades. A linguagem de programação foi escolhida pelos alunos, de acordo com o conhecimento adquirido em disciplinas de linguagem cursadas anteriormente.

4.3.2.2. Apresentação dos resultados

Foram definidas sete funcionalidades que deveriam ser entregues ao final do prazo estabelecido, tanto para aqueles que adotaram a abordagem ágil, quanto para aqueles que adotaram a abordagem tradicional.

O quadro 9 mostra as funcionalidades implementadas no projeto 2.

QUADRO 9 : Funcionalidades Implementadas no projeto 2

Projeto 2											
Funcionalidades	T2_ A	A2_ A	T2_ B	T2_ C	T2_ D	T2_ E	A2_ B	T2_ F	A2_ C	T2_ G	A2_ D
Cadastrar MP	AT	AT	NA	NA	NA	AP	NA	AT	AT	AP	AT
Cadastrar Produto	NA	AT	NA	NA	NA	NA	NA	AT	AT	AP	AT
Cadastrar Processos	NA	AT	NA	NA	NA	NA	NA	AT	AT	AP	AT
Cadastrar Roteiro de Fabricação	NA	NA	NA	NA	NA	NA	NA	AT	AT	NA	AP
Cadastrar Cliente	NA	NA	NA	NA	NA	AP	NA	AT	AT	NA	AT
Cadastrar Pedido	NA	AP	NA	NA	NA	NA	NA	AT	AT	AP	AP
Consultar Pedido	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	AP
Legenda		Abordagem Ágil					Delphi				
		Abordagem Tradicional					JAVA				
		AT - Atendida					Visual Basic 6.0				
		AP - Atendida Parcialmente									
		NA - Não atendida									

A figura 14 mostra as funcionalidades implementadas pelos alunos que desenvolveram pela abordagem tradicional.

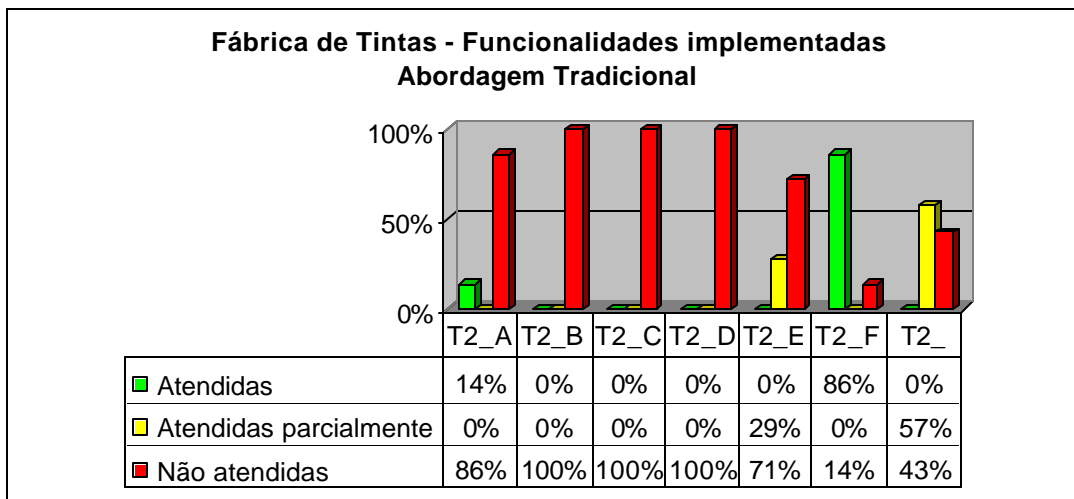


FIGURA 14: Projeto 2 - Funcionalidades implementadas (Tradicional)

A figura 15 mostra as funcionalidades implementadas pelos alunos que desenvolveram pela abordagem ágil.

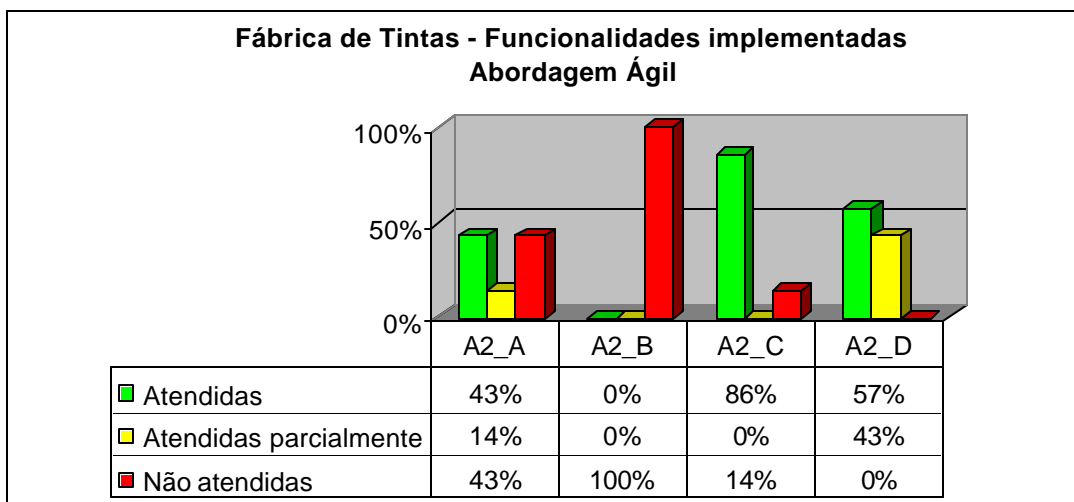


FIGURA 15: Projeto 2 - Funcionalidades implementadas (Ágil)

A figura 16 faz um comparativo entre as duas abordagens no projeto Fábrica de Tintas.

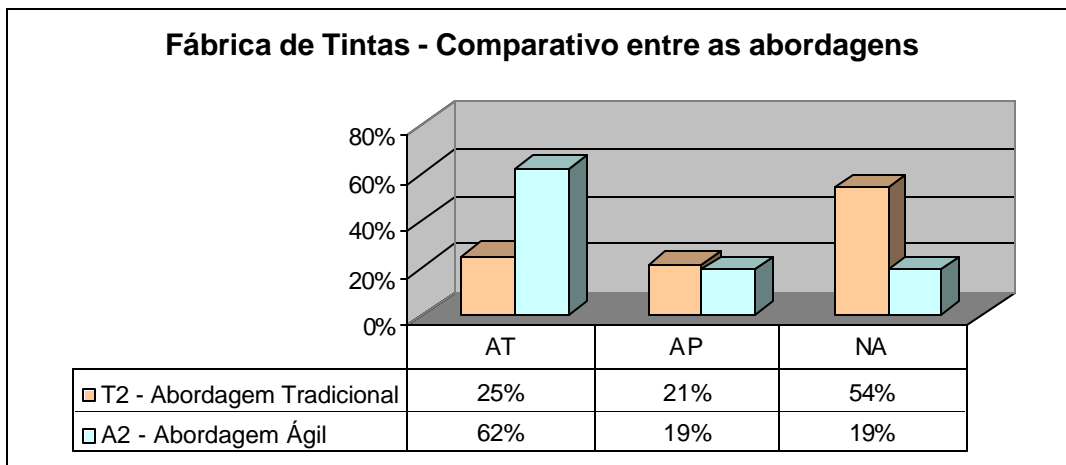


FIGURA 16: Projeto 2 – Comparativo entre as abordagens

4.3.2.3. Observações realizadas durante o estudo de caso

Os casos que apresentaram uma produtividade muito elevada, comparados à produção dos demais, foram alcançados por participantes que se destacavam do grupo pela motivação, determinação e responsabilidade, além de deterem conhecimento e habilidades adequadas à execução das tarefas.

4.3.2.4. Análise dos resultados

Os dados obtidos demonstram que a produtividade alcançada pelo grupo A2, que utilizou a abordagem ágil (XP), ultrapassou em 148% a produção do grupo T2, que adotou a tradicional.

Ao considerar o taxa de insucesso desses grupos, observa-se que o grupo A2 não conseguiu implementar 19% das funcionalidades, enquanto o grupo T2, 54%.

Sobressai a produção alcançada por um participante, do Grupo T2, que implementou 6 (seis) funcionalidades, contribuindo com 21,4%, o mesmo que um time do Grupo A2, que contribuiu com 28,6%.

Para a análise desses resultados, não foram considerados os dados dos grupos que não conseguiram realizar pelo menos uma implementação. Essa decisão foi tomada com o intuito de eliminar possíveis desvios que pudessem estar

associados a problemas de aprendizado de outras disciplinas, como o domínio de linguagens de programação, que impedem o êxito das tarefas solicitadas.

4.4. Impacto do uso de métodos ágeis associados ao uso de software gerador de aplicações

O objetivo deste estudo de caso foi verificar o quanto as práticas propostas pela abordagem ágil contribuem para o aprendizado e interferem na eficiência do processo de desenvolvimento, comparado com a abordagem tradicional, associado ao uso de ferramentas geradoras de aplicação.

4.4.1. Projeto 3 – Cenário: “Central Fast de Rádio Táxi”

A “Fast”, uma cooperativa de taxistas, possui uma central de atendimento que recebe todas as solicitações e aciona o cooperado, o taxista que está mais próximo da chamada. O cliente, no primeiro contato, informa o seu nome completo, telefone para contato, endereço onde se encontra, destino, número de pessoas e alguma outra informação que o cliente deseja manifestar, como urgência ou preferência. A central informa ao cliente, na efetivação da “corrida”, o valor a ser pago ao motorista. A “Fast” também realiza serviços de transporte de pequenos volumes. Da mesma forma, o cliente solicita a retirada e entrega de volumes, especificando hora e local de retirada, local para retirada e entrega e restrições que devem ser observadas na prestação do serviço. Na confirmação da solicitação de serviços, é informado o valor a ser cobrado ao cliente.

No final de cada mês, a empresa informa ao taxista o valor da taxa de serviços que deve ser pago à “Fast”. Esse valor corresponde a 3% (três por cento) da somatória dos valores referente a todos os serviços prestados no mês base.

4.4.1.1. Variáveis do estudo de caso

Os alunos tiveram cerca de 30 dias para realizar o projeto, e não foram impostas restrições quanto ao local de desenvolvimento. Participaram deste estudo de caso 16 alunos do curso de Processamento de Dados da Fatec-SP, do primeiro semestre de 2007, separados em 4 grupos, conforme especificado no quadro 10.

Para verificar se outros fatores, como experiência anterior, origem do ensino médio e outros fatores pudessem afetar os resultados, foi solicitado aos participantes responderem um questionário para identificação do perfil dos alunos (apêndice B). Dos 16 alunos, 15 responderam o questionário. Os dados coletados no questionário estão representados no apêndice E. Desse grupo, 27% aponta a área de Análise de Sistemas como sendo a área de interesse no curso, 40%, a área de programação e 33% indica outras áreas de interesse. Em relação ao tipo de ensino médio cursado, 66% dos alunos cursaram, exclusivamente, o ensino médio em escolas públicas, 20% exclusivamente em escolas particulares, 7% em escola pública federal e 7% estudaram a maior parte em escola particular. Cerca de 93% do grupo conhece modelo de ciclo de vida, e 87% indica que não tem conhecimento de métodos ágeis. Em relação à experiência em desenvolvimento de sistemas, 100% informam não tê-la, e 93% não tinha nenhuma experiência profissional. Os demais itens, coletados pelo questionário de identificação do grupo, não possuem fatores que possam distorcer os resultados da pesquisa.

Nesse estudo de caso, utilizamos um aluno do quinto semestre para representar o usuário e dono do produto. Ele ficou à disposição dos alunos, para tirar dúvidas dos requisitos do sistema.

QUADRO 10 - Distribuição dos grupos por abordagem

Distribuição dos grupos por abordagem		
Grupo	Abordagem	Linguagem
A3_A	Abordagem Ágil	Delphi
T3_A	Abordagem Tradicional	VB
A3_B	Abordagem Ágil	Genexus
T3_B	Abordagem Tradicional	Genexus

4.4.1.2. Apresentação dos resultados

Foram definidas treze funcionalidades que deveriam ser entregues ao final do prazo estabelecido, tanto para aqueles que adotaram a abordagem ágil, quanto para aqueles que adotaram a tradicional, com e sem a utilização de um software gerador de aplicação.

A tabela 1 mostra o tempo que os alunos levaram para desenvolver o projeto, e o quadro 11 mostra as funcionalidades implementadas pelos grupos.

TABELA 1: Horas utilizadas no desenvolvimento do projeto 3

Horas utilizadas no desenvolvimento do projeto Central Fast Radio Táxi					
Grupo	Abordagem	Linguagem	Total hrs do grupo	Nro de Participantes	Média de hrs Individual
A3_A	Ágil	Delphi	688 hrs	4	172
T3_A	Tradicional	VB	672 hrs	4	168
A3_B	Ágil	Genexus	504 hrs	4	126
T3_B	Tradicional	Genexus	544 hrs	4	136

QUADRO 11: Funcionalidades Implementadas no projeto 3

Central Fast Rádio Táxi				
Funcionalidades	A3_A	T3_A	A3_B	T3_B
Cadastro de Clientes	AT	NA	AT	AT
Cadastro de Ufe (Estados)	AT	NA	NA	NA
Tabela de Preço	AT	NA	NA	NA
Cadastro de Carro	AT	NA	AT	AT
Cadastro de Motorista	AT	NA	AT	AT
Cadastro de Setor	AT	NA	NA	NA
Cadastro de Rotas	AT	NA	NA	NA
Cadastro de Preço (Bandeiras)	AT	NA	NA	AT
Cadastro de Disponibilidade dos Táxis	AT	NA	NA	AT
Cadastro de Chamadas	AT	NA	AP	AT
Cadastro de Ocorrências	AT	NA	AT	AT
Cadastro de Faturamento	AT	NA	AP	NA
Cadastro de Agendamento	AT	NA	NA	NA
Legenda				
		Abordagem Ágil		
		Abordagem Tradicional		
		Abordagem Ágil com Ferramenta		
		Abordagem Tradicional com ferramenta		
		AT - Atendidas		
		AP - Atendidas parcialmente		
		NA - Não atendidas		

As figuras 17 a 21 traçam um comparativo entre as produções realizadas por abordagem.

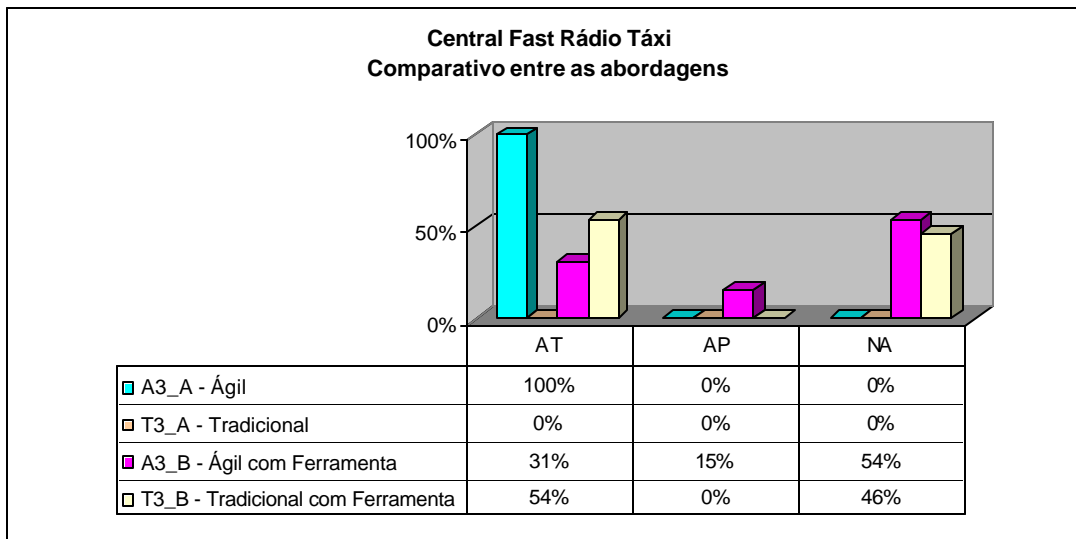


FIGURA 17: Projeto 3 – Comparativo entre as abordagens

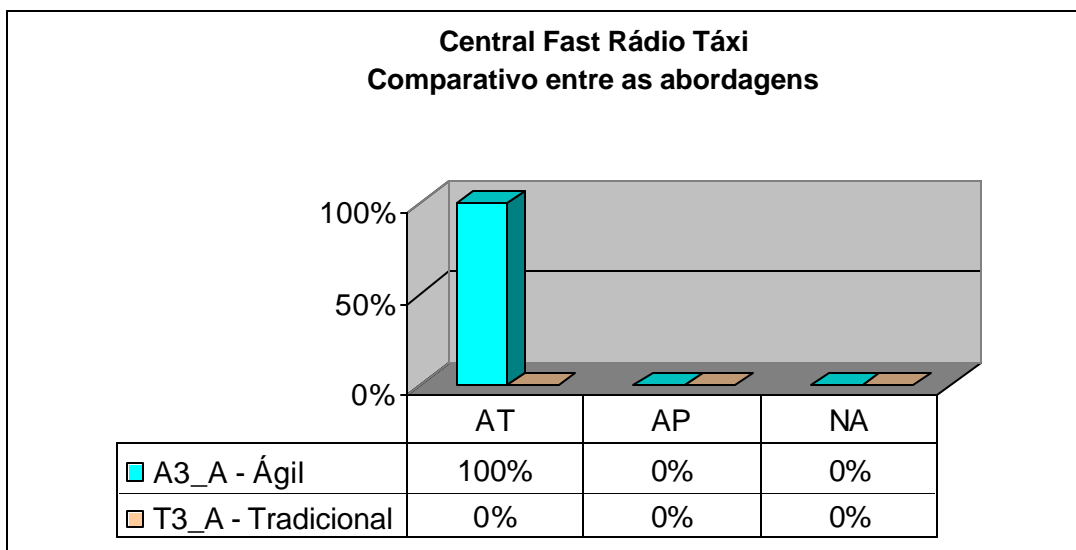


FIGURA 18: Projeto 3 - Funcionalidades implementadas sem o uso de ferramenta geradora.

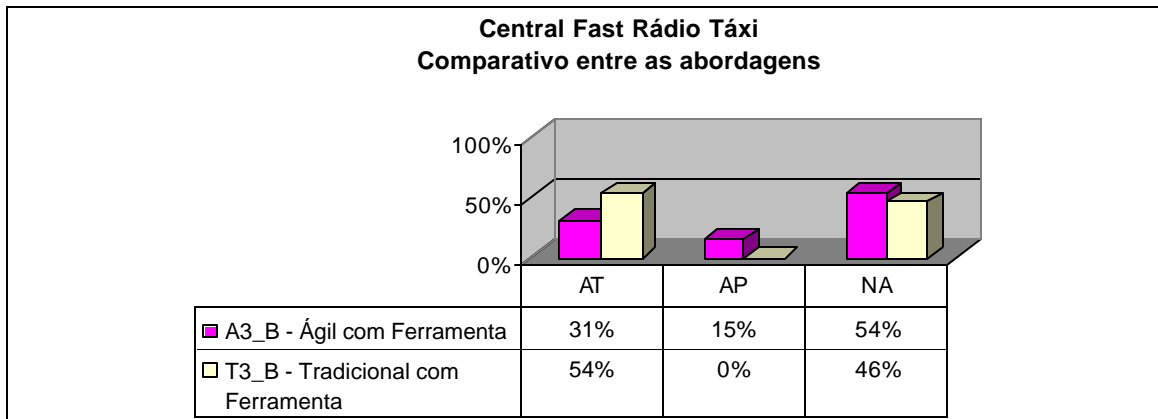


FIGURA 19: Projeto 3 - Funcionalidades implementadas com o uso de ferramentas

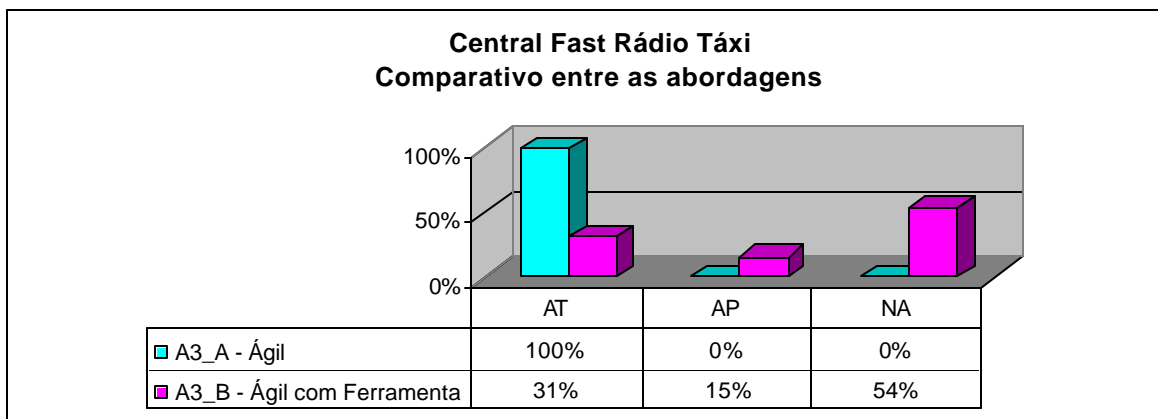


FIGURA 20: Projeto 3 - Funcionalidades implementadas utilizando a abordagem ágil

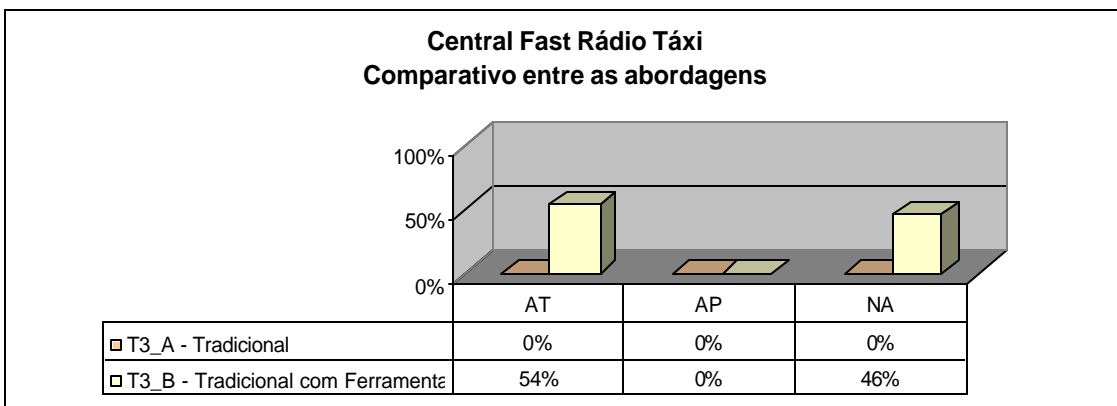


FIGURA 21: Projeto 3 - Funcionalidades implementadas utilizando a abordagem tradicional

4.4.1.3. Observações realizadas durante o estudo de caso

Os colaboradores do grupo A3_A podem ser classificados como “premium people”, fato que contribuiu significativamente para o desempenho diferenciado desse grupo.

O grupo T3_A seguiu o roteiro estabelecido para a abordagem tradicional, e privilegiou as atividades documentacionais, não tendo tempo de realizar a implementação.

O grupo A3_B, que adotou a abordagem ágil com software gerador de aplicação, não incorporou os valores e práticas dessa abordagem. Além disso, observou-se que o perfil dos integrantes desse grupo não contribuía para o uso adequado do software gerador, o que explica o desempenho inferior apresentado nas figuras 19 e 20.

Os participantes relataram que o pré-conhecimento da abordagem tradicional foi um facilitador para a execução das atividades baseadas na abordagem ágil. Esse pré-conhecimento constituiu-se num referencial que permitiu aos participantes terem uma visão geral do processo e daquilo que podiam otimizar.

4.4.1.4. Análise dos resultados

Os dados obtidos demonstram que a produtividade alcançada pelo grupo que adotou a abordagem ágil (XP), sem o uso de software gerador de aplicação, fez a entrega de 100% das funcionalidades solicitadas. Os que adotaram a abordagem tradicional, sem o uso desse software gerador de aplicação, não conseguiram implementar qualquer funcionalidade, cabendo ressaltar que esse grupo executou o projeto até a fase de especificação de interfaces, arquivos e tabelas. Ambos os grupos consumiram praticamente a mesma quantidade de horas, conforme tabela 1.

A adoção de software gerador de aplicação não contribuiu significativamente para a melhora do desempenho, sob a perspectiva da taxa de insucesso, conforme figura 16.

Conclui-se, a partir dos resultados obtidos, que:

- a abordagem ágil contribui para a obtenção de maior produtividade;
- a adoção de software gerador de aplicação não propiciou um desempenho significativamente maior, no contexto deste projeto;

4.4.1.5. Avaliação das abordagens realizada pelos participantes

Ao término do estudo de caso, os participantes foram convidados a opinar sobre questões relacionadas a:

- adequação das abordagens para uso em sala de aula, visando a aprendizagem;
- assimilação do processo de software a partir da abordagem adotada;
- facilidades oferecidas pela abordagem na execução das tarefas;
- eficiência da abordagem adotada na produção de artefatos.

Os participantes consideraram que ambas as abordagens podem ser utilizadas em sala de aula, conforme figura 22. Quanto à facilidade para a assimilação do processo de software, 75% entende que a abordagem tradicional é melhor (figura 23). Os participantes consideraram que, por ser a abordagem ágil mais leve, torna-se mais adequada à execução, conforme figura 24. Sob a perspectiva da eficiência, foi unânime a opinião de que a abordagem ágil é mais adequada (figura 25).

Outras considerações sobre as abordagens estão no apêndice H.

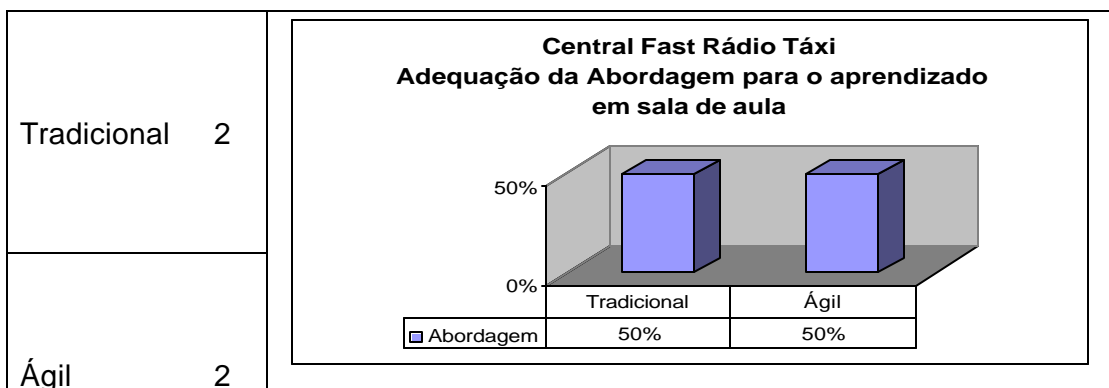


FIGURA 22: Projeto 3 - Adequação da abordagem para o aprendizado em sala de aula

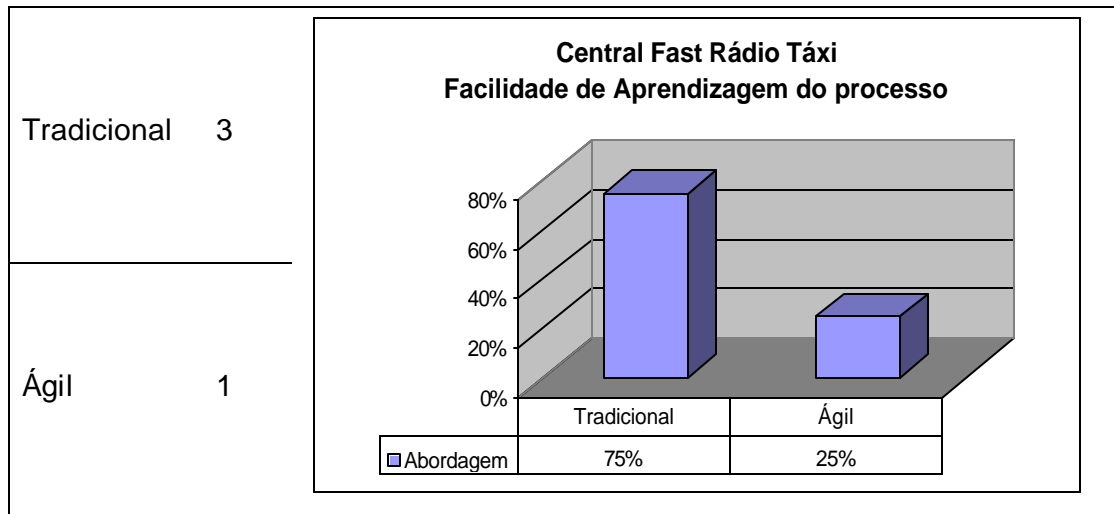


FIGURA 23: Projeto 3 – Facilidade de aprendizagem

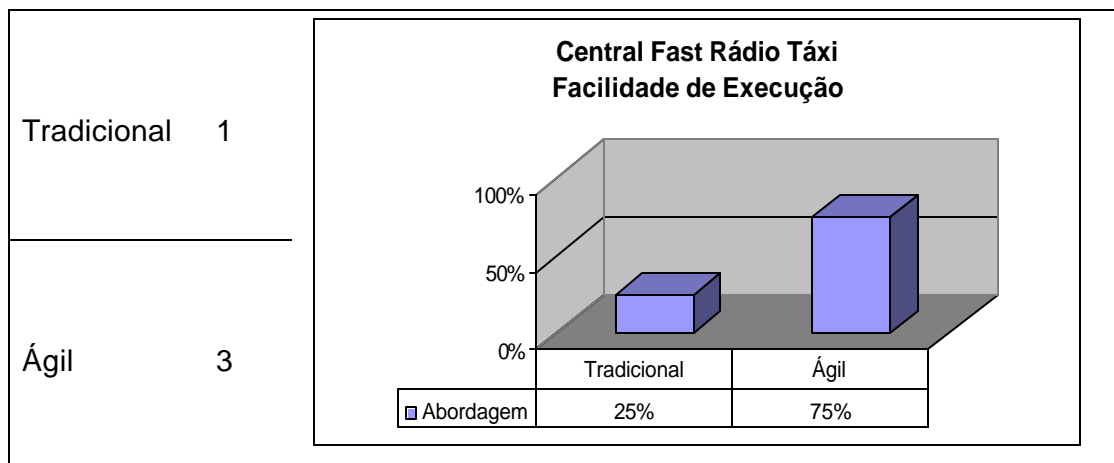


FIGURA 24: Projeto 3 - Facilidade de execução

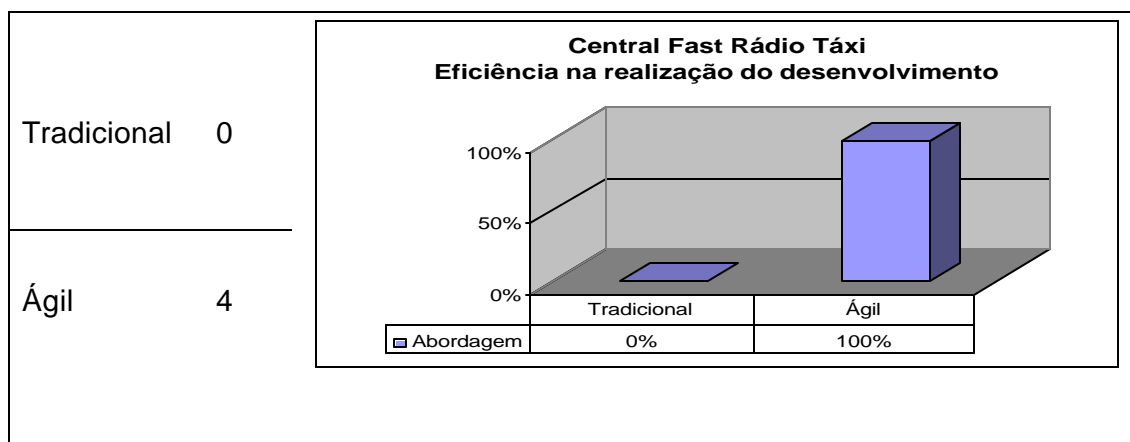


FIGURA 25: Projeto 3 - Eficiência na realização do desenvolvimento

4.5. Impacto do uso de métodos ágeis em relação à gestão do processo

O objetivo deste estudo de caso foi verificar como os alunos se comportariam frente às práticas propostas pela abordagem ágil, relacionadas à execução e ao gerenciamento do projeto, num projeto de maior porte, comparado aos três estudos de caso anteriores, e, ao final, obter o feedback desses alunos sobre a abordagem utilizada.

4.5.1. Projeto 4 – Cenário: PlastBom

A PLASTBOM é uma empresa do seguimento de produtos plásticos que fabrica pastas plásticas. Com o rápido crescimento da empresa, os diretores das áreas de Compras, Vendas, Produção e Financeira, resolveram informatizar os processos da empresa. Esse estudo de caso impõe aos participantes um maior nível de dificuldade, uma vez que exige a integração entre os módulos. É solicitado que sejam implementados os mecanismos de aprovação e determinação de níveis de tomada de decisão (alçada), sem os quais os processos não devem prosseguir. O estudo de caso detalhado se encontra no apêndice F.

4.5.1.1. Variáveis do estudo de caso

Participaram desse estudo de caso 47 alunos do quarto semestre do Curso de Tecnologia em Processamento de Dados da Faculdade de Tecnologia de São Paulo, do segundo semestre de 2007. No início do projeto, foi solicitado que os alunos respondessem a um questionário de identificação do perfil do grupo (apêndice B), para verificar se outros fatores, como experiência anterior, origem do ensino médio e outros itens pudessem afetar os resultados da pesquisa. Os dados coletados no questionário estão representados no apêndice G. Desse grupo, 51% dos participantes apontam a área de Análise de Sistemas como sendo a área de interesse no curso, 30%, a área de programação, e 19% indicam outras áreas de interesse. Em relação ao tipo de ensino médio cursado, 51% dos alunos cursaram, exclusivamente, o ensino médio em escolas públicas, 43%, exclusivamente, em

escolas particulares, e 6% estudaram a maior parte em escola particular. Cerca de 68% do grupo conhecem modelo de ciclo de vida, e 81% indicam que não têm conhecimento de métodos ágeis. Em relação à experiência em desenvolvimento de sistemas, 83% nunca exerceram atividades de desenvolvimento, e 77% não têm nenhuma experiência profissional. Os demais itens, coletados pelo questionário de identificação do grupo, não apontam fatores que pudessem distorcer os resultados da pesquisa.

Os alunos foram distribuídos em 3 equipes. Duas equipes desenvolveram o projeto utilizando a abordagem ágil, enfatizando os aspectos da auto-organização e reuniões semanais. Uma equipe desenvolveu utilizando a abordagem tradicional, seguindo o modelo de ciclo de vida Cascata.

Cada equipe foi decomposta em 4 grupos de trabalho, onde cada grupo ficou encarregado de desenvolver um módulo do estudo de caso: o módulo de compras, vendas, financeiro e produção. Cada grupo escolheu um líder, que ficou encarregado de discutir a integração entre os diversos módulos. Os líderes escolheram o gestor principal da equipe. O professor representou o papel do Dono do Produto.

4.5.1.2. Apresentação dos resultados

Foram definidas as funcionalidades por módulo, que deveriam ser entregues ao final do prazo estabelecido, 30 dias, tanto para aqueles que adotaram a abordagem ágil, quanto para aqueles que adotaram a tradicional, com e sem a utilização de um software gerador de aplicação.

A tabela 2 apresenta as horas gastas pelas equipes, no desenvolvimento do projeto, conforme relatório de atividades elaborado e entregue por cada grupo.

TABELA 2: Horas gastas por grupo no projeto PlastBom

Horas gastas por grupo no projeto PlastBom				
Grupo	Abordagem/Método	Ambiente de Desenvolvimento	Total hrs do grupo	Nro Participantes
G1	Método Ágil	Genexus	33 hrs	14
G2	Método Ágil	Genexus	495 hrs	14
G3	Abordagem tradicional	VB	328 hrs	19

As figuras 26 até a figura 29 apresentam a comparação entre os módulos desenvolvidos, e a figura 30 consolida os resultados.

O quadro 12 apresenta as funcionalidades entregues pelos grupos e a avaliação, quanto ao atendimento pleno (AT), parcial (AP) e não entregue ou não funcional (NA).

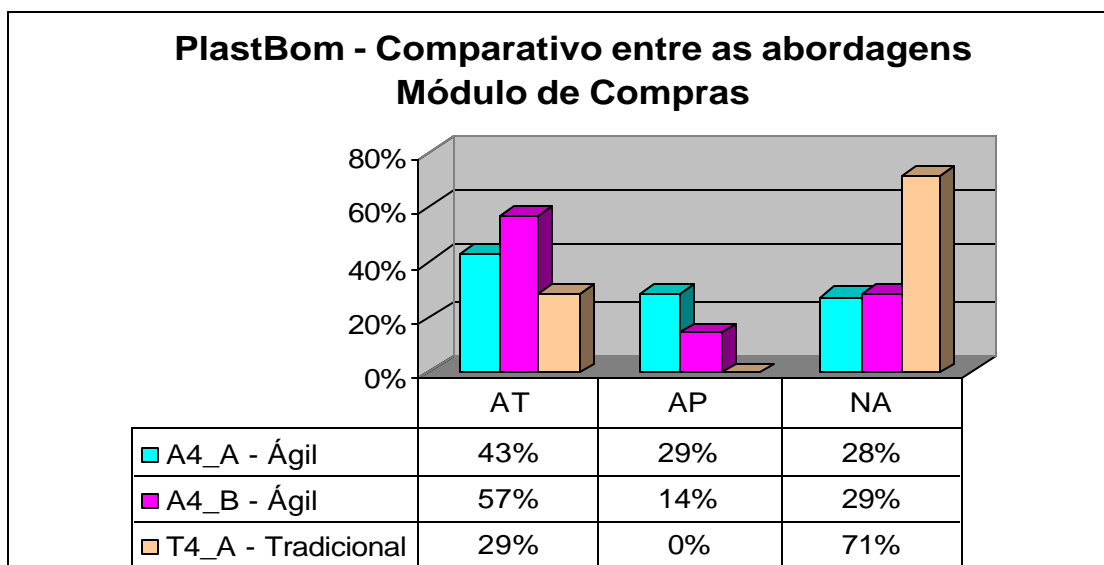


FIGURA 26: Projeto 4 – Comparativo entre as abordagens – Módulo de Compras

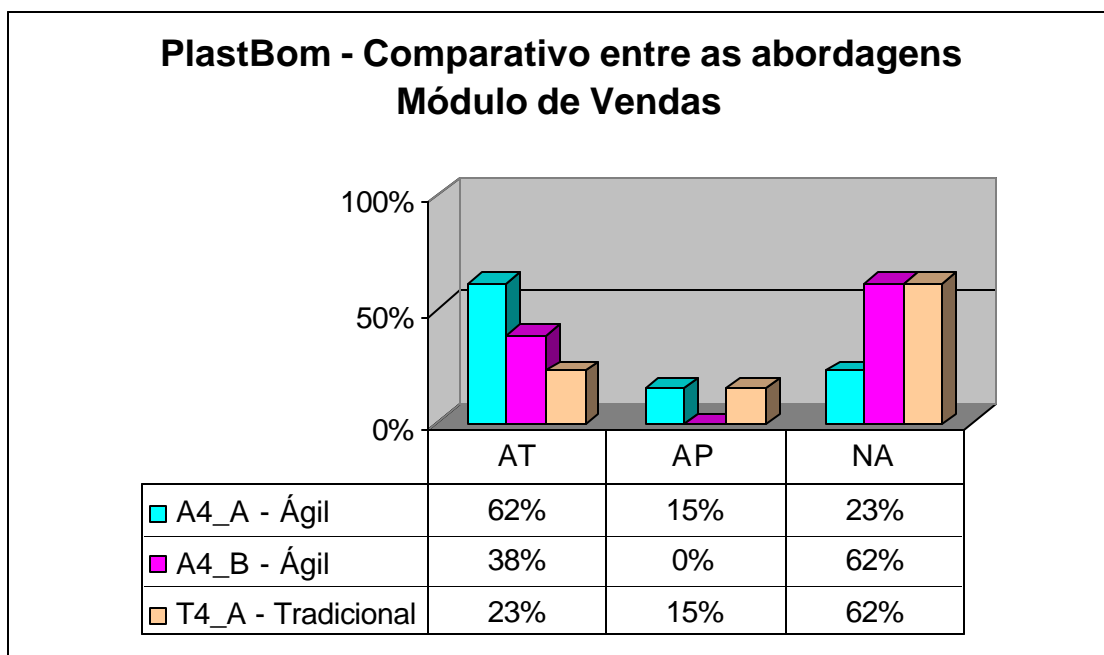


FIGURA 27: Projeto 4 – Comparativo entre as abordagens – Módulo de Vendas

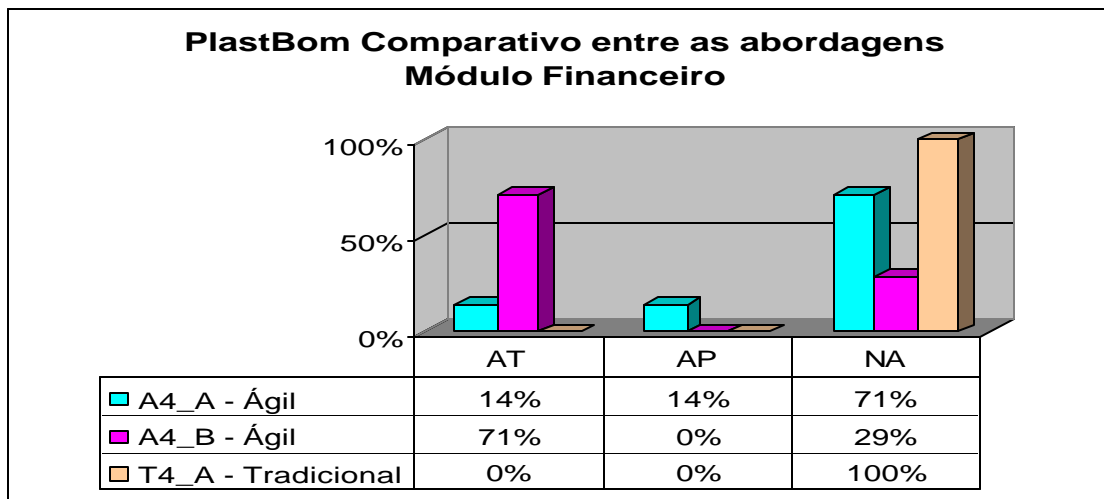


FIGURA 28: Projeto 4 – Comparativo entre as abordagens – Módulo Financeiro

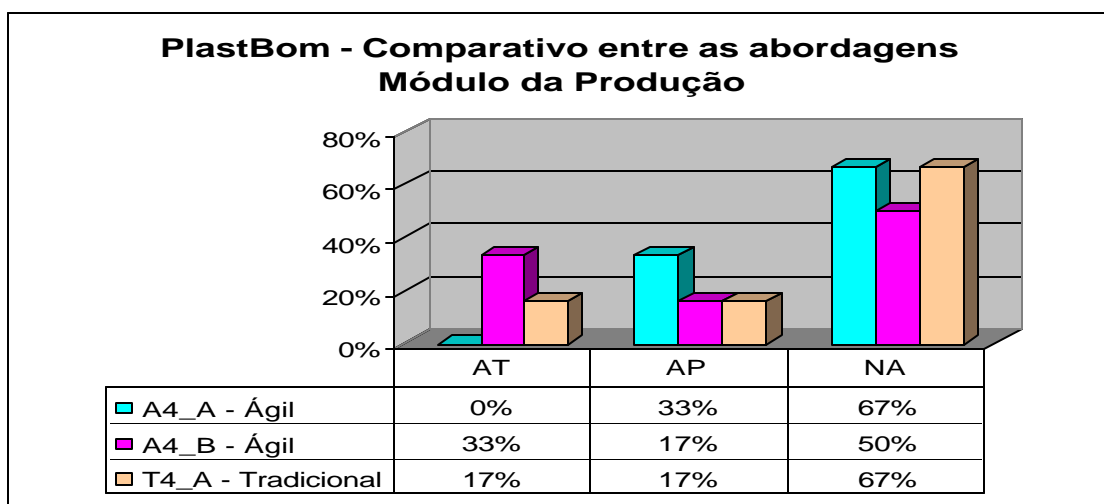


FIGURA 29: Projeto 4 – Comparativo entre as abordagens – Módulo da Produção

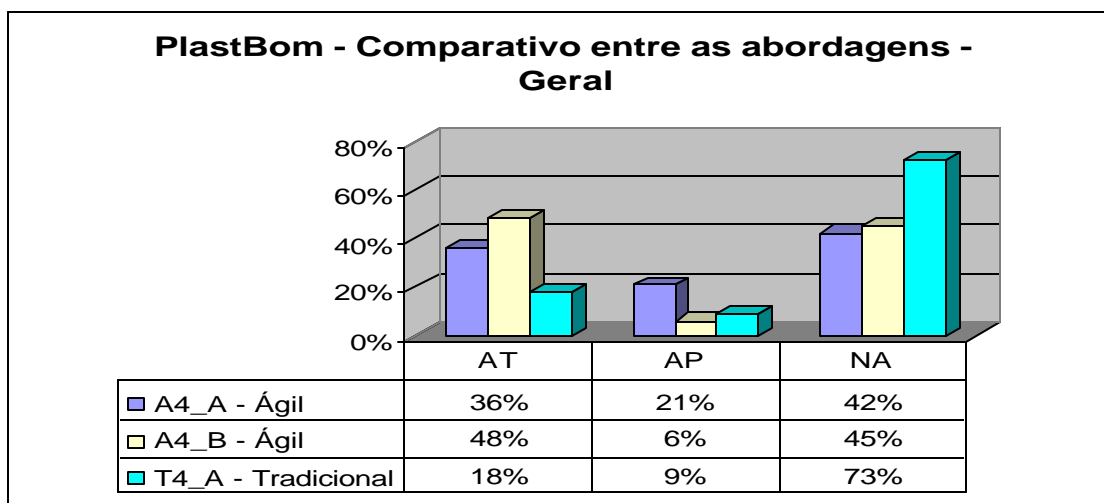


FIGURA 30: Projeto 4 – Comparativo entre as abordagens - Geral

Quadro 12: Funcionalidades implementadas no projeto 4

PlastBom - Funcionalidades Implementadas pelos grupos				
Módulo	Funcionalidades	A4 A	A4 B	T4
Compras	Cadastro de Fornecedores	AT	AT	AT
	Fornecimento	NA	AT	NA
	Cadastro de Produtos	AT	AT	NA
	Solicitação de Compras	NA	NA	NA
	Cotação	AP	AP	NA
	Pedido de Compras	AP	AT	AT
	Entrada de Mercadorias	AT	NA	NA
Vendas	Cadastro de Departamentos	AT	NA	NA
	Cadastro de Cargos	AT	NA	NA
	Cadastro de Funcionários	AT	NA	AP
	Cadastro de Regiões	AT	NA	NA
	Cadastro de Clientes	AT	AT	AT
	Associação de Vendedores a Clientes	NA	NA	AT
	Cadastro de Transportadora	AT	AT	NA
	Cadastro de Pedidos	AT	AT	AT
	Cálculo do Pagamento do Pedido	NA	AT	NA
	Programação de Entrega	AP	AT	NA
	Cancelamento do Pedido	AP	NA	NA
	Cadastro de Contatos	AT	NA	AP
	Aprovação do Pedido de Compras	NA	NA	NA
Financeiro	Cadastro de Bancos	AT	AT	NA
	Nota de Crédito	AP	AT	NA
	Contas a Pagar	NA	NA	NA
	Contas a Receber	NA	AT	NA
	Aprovação do Crédito do Cliente	NA	AT	NA
	AP	NA	AT	NA
	Aprovação do Pedido de Compras	NA	NA	NA
Produção	Cadastro da Estrutura do Produto	AP	AP	AP
	Cadastro de MP	AP	AT	AT
	Emissão de NFS	NA	NA	NA
	Cadastro de Unidade de Medida	NA	AT	NA
	Aprovação do Pedido de Compras	NA	NA	NA
	PCP	NA	NA	NA
Legenda		Abordagem Ágil	AT – Atendidas	
		Abordagem Ágil	AP – Atendidas parcialmente	
		Abordagem Tradicional	NA – Não atendidas	

4.5.1.3 Observações realizadas durante o estudo de caso

O líder, Scrum Master, escolhido do grupo A4_A, teve uma boa aceitação entre os componentes, além de se destacar nas atividades propostas pela disciplina. Os valores e práticas ágeis foram satisfatoriamente absorvidas e empregadas por esse grupo.

O grupo A4_B se caracterizava por ser formado por elementos motivados, mas que não conseguiam se auto-organizar e definir claramente as atividades, provocando consumo excessivo de horas para definições dos trabalhos a serem realizados.

O líder do grupo T4 teve que conciliar divergências entre os componentes do grupo e teve certa dificuldade para liderar sua equipe. Esse grupo tinha problemas importantes de relacionamento entre os participantes.

O grupo que adotou a abordagem tradicional gerou mais documentação do projeto. O grupo que seguiu a abordagem ágil não se preocupou com a documentação, gerando um mínimo necessário para o entendimento da equipe.

4.5.1.4. Análise dos resultados

Os dados consolidados apresentados na figura 29 demonstram que a produção foi significativamente maior naqueles que adotaram a abordagem ágil, com o uso de software gerador de aplicação. Se considerado o universo dos produtos não entregues (NA), constata-se que aqueles que adotaram a abordagem tradicional deixaram de entregar 73% das funcionalidades, enquanto os que adotaram o ágil deixaram de entregar 43,5% (média).

Sobressai a atuação do grupo A4_A, que adotou a abordagem ágil, e consumiu 33 horas implementando 12 funcionalidades avaliadas como AT, plenamente atendidas, enquanto o grupo A4_B, também ágil, consumiu 495 horas e implementou 16 funcionalidades. A produtividade do grupo A4_A foi de 2,75 horas por funcionalidade, enquanto o grupo A4_B foi de 30,94 horas por funcionalidade.

Conclui-se a partir dos resultados obtidos, que:

- a abordagem ágil contribui para a obtenção de maior produtividade;
- o uso de software gerador de aplicação também contribui para a maior produtividade;
- as competências, individual e coletiva, dos participantes influem significativamente na produtividade.

4.5.1.5. Avaliação das abordagens realizadas pelos participantes

Ao término do estudo de caso, os participantes foram convidados a opinar sobre questões relacionadas a:

- adequação das abordagens para uso em sala de aula, visando a aprendizagem;
- assimilação do processo de software a partir da abordagem adotada;
- facilidades oferecidas pela abordagem na execução das tarefas;
- eficiência da abordagem adotada na produção de artefatos.

A maioria dos participantes, 76%, considerou a abordagem tradicional como sendo a mais adequada para ser adotada em sala de aula, visando o aprendizado (figura 31). Quanto à facilidade para a assimilação do processo de software, 83% dos participantes entendem que a abordagem tradicional é melhor para a apresentação dos processos (figura 32). Quanto à facilidade de execução, 66% dos participantes consideraram a abordagem ágil mais adequada (figura 33). Sob a perspectiva da eficiência, 78% consideraram a abordagem ágil aquela que mais contribui para esse quesito (figura 34).

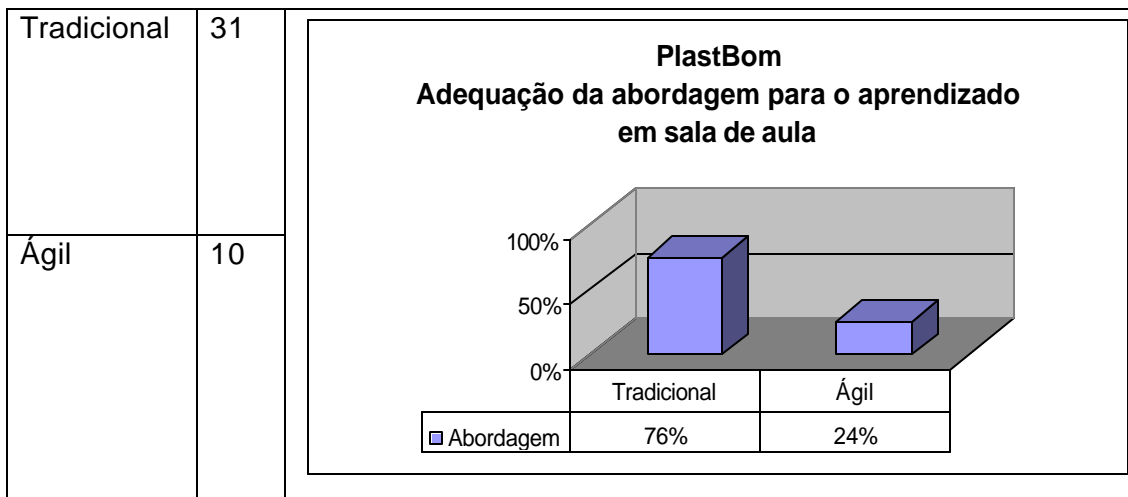


FIGURA 31: Projeto 4 - Adequação da abordagem para o aprendizado em sala de aula.

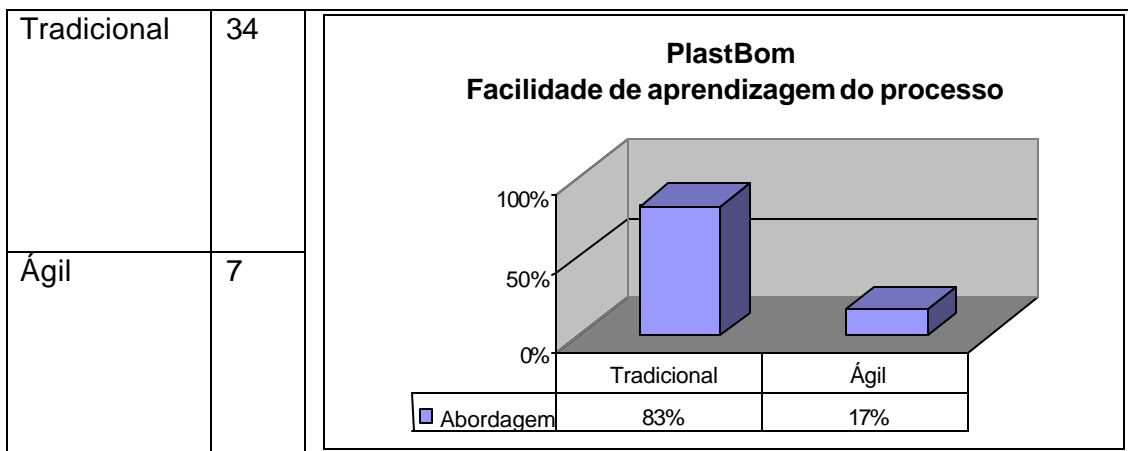


FIGURA 32: Projeto 4 - Facilidade de aprendizagem do processo

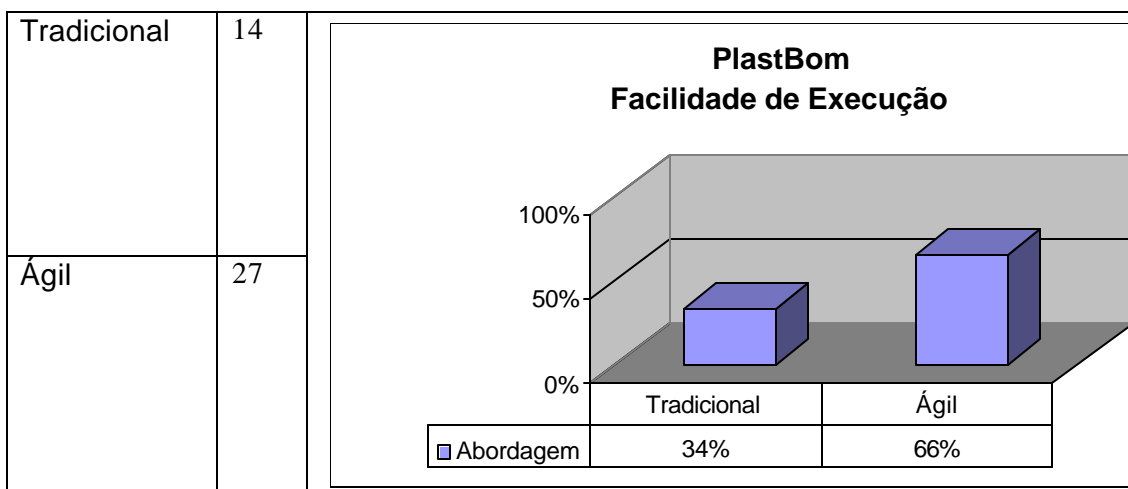


FIGURA 33: Projeto 4 - Facilidade de execução

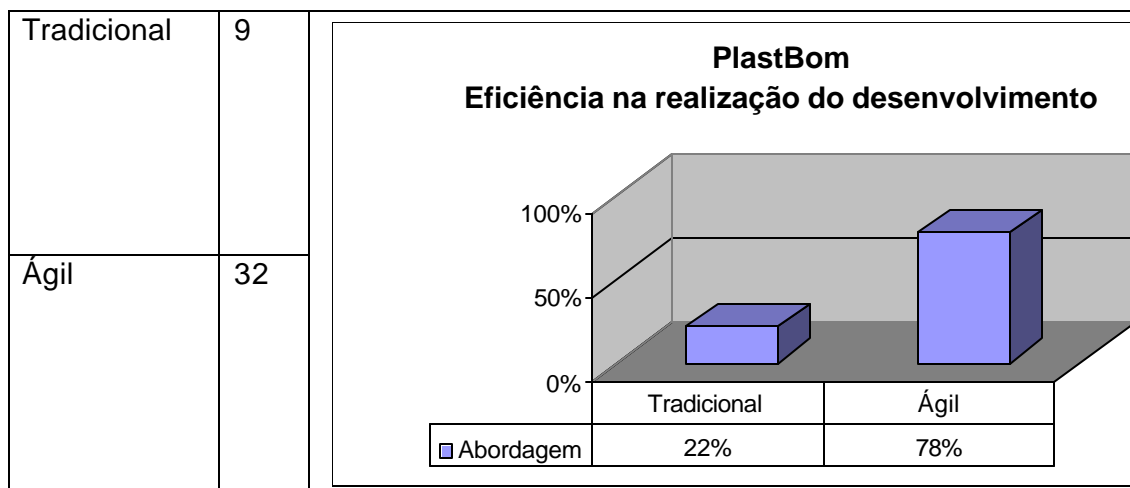


FIGURA 34: Projeto 4 - Eficiência na realização do desenvolvimento

Acrescentam:

- sob a perspectiva da aprendizagem da abordagem tradicional, os seguintes aspectos positivos: processo bem definido, preditivo, seguro, estável, e facilidade de aprendizagem/compreensão do processo. Consideram importante aprender o processo de forma linear, conhecendo todos os passos para o desenvolvimento de um sistema. Com o conhecimento do processo, eles alegam ter condições de reduzir alguns passos do processo, avaliando certos aspectos de um projeto, como tamanho do projeto, complexidade, tempo para a realização, entre outros;
- sob a perspectiva da aprendizagem da abordagem ágil, os seguintes aspectos positivos: a rapidez no desenvolvimento e, em consequência, a diminuição do tempo para a entrega de produtos para o cliente. Reconhecem a importância da entrega do produto de software para os clientes atendendo aos requisitos solicitados;
- com relação ao uso de software gerador de aplicação, o reconhecimento de sua utilidade, principalmente no que se refere à geração automática de código;
- em relação ao desenvolvimento auto-organizado, como aspectos positivos, a agilidade na tomada de decisão, facilidade e autonomia no processo de desenvolvimento. Citam como preocupação o risco da

falta de organização, caso não haja comprometimento e responsabilidade por parte da equipe;

- a importância que as reuniões frequentes tiveram na definição do escopo do projeto e, também, facilitando o acompanhamento das tarefas.

Outras considerações sobre a adoção das abordagens estão listadas no apêndice I.

4.6. Conclusão do estudo de caso

Os objetivos da aplicação dos quatro estudos de caso consistiram em verificar, na prática, os impactos da adoção da abordagem ágil na aprendizagem do processo de software.

Os resultados sintetizados, em especial as observações durante a execução dos estudos de caso, foram os seguintes:

- a abordagem ágil contribui para a obtenção de maior produtividade;
- os valores e práticas da abordagem ágil contribuem para que ocorra troca de experiências, a partir do momento em que incentiva a comunicação entre os envolvidos;
- tanto na abordagem ágil quanto na tradicional, sobressaem as competências individuais e as posturas adotadas frente aos trabalhos a serem executados. Os colaboradores classificados como “premium people” contribuem significativamente para o sucesso do projeto, independentemente da abordagem empregada;
- a adoção de software gerador de aplicação propiciou um alto desempenho no desenvolvimento das aplicações;
- o pré-conhecimento da abordagem tradicional mostrou-se um facilitador para a execução das atividades baseadas na abordagem ágil.
- a liderança exercida por algum integrante é fator de sucesso, não significando que haja uma hierarquia formal;
- a auto-organização constitui-se numa capacidade essencial de grupo;
- a abordagem tradicional é considerada a melhor abordagem para a aprendizagem do processo de software.

Considerações Finais

Este trabalho teve como objetivo verificar o impacto do uso de métodos ágeis no processo de ensino e aprendizagem da Engenharia de Software, no contexto do curso de Tecnologia de Processamento de Dados da Fatec-SP.

O reconhecimento prévio da amplitude e complexidade do tema levou a autora a realizar recortes, de maneira que se fixou, como traduzido no próprio título desta dissertação, o interesse no impacto dos métodos ágeis frente ao adotado, o tradicional, conhecido como modelo cascata ou ciclo de vida clássico.

A partir do estudo de caso aplicado constatou-se o seguinte:

- os métodos ágeis, XP e SCRUM, são atraentes, pois orientam a realização das atividades com menos formalidade e documentação, mostrando-se uma abordagem que pode contribuir, quando se pretende focar o saber-fazer, ou seja, o desenvolvimento das habilidades, conforme apresentado nas figuras 25 e 34;
- a abordagem tradicional, ciclo de vida clássico, por ser prescritiva, definida e seqüencial, oferece uma visão do processo e produtos mais evidentes, facilitando a aprendizagem do processo de software, conforme apresentado nas figuras 23 e 32;
- a realização dos trabalhos da abordagem ágil é facilitada a partir do pré-conhecimento da abordagem tradicional;
- a produtividade a partir dos métodos ágeis é maior que pela abordagem tradicional;
- os geradores de código contribuem para melhorar a celeridade do processo de software;

Diante desses resultados, mesmo considerando as possíveis influências das histórias de vida de cada participante, é possível afirmar que a adoção dos métodos ágeis resulta em melhora no desempenho do aprendiz e na construção dos produtos de software. Quando associados à utilização de geradores de código, a partir de uma mesma abordagem metodológica, os benefícios podem ser mais expressivos, se houver o domínio desse ambiente de desenvolvimento.

Vale ressaltar que a abordagem tradicional serviu como contraponto à abordagem ágil, uma vez que os participantes reconheceram a importância do seu pré-conhecimento.

É importante ressaltar que os avanços tecnológicos afetam desenvolvedores, usuários e processos utilizados no desenvolvimento de sistemas (Pressman, 2006), mas as pessoas envolvidas, que trazem consigo seus próprios estilos de trabalho, diferenças importantes quanto a habilidades, criatividade, regularidade, consistência e espontaneidade, propiciam também importante influência (Cockburn, 2006).

Como constatado, ao longo dos resultados apresentados, o fator humano constitui-se num aspecto essencial para o sucesso dos projetos, o que justifica uma pesquisa específica no contexto da engenharia de software, aplicando métodos ágeis. Essa constatação é corroborada por Boehm (2002), que enfatiza que a eficácia da adoção das abordagens ágeis deve merecer uma avaliação cuidadosa pelo fato de ter, no componente humano, um ponto importante para o sucesso da adoção da abordagem.

Este trabalho ateu-se a alguns aspectos dos métodos ágeis XP (extreme programming) e Scrum. Espera-se que este trabalho possa ter continuidade, a partir da realização de outros projetos de pesquisa sobre o tema. Destacam-se alguns assuntos que merecem ser abordados no futuro, tais como:

- o uso dos métodos ágeis em problemas de maior magnitude e criticidade no âmbito corporativo;
- a postura da comunidade acadêmica, focada em engenharia de software, para apresentar essas novas abordagens aos aprendizes, de modo a contribuir para minimizar ou eliminar os problemas crônicos que tem acompanhado a produção de software, ao longo dos anos.
- a gerência de projetos frente ao paradigma das abordagens ágeis;
- acompanhamento da atuação dos formandos no ambiente corporativo, com o intuito de verificar o quanto as abordagens estudadas contribuem profissionalmente.

Espera-se que estudos e pesquisas complementares possam ser conduzidos, a fim de contribuírem para o importante papel da comunidade acadêmica, que compartilha com a sociedade a missão de primar pela excelência da formação dos seus estudantes.

REFERÊNCIAS BIBLIOGRÁFICAS

- Amaral Filho, Antonio Rubens Anciães. Projeto estruturado: fundamentos e técnicas. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora S.A., 1988.
- Ambler, Scott. Modelagem ágil: práticas eficazes pra a programação extrema e o processo unificado. Porto Alegre: Bookman, 2004.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 12207: tecnologia da Informação – Processo de ciclo de vida de software. Rio de Janeiro, 1998.
- _____. NBR ISO/IEC 9126-1: Engenharia de Software – Qualidade de Produto. Rio de Janeiro, 2003.
- Bartié, Alexandre. Garantia da qualidade de software: adquirindo maturidade organizacional. Rio de Janeiro: Campus, 2002.
- Begosso, Luiz Ricardo. Ambiente para o desenvolvimento de maturidade em Engenharia de Software em um curso de Ciência da Computação. São Paulo, 2002. Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo.
- Bingham, J.E. Manual de análise de sistemas. Rio de Janeiro: Interciência, 1977.
- Boehm, Barry. Get Ready for Agile Methods, with Care. IEEE 2002.
- Boehm, Barry. A view of 20th and 21st century software engineering. ICSE 2006.
- Braga, Antônio. Análise de ponto de função. Rio de Janeiro: Infobook, 1996.
- Campos, Fábio B. et al. Abordagem em níveis para avaliação e melhoria de processos de software. V Simpósio Brasileiro de Qualidade de Software, 2006. www.sbc.org.br/bibliotecadigital/download.php?paper=971. Acessado em 06/01/2008.
- Carvalho, Luis Carlos de Sá. Análise de sistemas: o outro lado da informática. Rio de Janeiro: LTC – Livros Técnicos e Científicos, 1988.
- Castells, Manuel. A Sociedade em Rede. São Paulo: Editora Paz e Terra S.A., 2000.
- Cecil Martin, Robert. Agile Software Development: Principles, Patterns, and Practices. New Jersey: Pearson Education, Inc., 2003.
- Chen, Peter. Gerenciando banco de dados. São Paulo: McGraw-Hill, 1990.
- Chiossi, Thelma C. dos Santos; Moraes, Regina Lucia O. Especificação de sistemas de software utilizando Análise e Projeto Estruturados. Campinas, SP: Editora da Unicamp, 2006.

Chorafas, Dimitris N. Sistemas especialistas: aplicações comerciais. São Paulo: McGraw-Hill, 1988.

Coleman, Derek et al. Desenvolvimento orientado a objetos: o método fusion. Rio de Janeiro: Campus, 1996.

Cockburn, A.; Highsmith, J. Agile Software Development: The People Factor”, Computer, Nov. 2001, p.131-133.

Cockburn, A. Agile Software Development: The corporative game. USA: Addison Wesley, 2006.

DeMarco, Tom. Análise estruturada e especificação de sistemas. Rio de Janeiro: Campus, 1989.

Fisher, Alan S. CASE: utilização de ferramentas para desenvolvimento de software. Rio de Janeiro: Campus, 1990.

Fowler, Martin., “The New Methodology”, última atualização em 13 de dezembro de 2005, <http://www.martinfowler.com/articles/newMethodology.html>, acessado em 23/10/2007.

Furlan, José Davi. Modelagem de objetos através da UML – The Unified Modeling Language. São Paulo: Makron Books, 1998.

Gane, Chris; Sarson, Trish. Análise estruturada de sistemas. Rio de Janeiro: LTC – Livros Técnicos e Científicos S.A., 1984.

Gustafson, David A. Schaum’s Outlines of Theory and Problems of Software Engineering. USA: McGraw-Hill, 2002.

Keller, Robert. Análise estruturada na prática: desmistificando mitos. São Paulo: McGraw-Hill, 1990.

Larman, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objetos. Porto Alegre: Bookman, 2000.

Maffeo, Bruno. Engenharia de software e especificação de sistemas. Rio de Janeiro: Campus, 1992.

[Martin, 1991a] Martin, James. Engenharia da Informação: introdução. Rio de Janeiro: Campus, 1991.

[Martin, 1991b] Martin, James; McClure, Carma. Técnicas estruturadas e CASE. São Paulo: McGraw-Hill, 1991.

Martin, James. Princípios de análise e projeto baseados em objetos. Rio de Janeiro: Campus, 1994.

- Martin, James; Odell, James J. *Análise e projeto orientado a objeto*. São Paulo: Makron Books, 1995.
- McMenamin, Stephen M. *Análise essencial de sistemas*. São Paulo: McGraw-Hill, 1991.
- Melendez Filho, Rubem. *Prototipação de sistemas de informações: fundamentos, técnicas e metodologias*. Rio de Janeiro: LTC – Livros Técnicos e Científicos, 1990.
- Mendes, Antônio. *Arquitetura de software: desenvolvimento orientado para a arquitetura*. Rio de Janeiro: Campus, 2002.
- Molinari, Leonardo. *Testes de software: produzindo sistemas melhores e mais confiáveis*. São Paulo: Érica, 2003.
- Paula Filho, Wilson de Pádua. *Engenharia de software*. Rio de Janeiro: LTC – Livros Técnicos e Científicos S.A., 2003.
- PISELLO, Thomas; STRASSMANN, Paul. *IT Value Chain Management – maximizing the ROI from IT. Section 1: IT Spending and Financial Justification*. Alinean, LLC, 2003.
- Poppendieck, Mary. *Implementing lean software development: from concept to cash*. Boston: Addison Wesley, 2007.
- Pressman, Roger S. *Engenharia de software*. 3ª. edição. São Paulo: Makron Books, 1995.
- Pressman, Roger S. *Engenharia de software*. 5ª. edição. Rio de Janeiro: McGraw-Hill, 2002.
- Pressman, Roger S. *Engenharia de software*. 6ª. edição. São Paulo: McGraw-Hill, 2006.
- Rabello, Márcia Rodrigues; De Bartoli, Lis Ângela. *Estrela: modelo de um processo de desenvolvimento para aplicações de comércio eletrônico*. Passo Fundo: Ed. Universidade de Passo Fundo, 2006.
- Rocha, Ana Regina Cavalcanti da. *Análise e projeto estruturado de sistemas*. Rio de Janeiro: Campus, 1990.
- Rocha, Ana Regina Cavalcanti; Maldonado, José Carlos; Weber, Kival Chaves. *Qualidade de software*. São Paulo: Prentice Hall, 2001.
- Royce, W. "Managing the Development of Large Software Systems", In: IEEE WESCON, IEEE Press, pp. 1-9, San Francisco, 1970.
- Rozanski, Nick; Woods, Eoin. *Software Systems Architecture: working with stakeholders using viewpoints and perspective*. USA: Pearson Education, Inc., 2005.

Schwaber, Ken. Agile Project Management with Scrum. Redmond, Washington, EUA: Microsoft Press, 2004.

Setzer, Valdemar W.; Silva, Flávio Soares Correa da; São Paulo: Edgard Blucher, 2005.

Shlaer, Sally; Mellor, Stephen J. Análise de sistemas orientada para objetos. São Paulo: McGraw-Hill:Newstec, 1990.

Silva, Alexandre Freire de. Reflexões sobre o ensino de metodologias ágeis na academia, na indústria e no governo. 2007. 137f. Dissertação de Mestrado – Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2007.

Staa, Arndt von. Engenharia de programas. LTC – Livros Técnicos e Científicos S.A., 1987.

Sommerville, Ian. Engenharia de Software. São Paulo: Addison Wesley, 2003.

Swebok: Guide to the Software Engineering Body of Knowledge, 2004.
Disponível em <http://www.swebok.org/htmlformat.html>. Acesso em 15/11/2007.

Tate, Kevin. Sustainable Software Development – An Agile Perspective. USA: Pearson Education, Inc., 2006.

Taylor, David A. Engenharia de negócios com tecnologia de objetos. Rio de Janeiro: Axcel Books, 1995.

Teixeira, Sérgio Roberto Pinto. Engenharia de Software: experiência e recomendações. São Paulo, Edgard Blucher, 1979.

Teles, Vinicius Manhães. Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade. Novatec Editora Ltda: São Paulo. 2004.

Yourdon, Edward. Análise estruturada moderna. Rio de Janeiro: Campus, 1990.

Weiss, Sholom M. Guia prático para projetar sistemas especialistas. Rio de Janeiro: LTC – Livros Técnicos e Científicos S.A., 1988.

www.genexus.com. Acessado em 02/06/2007.

<http://www.agilemanifesto.org/principles.html>, acessado em 10/11/2007.

Apêndice A

Roteiro para o trabalho de Análise de Projeto de Sistemas

Esse roteiro orienta os alunos das disciplinas APS I, APS II e APS III, do período da tarde, e das disciplinas APSII e APSIII, do período da manhã.

Faculdade de Tecnologia de São Paulo
Departamento de Processamento de Dados
Análise e Projeto de Sistemas

Roteiro para o trabalho de APS	APS I	APS II	APS III
01. Dados da empresa ☒ Organograma ☒ Descrição do negócio Descrição das atividades da área objeto do levantamento	X		
02. Estudo de Necessidades ☒ Definição do problema ☒ Definição de objetivos Definição da Abrangência	X		X
03. Estudo de Viabilidade Técnica/Econômica	X		X
04. Planejamento	X	X	X
05. Levantamento da situação atual	X		
06. Representação da situação atual (DFD's)	X		
07. Documentos Levantados	X		
08. Especificar a abrangência do sistema	X		
09. Detalhar a alternativa escolhida	X		
10. Representação do novo sistema (DFD's)	X		
11. MER (utilizar notação de Peter Chen)	X	X	
12. Dicionário de Dados	X	X	
13. Projeto de Arquivos		X	
14. Projeto de Entradas (telas, formulários, etc.)		X	
15. Projeto de Saídas (telas, relatórios, etc.)		X	
16. Especificação de programas		X	X
17. Plano de testes		X	X
18. Procedimentos/Mecanismos de Controle e Segurança implementados ☒ Segurança lógica ☒ Segurança física		X	X
19. Plano de Treinamento			X
20. Manual do Usuário			X
21. Manual de Operação			X
22. Plano de Contingência			X
23. Sistema Executável (zero erros)/Documentado.			X

Apêndice B

Questionário de Identificação do perfil do Grupo



Identificação do perfil do grupo – Q1

Data ___/___/___

1) Dados pessoais	
Nome:	
Data de Nascimento:	Sexo: ()F ()M
Cursando:	Turno:
() APS I	() Manhã
() APS II	() Tarde
() APS III	() Noite
2) Área de Interesse	
() Análise de Sistemas	() Redes
() Programação	() Outros _____
3) Capacitação (Origem)	
a) Formação acadêmica	
a1) Que tipo de estudo você realizou?	
() Ensino Médio Comum	
() Curso Técnico	
() Curso para Magistério	
() Outro	
a2) Onde você realizou seus estudos de Ensino Médio?	
() Só em escola pública (Estadual ou Municipal)	
() Só em escola pública Federal	
() Só em escola particular	
() Maior parte em escola pública	
() Maior parte em escola particular	
() Metade em escola pública, metade em escola particular	
() Em supletivo	
a3) Em que período você realizou seus estudos de Ensino Médio?	
() Diurno (só manhã ou só tarde)	
() Diurno integral (manhã e tarde)	
() Noturno	
() Maior parte diurno	
() Maior parte noturno	
a4) Você já começou outro curso superior?	
() Não	
() Sim, mas abandonei	
() Sim, estou cursando	
() Sim, já concluí	

b) Cursos Livres	
b1) Já fez outros cursos na sua área de interesse?	
<input type="checkbox"/> Sim. Quais: _____ <input type="checkbox"/> Não	
c) Autodidata	
c1) Nos últimos 3 anos, quantos livros na sua área de interesse você leu?	
<input type="checkbox"/> Nenhum <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> mais de 3	
4) Conhecimentos Específicos	
4.1) Conhece Modelos de Ciclo de Vida?	
<input type="checkbox"/> Sim <input type="checkbox"/> Não	
4.2.) Conhece Métodos Ágeis?	
<input type="checkbox"/> Sim <input type="checkbox"/> Não	
5) Habilidades – Metodologias de Desenvolvimento e Manutenção de Sistemas (MDMS)	
5.1) Já desenvolveu utilizando alguma metodologia?	
<input type="checkbox"/> Sim <input type="checkbox"/> Não	
Se sim, qual?:	
<input type="checkbox"/> MDMS Tradicional <input type="checkbox"/> Métodos Ágeis	
6) Habilidades – Linguagem de programação	
6.1) Sabe programar em qual(s) linguagem(s)	
<input type="checkbox"/> Visual Basic <input type="checkbox"/> Delphi <input type="checkbox"/> VB .Net <input type="checkbox"/> Genexus <input type="checkbox"/> Outras. Qual(s)?: _____	
7) Experiência Profissional	
7.1) Em relação à experiência profissional em Desenvolvimento de Sistemas	
<input type="checkbox"/> Não tem experiência <input type="checkbox"/> Como programador. Quanto tempo? _____ <input type="checkbox"/> Como analista. Quanto tempo? _____ <input type="checkbox"/> Como analista de negócios. Quanto tempo? _____	
7.2) Outras atividades profissionais	
Função: _____	Tempo: _____
Função: _____	Tempo: _____

Apêndice C

Questionário de Avaliação I – Avaliação das abordagens



Questionário de Avaliação I

Nome: _____ Data: ____/____/____

Disciplina: _____ Turno: _____

1. Você considerou que a realização do desenvolvimento do sistema foi mais eficiente em qual das abordagens?
 1. Tradicional
 2. Métodos Ágeis
2. Você considera que a aprendizagem sobre o processo de desenvolvimento de sistemas foi mais fácil utilizando qual das abordagens?
 - a. Tradicional
 - b. Métodos Ágeis
3. A sistemática de trabalho (execução das atividades) que você considerou mais adequada foi aquela oferecida por qual abordagem?
 - a. Tradicional
 - b. Métodos Ágeis
4. Qual das abordagens metodológicas você considera mais adequada para a aprendizagem do processo de desenvolvimento de sistemas pelos seus colegas em sala de aula?
 - a. Tradicional
 - b. Métodos Ágeis
5. O processo de desenvolvimento, a partir da abordagem ágil, mostrou-se mais fácil de aprender?
6. Quais aspectos você considerou negativos, sob a perspectiva da aprendizagem, na abordagem tradicional?
7. Quais aspectos você considerou positivos, sob a perspectiva da aprendizagem, na abordagem tradicional?
8. Quais aspectos você considerou negativos, sob a perspectiva da aprendizagem, na abordagem ágil?
9. Quais aspectos você considerou positivos, sob a perspectiva da aprendizagem na abordagem ágil?

Apêndice D

Questionário de Avaliação II



Questionário de Avaliação II

Nome: _____ Data: ____/____/____

Disciplina: _____ Turno: _____

- 1) Você considerou que a realização do desenvolvimento do sistema foi mais eficiente em qual das abordagens?
 - a. Tradicional
 - b. Métodos Ágeis
- 2) Você considera que a aprendizagem sobre o processo de desenvolvimento de sistemas foi mais fácil utilizando qual das abordagens?
 - a. Tradicional
 - b. Métodos Ágeis
- 3) A sistemática de trabalho (execução das atividades) que você considerou mais adequada foi aquela oferecida por qual abordagem?
 - a. Tradicional
 - b. Métodos Ágeis
- 4) Qual das abordagens metodológicas você considera mais adequada para a aprendizagem do processo de desenvolvimento de sistemas pelos seus colegas em sala de aula?
 - a. Tradicional
 - b. Métodos Ágeis
- 5) O processo de desenvolvimento, a partir da abordagem ágil, mostrou-se mais fácil de aprender?
- 6) Quais aspectos você considerou negativos, sob a perspectiva da aprendizagem, na abordagem tradicional?
- 7) Quais aspectos você considerou positivos, sob a perspectiva da aprendizagem, na abordagem tradicional?

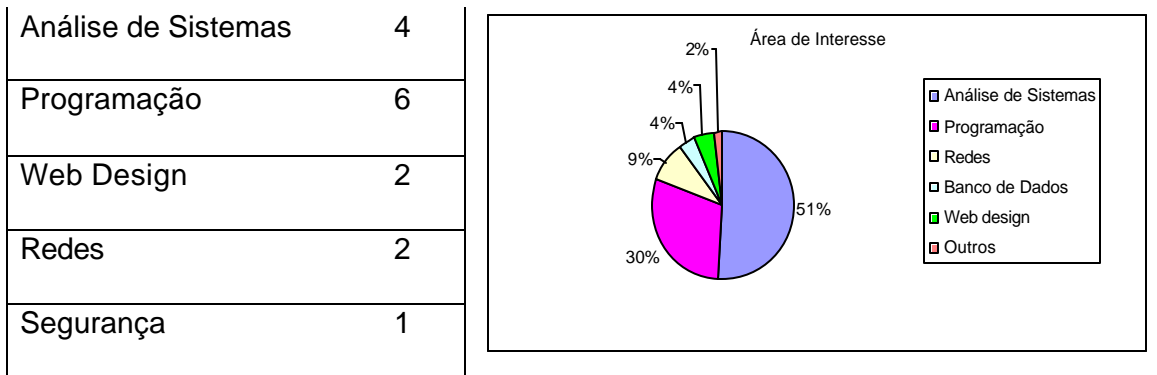
- 8) Quais aspectos você considerou negativos, sob a perspectiva da aprendizagem, na abordagem ágil?
- 9) Quais aspectos você considerou positivos, sob a perspectiva da aprendizagem, na abordagem ágil?
- 10) Indique a linguagem de programação utilizada. Justifique a escolha.
- 11) Foi utilizado o gerador de aplicação Genexus versão 8.0? Se sim, justifique a escolha.
- 12) Se utilizou o gerador de aplicação, liste os pontos positivos e negativos constatados a partir do uso dessa ferramenta.
- 13) Anexe o relatório de horas atividades dedicadas ao projeto, detalhando as atividades realizadas e o tempo gasto para executá-las.
- 14) Considerando que foi solicitado o desenvolvimento auto-organizado, relate os aspectos positivos e negativos dessa experiência.
- 15) Relate como aconteceu a integração dos módulos desenvolvidos e as ações empreendidas por cada participante, em especial o líder, visando essa integração.
- 16) Relate como foram definidas as prioridades e as respectivas modificações dessas prioridades.
- 17) Relate a sua opinião quanto à validade das reuniões semanais.
- 18) Quanto ao papel do líder, você considera que o mesmo adotou uma postura que motivou o grupo a realizar as tarefas designadas?

Apêndice E

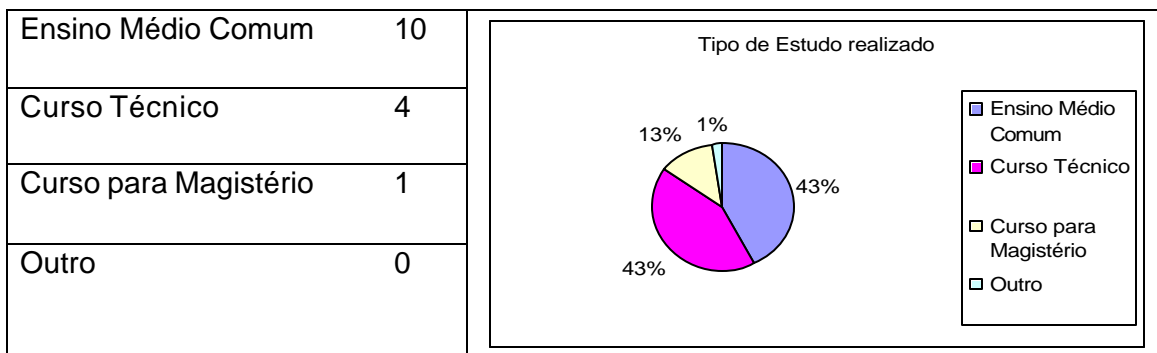
Perfil dos estudantes do estudo de caso “Central Fast Rádio Táxi”

Este apêndice contém os resultados consolidados da identificação do perfil do grupo que participou do projeto Central Fast Rádio Táxi.

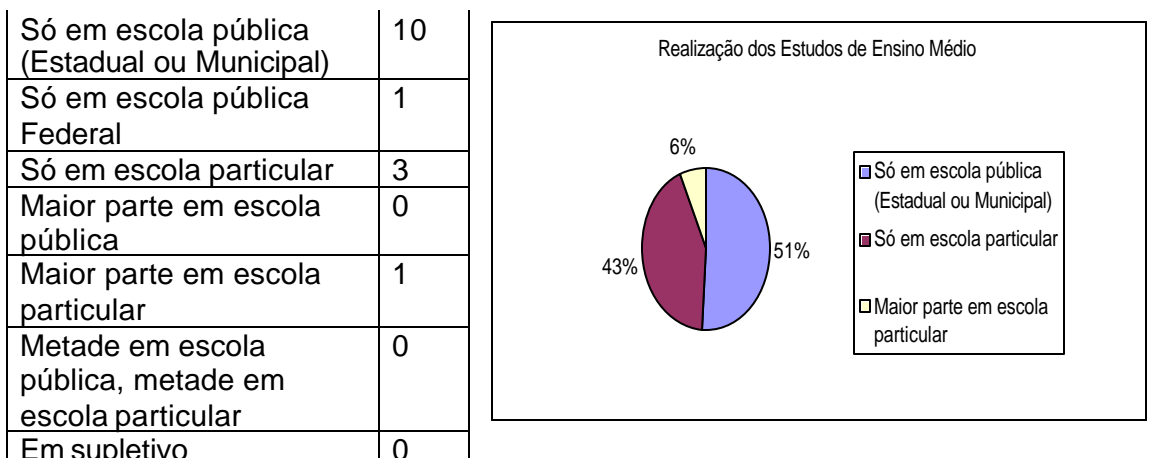
1) Área de Interesse



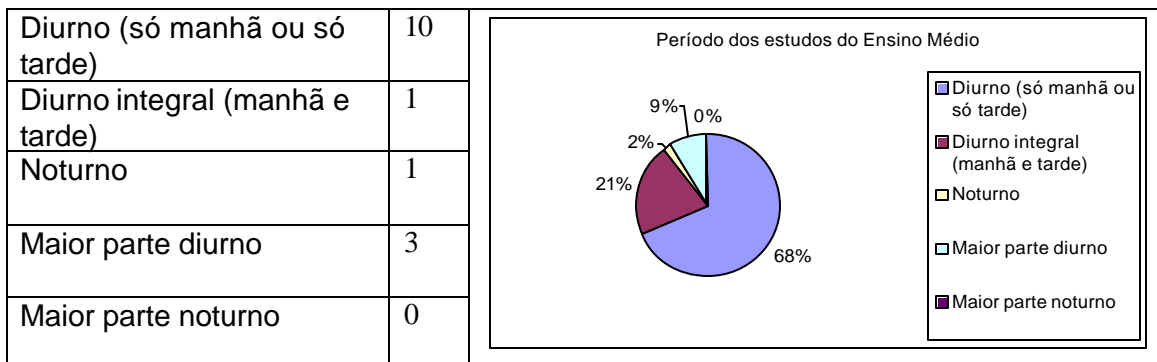
2) Que tipo de estudo você realizou?



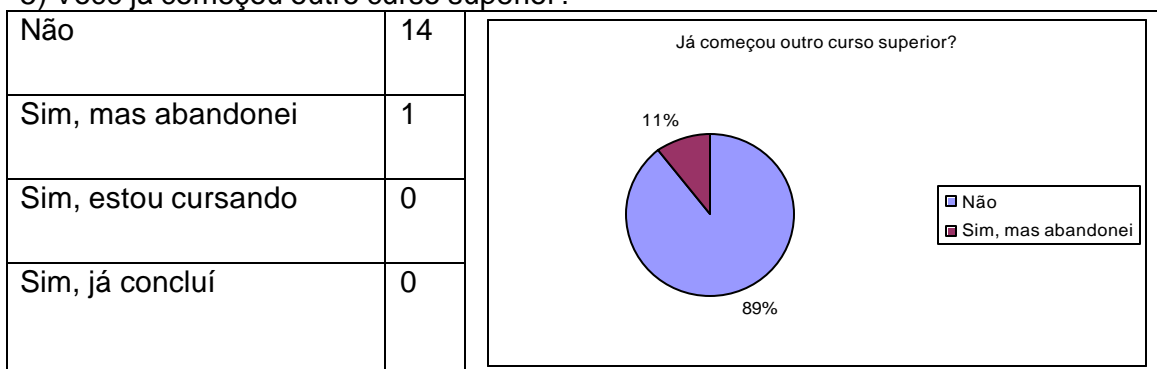
3) Onde você realizou seus estudos de Ensino Médio?



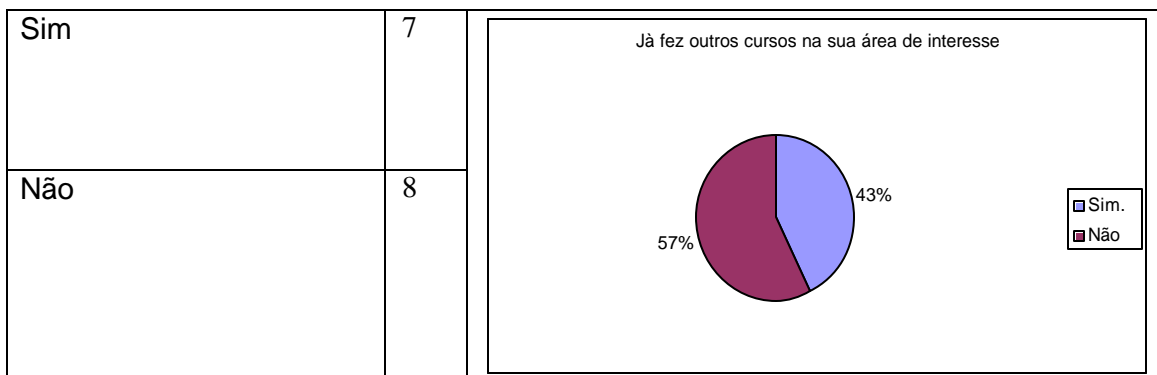
4) Em que período você realizou seus estudos de Ensino Médio?



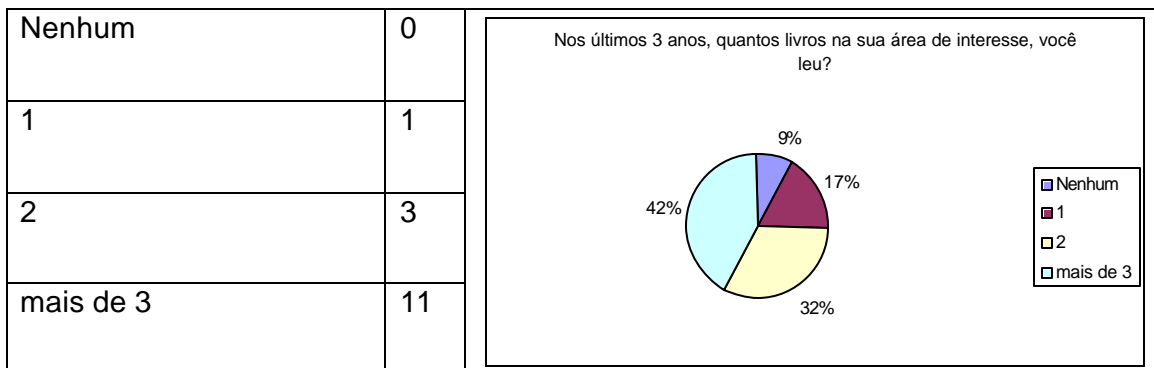
5) Você já começou outro curso superior?



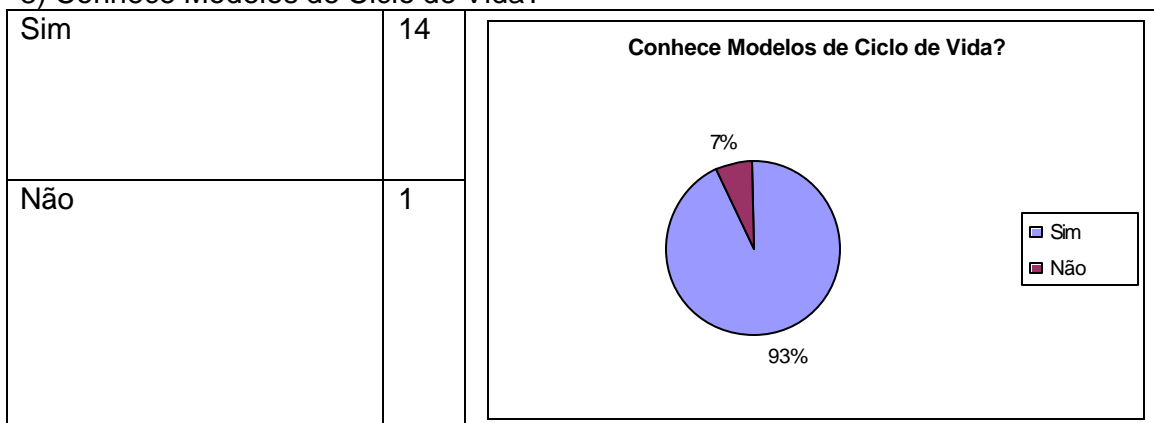
6) Já fez outros cursos na sua área de interesse?



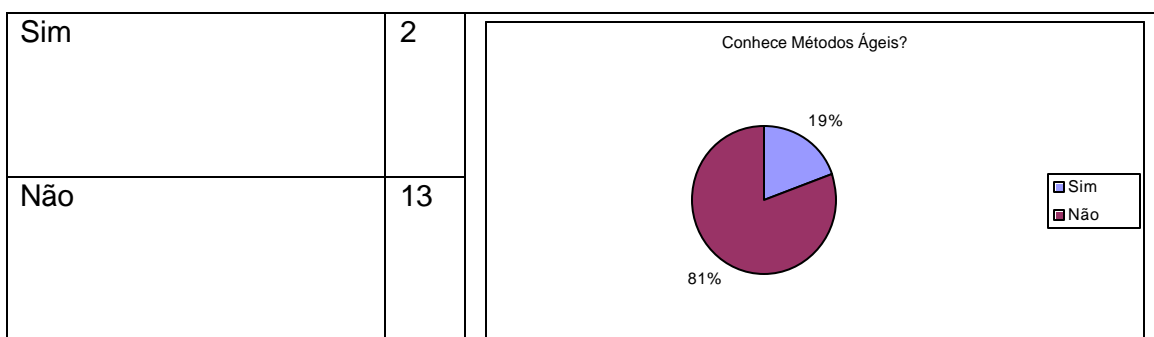
7) Nos últimos 3 anos, quantos livros na sua área de interesse você leu?



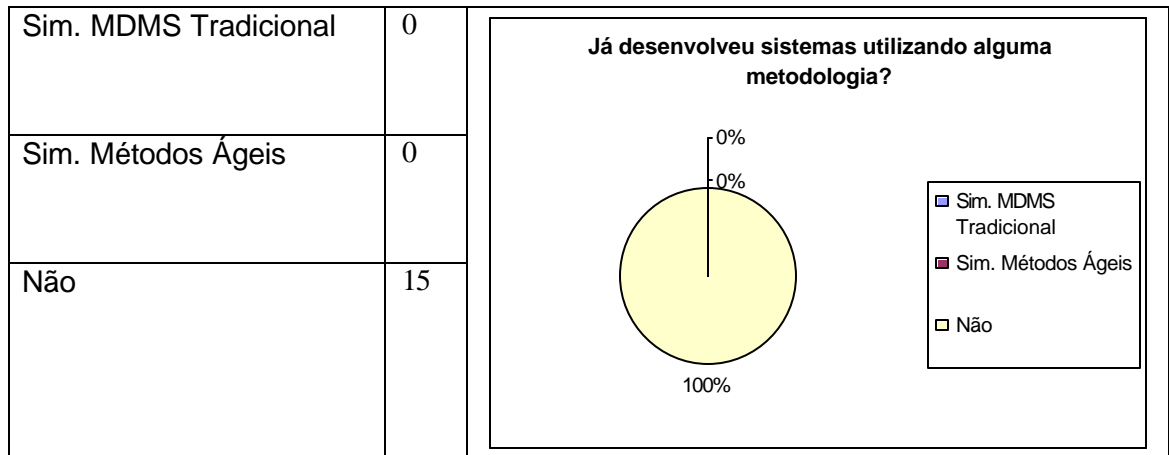
8) Conhece Modelos de Ciclo de Vida?



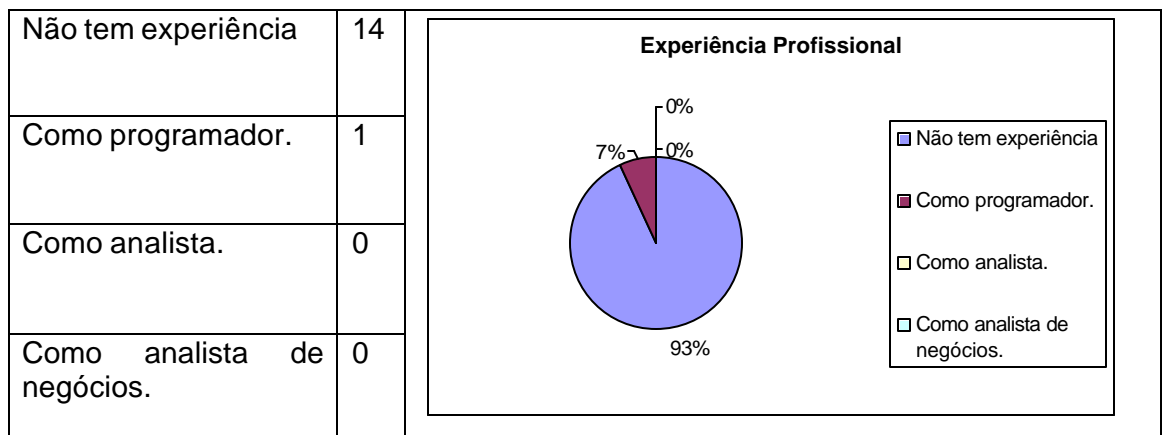
9) Conhece Métodos Ágeis?



10) Já desenvolveu sistemas utilizando alguma metodologia?



11) Em relação à experiência profissional em Desenvolvimento de Sistemas



Apêndice F

Estudo de Caso PlastBom

Os diretores da empresa PLASTBOM, também proprietários da mesma, resolveram investir no seguimento de produtos plásticos em 02 de janeiro de 2000. Após a constituição da empresa foram adquiridos equipamentos, máquinas, matérias-primas e demais produtos necessários no processo de fabricação das pastas plásticas. Os diretores detinham grande conhecimento e experiência, adquiridos durante 35 anos de trabalhos realizados em multinacionais desse mesmo setor. Nesse momento, também foram contratados os primeiros colaboradores (funcionários) para o início dessa empreitada.

O Sr. José da Silva, um dos diretores da empresa, ficou sendo o responsável pela área de vendas. O Sr. João Oliveira, também sócio, ficou sendo o responsável pela área de produção. O Sr. André Duarte, também sócio, ficou sendo o responsável pela área de compras. E, finalmente, o Sr. Heitor Honesto, também sócio, ficou sendo o responsável pela área financeira. Os diretores resolveram, nesse início, terceirizar a contabilidade e assuntos fiscais.

A diretoria realizava uma reunião toda segunda-feira (das 8:00hs às 10:00hs), pontualmente, para avaliar os resultados da semana anterior e traçar as metas para os dias seguintes. Nessa reunião, eram estabelecidas as metas que deveriam ser atingidas no mês. Além disso, eles procuravam estabelecer objetivos que deveriam ser atingidos no semestre e no ano.

No primeiro ano, a PLASTBOM fechou o ano com 100 clientes. Em 2001, já eram 1000 clientes. No ano seguinte, a clientela aumentou para 3000. Em 2003, o Sr. José e toda a diretoria comemoravam ter conquistado a marca de 5000 clientes ativos. Em 2004, já ultrapassavam a marca dos 6000 clientes. Atualmente, já são mais de 8000 clientes ativos.

Esse rápido crescimento da carteira de clientes só foi possível graças à estratégia do Sr. José, que resolveu, em comum acordo com os demais diretores, contratar representantes de vendas e vendedores (colaboradores externos), que eram comissionados e premiados mensalmente, se atingissem as metas estabelecidas pelo Sr. José. Cada vendedor ou representante tinha uma meta estabelecida.

O dia-a-dia da Área de Vendas

O Sr. José da Silva tem um assistente de vendas. Os pedidos são recebidos pelo assistente de vendas. Quando o pedido é feito por telefone, durante o

telefonema, o assistente transcreve o que é pedido num formulário denominado de PEDIDO DE VENDAS. Nesse pedido, é identificado o cliente (razão social, cnpj, inscrição estadual, endereço completo, condições de pagamento, forma de pagamento, itens pedidos, quantidades de cada item pedido, preço unitário negociado, valor total do item pedido, valor total do pedido, transportador de preferência do cliente, data de entrega combinada, entre outros itens). Os pedidos, a critério do cliente, podem ser entregues em "n" remessas, ou seja, o pedido pode ser atendido através de diversas entregas. A condição de pagamento pode ser fixada em "n" vencimentos, desde que aprovado pelo Sr. José. Já o comprometimento em relação às entregas necessita da ciência e aprovação do Sr. João. Cabe ao Sr. João, independente da quantidade, avaliar se é possível atender o pedido. Caso haja alguma restrição informada pelo Sr. João, o Sr. José imediatamente informa o cliente sobre a restrição e "tenta" renegociar os prazos de entrega, de acordo com o que foi previamente informado pelo Sr. João. Não havendo nenhuma restrição, o pedido é confirmado e é enviada uma correspondência ao cliente, confirmando o pedido e todas as condições negociadas.

Tendo sido o pedido totalmente aprovado, o Sr. José encaminha cópia desse pedido para o Sr. João (produção), o Sr. André (compras) e o Sr. Heitor (financeiro) para ciência e providências que couberem a cada uma das áreas.

O Sr. José recebe, diariamente, a posição dos pedidos atendidos, em atendimento e em estado de espera. Caso seja detectado qualquer problema que possa provocar algum atraso que não possa ser evitado, o Sr. José imediatamente entra em contato com o cliente, para informá-lo do imprevisto e negociar uma prorrogação do prazo de entrega ou outra alternativa, como por exemplo, a realização de entregas parciais, para minimizar ou eliminar o possível impacto que o atraso possa acarretar ao cliente. Em condições normais de atendimento os produtos são faturados e entregues através da transportadora escolhida previamente, constante no pedido.

O Sr. José monitora todas as entregas e dá por concluído o processo de atendimento a determinado pedido, no momento em que o cliente confirma o seu recebimento e dá o aceite do produto enviado.

Após a entrega, o Sr. José sempre liga para o cliente, para saber se está satisfeito com o produto enviado, e questiona o cliente se há algo mais em que a PLASTBOM possa atendê-lo.

Todos os contatos realizados pelo Sr. José são, rigorosamente, anotados na sua agenda. Essas anotações ele as faz de próprio punho. Por imposição do Sr. José, o assistente de vendas, Aguinaldo, também adota a mesma sistemática. Anota tudo na sua agenda.

Nas reuniões semanais, realizadas sempre nas segundas-feiras, o Sr. José apresenta o balanço das vendas da semana anterior e projeta a sua meta para os dias seguintes e para o mês. Nessa reunião, ele apresenta a evolução das vendas (mês a mês), desde o início da operação da empresa. Além da análise horizontal, ele sempre se preocupa em apresentar a evolução das vendas comparada ao mesmo período dos anos anteriores, visando dar explicações sobre um determinado resultado frente a questões sazonais que possam ajudar a identificar as possíveis causas dos comportamentos de vendas observados.

Os vendedores e representantes participam, semestralmente, de um evento realizado exclusivamente para a premiação dos mais bem sucedidos. O relatório contendo o "ranking" é distribuído a todos, para conhecimento e busca de novos patamares de sucesso. Esse ranking tomou-se um importante instrumento de motivação, que acabou impulsionando toda a equipe de vendas, na busca de melhores resultados. O Sr. José também premia aquele que, percentualmente, evoluiu significativamente, mesmo não estando entre os que obtiveram os melhores resultados. Esses incentivos acabam fazendo com que sua equipe de vendas alcance freqüentemente as metas estabelecidas.

A equipe do Sr. José é composta por sete colaboradores. O Aguinaldo ocupa o cargo de gerente de vendas, e os seis colaboradores (Valter, Jair, Eduardo, Antonio, Junior e Fernando) são os supervisores de vendas. Cada supervisor é responsável por uma determinada região previamente demarcada. Os clientes de um vendedor não podem, em hipótese alguma, ser "trabalhados" por outro vendedor; é proibido. A partir do momento em que um vendedor realiza o cadastro de um cliente, esse cliente fica associado a esse vendedor, até o momento do seu desligamento/afastamento ou através de um remanejamento autorizado pelo Sr. José.

O Sr. José sempre incentivou o respeito de que o cliente é merecedor. Para o Sr. José, o cliente deve ser sempre tratado com respeito, e as suas "vontades" sempre atendidas, na medida do possível, mesmo que venham a causar algum transtorno para a PLASTBOM. Exemplos desses transtornos são, entre outros, os

cancelamentos de pedido realizados pelos clientes na última hora, ou seja, quase que na véspera da entrega do produto. Esses cancelamentos, às vezes, são parciais, mas há aqueles que cancelam todo o pedido. O Sr. José tenta negociar, buscando reverter a situação. Em alguns casos, consegue, através do remanejamento das datas e quantidades de entrega. Em alguns casos, é obrigado a rever os preços praticados, podendo até oferecer descontos ou condições de parcelamento que possam ser cumpridas pelos clientes. Outro fato que afeta a PLASTBOM é a inadimplência (atraso ou não pagamento). A área financeira, gerida pelo Sr. Heitor, tão logo percebe a situação de atraso, informa ao Sr. José, para que tenha ciência da situação do cliente da PLASTBOM e possa "ajudar" de alguma forma. Quando essas ocorrências, que sinalizam alguma dificuldade enfrentada pelo cliente, são identificadas, imediatamente todos os diretores são informados, para que - se necessário - alguma iniciativa seja posta em prática, como, por exemplo, o bloqueio dos pedidos que se encontram nas fases que precedem o respectivo atendimento, produção ou entrega. Outro fato importante são as devoluções. As devoluções acontecem por diversos motivos, entre os quais se destacam: produtos fora da especificação do cliente, feriado na cidade, empresa fechada, devolução parcial decorrente de defeito de fabricação, etc.

Todas as entradas e saídas de produtos na PLASTBOM só são permitidas desde que acompanhadas das respectivas notas fiscais. Nenhum produto pode entrar ou sair da empresa sem ter suporte documental, exigido pela legislação em vigor.

Todos os clientes cadastrados na PLASTBOM recebem um crédito. Esse valor é atribuído pela área financeira que, baseada nos documentos e informações encaminhadas, decide sobre o valor do crédito que será concedido ao cliente. É habitual o vendedor sugerir um limite de crédito. Cabe à área financeira definir o limite de crédito que será concedido ao cliente. Esse limite de crédito só é aprovado pelo Sr. Heitor. Um cliente só pode ter pedidos para serem atendidos, desde que não ultrapassem o limite de crédito concedido. Caso esse valor ultrapasse o valor estabelecido, a área financeira é acionada, para decidir se o pedido poderá ou não ser aprovado. Todo pedido é aprovado pela área financeira (quanto ao crédito), pela produção (quanto à viabilidade do atendimento) e pela área de vendas.

Após o pedido de venda ter sido aprovado por todas as instâncias, a área de produção insere esse pedido no seu plano de produção. A partir desse plano de

produção, são geradas as ordens de fabricação. Essas ordens de fabricação são sempre associadas aos respectivos pedidos. Nesse momento, se detectado que há ou haverá falta de matéria-prima, a área de compras, gerida pelo Sr. André, é imediatamente acionada através de uma requisição de compra de matéria prima. Essa requisição tem a identificação de todas as matérias-primas e respectiva quantidade sugerida de compra. A área de compras tem como política comprar a partir da realização de cotações. Qualquer cotação deve ser feita considerando os preços, condições e características dos produtos de pelo menos três fornecedores. Após a escolha do fornecedor, o pedido de compra é enviado ao fornecedor. Cabe à área de compras monitorar o processo de compra, para que a matéria-prima seja entregue na data combinada. O atraso, se acontecer, poderá prejudicar o plano de produção. A área de compras deve buscar alternativa, em tempo, caso perceba que alguma entrega de matéria-prima não será realizada nos prazos e condições combinados.

O dia-a-dia da Área Financeira (Crédito e Cobrança)

O Sr. Heitor Honesto, também sócio, é responsável pela área financeira. Ele, em comum acordo com os demais sócios diretores, resolveu instituir algumas regras para:

- estabelecimento de limite de crédito aos clientes, ou seja, uma Política de Crédito;
- estabelecimento de critérios para concessão de condições de pagamento (à vista, 7 dias da data, 14 dias da data da emissão da nota fiscal, 21 dias da data da emissão da nota fiscal, 30 dias da data da emissão da nota fiscal, e outros que julgar necessário);
- estabelecimento de regras para o parcelamento (somente quando aprovado por no mínimo 2 diretores, o Diretor Financeiro e o Diretor Comercial);
- estabelecimento de regras para a gestão do contas a receber e contas a pagar.

Todos os vendedores ou representantes de vendas encaminham os cadastros dos clientes novos para a gerência da área comercial. A área comercial, após sua análise e aprovação, encaminha o cadastro dos clientes novos para a área financeira, para a atribuição do limite de crédito. Os colaboradores da área

financeira, responsáveis pela atribuição do limite de crédito, têm alçadas definidas a saber:

- o colaborador de nível 1 pode aprovar limites de crédito de até R\$5.000,00;
- o colaborador de nível 2 pode aprovar limites de crédito de até R\$15.000,00;
- o colaborador de nível 3 pode aprovar limites de crédito de até R\$50.000,00;
- o colaborador de nível 9 pode aprovar limites de crédito acima de R\$50.000,00.

Após a concessão do limite de crédito, que implica na aprovação do crédito pela área financeira, todos os envolvidos (vendedor ou representante, supervisores de vendas, gerência da área comercial e gerência da área financeira) são informados da aprovação, para ciência ou providências.

Os vendedores ou representantes podem, também, solicitar alterações cadastrais ou alterações de limite de crédito. Também podem solicitar o bloqueio ou reativação de clientes. Quando se tratar de alterações de dados cadastrais, apenas a área comercial se envolverá para a atualização dos dados do cliente. Quando implicar em limite de crédito e bloqueio ou reativação de clientes, além da gerência comercial, a gerência financeira deverá se manifestar.

Após as vendas efetivadas, ou seja, após a emissão das notas fiscais e a comprovação do efetivo recebimento dos produtos constantes dessas notas fiscais, a área financeira, que tem o registro de cada parcela, aguarda a liquidação nas respectivas datas de vencimento. Caso o título não seja liquidado até dois dias do vencimento, o cliente é contatado para verificar o motivo. Nesse contato, pode ser realizada alguma tratativa, buscando conseguir que o cliente realize a liquidação nas condições que forem combinadas.

Alguns fatos podem interferir na operação de venda. Esses fatos podem ser:

- devolução total das mercadorias contidas na nota fiscal encaminhada ao cliente;
- devolução parcial das mercadorias contidas na nota fiscal encaminhada ao cliente;
- encaminhamento de mercadorias que não conferem com as mercadorias

listadas na nota fiscal encaminhada ao cliente;

- encaminhamento de quantidades que não conferem com a quantidade real.

Os valores referentes às devoluções parciais ou diferenças nas quantidades enviadas devem ser descontados dos valores dos respectivos títulos de cobrança. No caso das devoluções do total das mercadorias, o título deve ser baixado por devolução total da nota fiscal.

Poderá acontecer, após o pagamento, o fato do cliente reivindicar algum valor a seu favor. Se a reivindicação proceder, a empresa dará um crédito ao cliente, através de uma nota de crédito a cliente, onde constará o motivo que justifica essa nota de crédito. Essa nota de crédito só poderá ter validade após a sua aprovação pelo gerente da área financeira. As notas de crédito acima de R\$1.000,00 deverão, obrigatoriamente, ser aprovadas - também - pelo diretor financeiro.

A cobrança da empresa é efetuada através dos bancos conveniados. A gerência financeira determina para quais bancos devem ser encaminhados os títulos de cobrança. O diretor financeiro determina o montante que deve ser distribuído entre os bancos. A gerência financeira executa a distribuição dos títulos, de maneira a cumprir o que foi determinado pela diretoria financeira.

Após a definição dos bancos para os quais irão ser remetidos os títulos de cobrança, são gerados os arquivos de cobrança e transmitidos para os respectivos bancos (de acordo com o padrão estabelecido em cada banco). O banco se encarrega de emitir o boleto de cobrança e proceder ao respectivo recebimento. Em "d+1" da data de pagamento, o banco encaminha o arquivo de "retorno", que servirá para a baixa dos respectivos títulos. Os bancos recebem os valores e os respectivos acréscimos, quando for o caso. Os bancos podem receber com desconto, desde que devidamente orientados pela empresa, através de instruções, desde que estejam dentro das condições contratadas entre as partes.

A diretoria financeira elabora, diariamente, o relatório que apresenta o "aging" das contas a receber, ou seja, a idade dos títulos em atraso. Nesse relatório são apresentadas "faixas" de dias em atraso e o respectivo montante.

A diretoria financeira posiciona, semanalmente, os demais diretores sobre o desempenho da empresa, através de relatório específico.

O dia-a-dia da Área Financeira (Contas a Pagar)

O Sr. Heitor Honesto, também sócio, é responsável pela área financeira. A área financeira subordina o setor de contas a pagar. Esse setor é encarregado de gerir todos os títulos/contas a pagar. Por definição dos diretores, qualquer título deve, obrigatoriamente, ser lastreado por um documento fiscal, ou outro, que possa ser aceito, de acordo com a legislação vigente.

O Sr. Heitor determina que o setor de contas a pagar deve informá-lo semanalmente dos compromissos a serem pagos. Essa informação deve apresentar os valores que devem ser pagos nas quatro semanas seguintes e, a partir desse período, a posição do contas a pagar mês a mês, até o último compromisso cadastrado. Essas informações, disponibilizadas em meio magnético, servirão para a elaboração do fluxo de caixa.

Todos os títulos devem ser pagos através de cheque. Os títulos a pagar são, normalmente, "juntados" por dia de vencimento. Esses títulos compõem o documento denominado Autorização de Pagamento (AP). Dessa forma uma AP pode ser liquidada por um único cheque. Essa AP deve ter a informação da identificação do cheque que liquidou os títulos nela contidos. Atualmente os cheques são emitidos com cópia. Na cópia é registrada a AP que esse cheque está pagando. Os cheques são emitidos para a liquidação dos títulos listados nas APs, desde que essas APs tenham sido aprovadas pelo diretor financeiro.

Alguns documentos, como as Notas de Crédito (NC), devem ser considerados para a concessão de créditos aos clientes. Essas NCs só podem ser consideradas, se aprovadas por, pelo menos, dois diretores da empresa, sendo o diretor financeiro um deles, obrigatoriamente. As NCs são documentos gerados por devoluções, produtos com defeitos de fabricação, etc.

Cada Nota Fiscal recebida pela empresa pode ser liquidada por diversos títulos. Isso acontece, quando se verifica o parcelamento na operação de compra realizada.

Podem ocorrer pagamentos de títulos, sem o correspondente documento fiscal (nota fiscal) apenas nos casos de adiantamentos a fornecedores. Mensalmente, esses adiantamentos devem ser "correspondidos" (associados) aos respectivos documentos, se tiverem sido recebidos pela empresa. A diretoria é informada semanalmente sobre os pagamentos realizados, a título de adiantamento, que se encontram no aguardo do respectivo documento fiscal.

O setor de contas a pagar também gera APs, para pagamento da folha de pagamento e pagamento de tributos/impostos, entre outras contas a pagar, que não têm o suporte de um documento fiscal (nota fiscal). Essas contas a pagar são geradas pelas outras áreas ou sistemas.

É importante ressaltar que todo documento que contém um fato contábil gera os respectivos lançamentos, para que a contabilização seja realizada.

A diretoria financeira elabora, diariamente, um relatório sintético que apresenta as contas a pagar. Esse relatório é apresentado por faixa de valores x faixa de dias a vencer.

A diretoria financeira posiciona, semanalmente, os demais diretores sobre o desempenho da empresa, através de relatório específico.

O dia-a-dia da Área de Produção

O Sr João Oliveira, Diretor da Área de Produção, é o responsável por todo o processo de fabricação de produtos da PLASTBOM. O processo produtivo é realizado sempre a partir de pedidos aprovados. Exceções podem acontecer, mas a priori os produtos são fabricados com base nos pedidos aprovados. A Área de Produção possui um setor de Engenharia de Produto, que tem como responsabilidade o desenvolvimento dos produtos da PLASTBOM e manutenção da documentação de todos os produtos desenvolvidos e em desenvolvimento. Cada produto, obrigatoriamente, possui uma estrutura de produto (algo que equivale a uma fórmula ou uma receita). Essa estrutura de produto contém todos os componentes e respectiva quantidade/percentual, além do percentual (x%) da perda técnica, que se verifica no processo produtivo. A Engenharia desenvolve e mantém as estruturas dos produtos, considerando como unidade o equivalente a 1 kg (1 quilograma). Os produtos fabricados são armazenados no almoxarifado central. De acordo com a programação de entrega, conforme os prazos de entrega confirmados e constantes nos pedidos de venda, os produtos são retirados do almoxarifado central e embalados. Cada produto é embalado de acordo com a informação contida no pedido. Os produtos são preparados para embarque a partir do documento denominado "Autorização de Despacho de Produtos Acabados - ADPA". Esse documento, obrigatoriamente, deve ter a assinatura do diretor da Área de Produção e, na sua ausência, de outro diretor da empresa.

Após a confirmação da preparação da carga para despacho, é informado o

setor de faturamento, que se encarrega de emitir a respectiva nota fiscal. A nota fiscal deve, obrigatoriamente, acompanhar a mercadoria até a entrega ao cliente. O cliente, ao receber a mercadoria, deve - obrigatoriamente - assinar o canhoto da nota fiscal. Esse canhoto deve retornar à empresa e ser entregue ao setor de faturamento. O setor de faturamento, de posse do canhoto, anexa esse canhoto à respectiva "via fixa" da nota fiscal. Esse controle evidencia o recebimento da mercadoria pelo cliente.

Os pedidos podem ser atendidos no seu total ou parcialmente. Os pedidos podem ser cancelados, total ou parcialmente, desde que autorizados pelo diretor comercial e de produção.

A diretoria de produção também responde pelo setor de almoxarifado. Todas as entradas de insumo ou outros produtos devem, obrigatoriamente, ser acompanhadas de documento fiscal (nota fiscal), assim como as saídas só podem ser realizadas com o devido documento de suporte fiscal, a nota fiscal de saída. As regras fiscais para a emissão das notas fiscais de saída são, atualmente, dependentes do Sr. Justus, faturista. Ele sabe todas as operações, mas, mesmo assim, comete alguns erros. Já na entrada, o Sr. Jacinto responde pelo registro de todas as entradas. As entradas de mercadorias devem ser sempre acompanhadas de nota fiscal. Cada entrada de produto contribui para que o preço médio dos produtos seja atualizado.

No final de cada dia, são elaborados relatórios sintéticos e analíticos, demonstrando todas as entradas e saídas que ocorreram no almoxarifado.

As notas fiscais de entrada são encaminhadas ao setor financeiro, bem como as notas fiscais de frete. Caso as faturas, duplicatas ou boletos de cobrança acompanhem a nota fiscal, também são encaminhados ao setor financeiro.

O setor financeiro, ao receber esses documentos, faz o registro para posterior pagamento. Há fornecedores que encaminham os documentos de cobrança (boletos de cobrança) diretamente ao setor financeiro, por intermédio dos serviços dos correios. Esses boletos são, obrigatoriamente, correspondidos com as respectivas notas fiscais.

No momento da emissão de algumas notas fiscais, são emitidos os boletos de cobrança para entrega, juntamente com a nota fiscal, ao cliente no ato do recebimento da mercadoria. O responsável pelo almoxarifado realiza inventários mensais, para verificar se a posição dos estoques corresponde aos registros

realizados. Além do inventário mensal, é realizado o inventário anual, que tem fins contábeis.

Quando se apura alguma divergência, ajustes são realizados. Esses ajustes são aprovados pelo diretor da área e pelo responsável pela contabilidade da empresa.

O dia-a-dia da Área de Compras

O Sr. André Duarte, diretor responsável pela área de compras da PLASTBOM, tem sob sua responsabilidade todas as operações de compras da empresa. Nesse contexto, a sua área é responsável pela homologação de fornecedores, cotações, gestão de pedidos (negociação, fechamento, acompanhamento até a entrega total do pedido). O setor de cadastro é o responsável pela busca de novos fornecedores e homologação dos mesmos. O processo de busca de novos fornecedores (prospecção) refere-se às atividades que objetivam buscar no mercado "novos fornecedores", que possam fornecer produtos de melhor qualidade e preços/condições de pagamento mais vantajosos para a PLASTBOM. O Sr. André sempre se mostra muito rigoroso na seleção de novos fornecedores. Além das boas referências exigidas de cada fornecedor, os produtos fornecidos sempre devem estar em conformidade com as especificações exigidas pela PLASTBOM. A manutenção da homologação dos fornecedores é objeto de avaliação constante. O Sr. André acompanha mensalmente, através dos relatórios preparados pela área responsável pelo recebimento e controle da qualidade dos produtos (que se reporta à área de almoxarifado), que informa a ocorrência de não conformidades encontradas no período. Todos os fornecedores que tiveram alguma ocorrência apontada pelo setor de controle da qualidade são contatados e informados sobre o que foi constatado, para que possam melhorar continuamente e possam entregar seus produtos sempre de acordo com as especificações da PLASTBOM.

O Sr. André exige que o processo de compras seja antecedido, sempre, pelo processo de cotação. A cotação só é dispensada com a sua autorização e de mais um diretor, normalmente o de produção. Para a realização das cotações, é exigido que sejam pesquisados (formalmente) no mínimo 3 (três) fornecedores. Todos os produtos devem ter no mínimo três fornecedores. As exceções, quando existirem por falta de fornecedores no mercado, só podem ser aceitas com autorização do Sr.

André e de outro diretor.

O final do processo de cotação culmina com a escolha de um fornecedor para determinado produto. O produto e o fornecedor escolhido poderão, a partir desse momento, ser objetos da emissão do pedido de compra de produtos por parte da PLASTBOM. Esse pedido pode conter diversos produtos. O Sr. André, assim como toda a diretoria, exige que haja rastreabilidade em todos os processos da empresa. Todas as condições negociadas (preço unitário, condições de pagamento, quantidades/datas de entrega, etc), no processo de cotação, devem estar presentes nos respectivos pedidos. Atualmente, cada pedido pode ter itens que contenham condições diferentes. A empresa só permite a entrada de produtos que tenham, obrigatoriamente, um pedido de compra aprovado. A portaria, ao recepcionar os entregadores dos produtos, pesquisa a existência dos respectivos pedidos. Existindo, são autorizados a se deslocar até o almoxarifado. Caso não haja pedido, a área de compras é imediatamente acionada para dar uma solução ao problema. O almoxarifado anexa a nota fiscal ao respectivo pedido e envia os documentos para a área de compras. O almoxarifado envia a primeira e segunda via das notas fiscais para a contabilidade.

O Sr. André recebe diariamente a posição dos pedidos "novos" encaminhados aos fornecedores, a posição dos pedidos parcialmente atendidos e dos pedidos totalmente atendidos.

Esses pedidos de compra podem ser cancelados a qualquer momento, mesmo já tendo alguns dos seus itens atendidos.

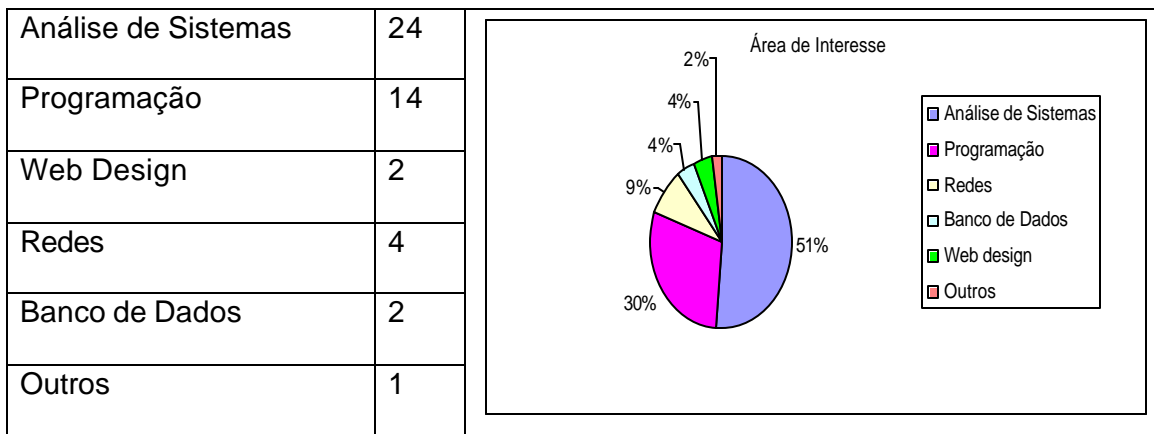
Observa-se que o processo de compra é iniciado, quando a área de produção ou qualquer outra área da empresa requisita um determinado produto. Nessa requisição, a área solicitante informa todas as especificações do produto e as datas em que a área requisitante necessita do produto. Para os produtos utilizados na área de produção, devem ser observados os prazos de entrega e os tempos de fabricação de todos os componentes utilizados para a produção do produto acabado.

Apêndice G

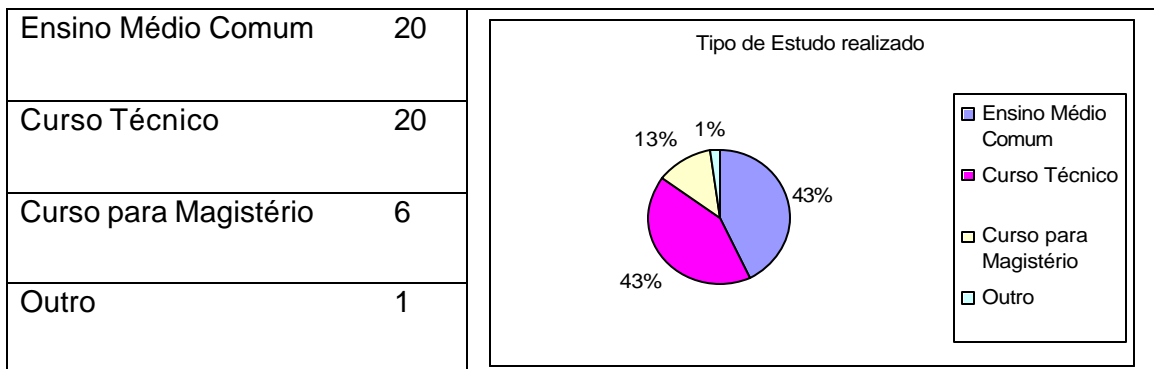
Perfil dos estudantes do estudo de caso PlastBom

Este apêndice contém os resultados consolidados da identificação do perfil do grupo que participou do projeto PlastBom.

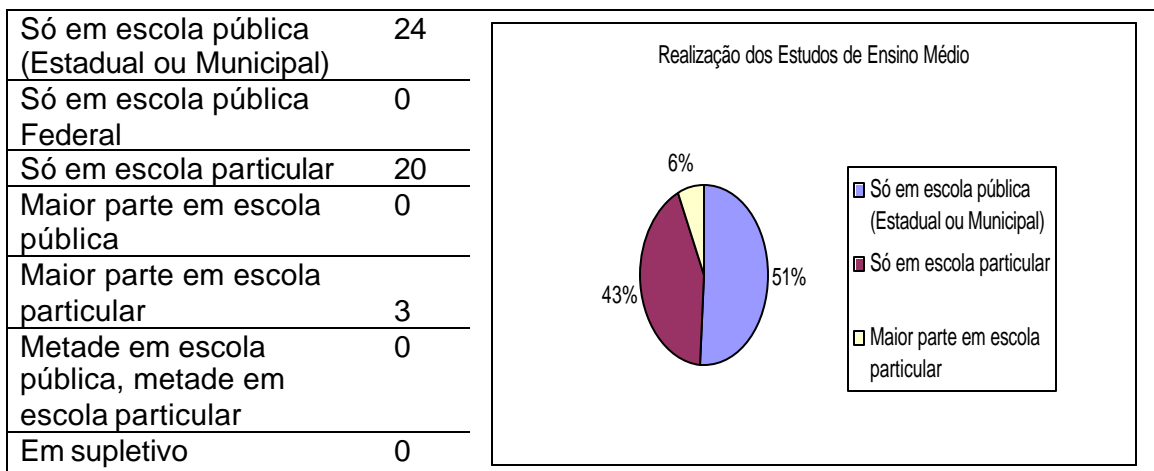
1) Área de Interesse



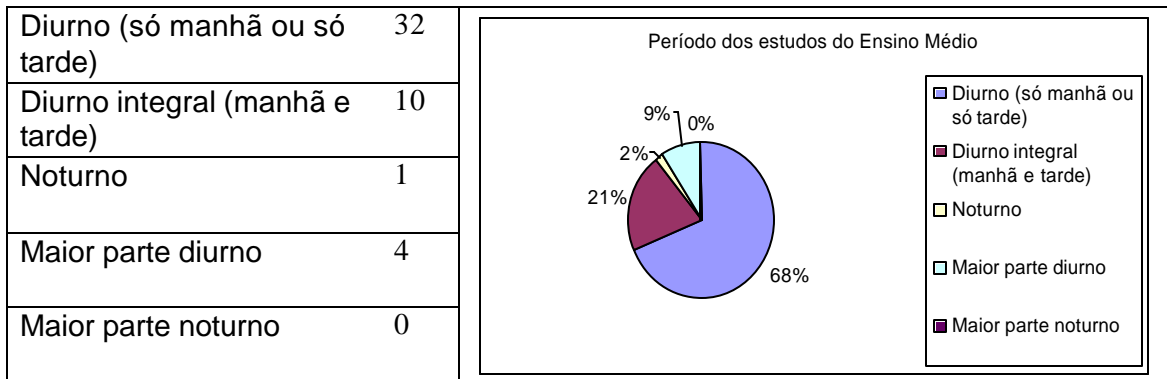
2) Que tipo de estudo você realizou?



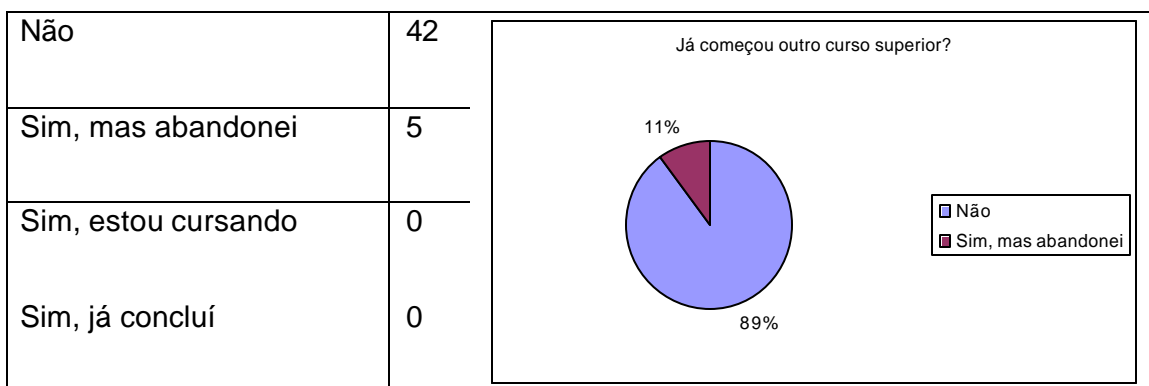
3) Onde você realizou seus estudos de Ensino Médio?



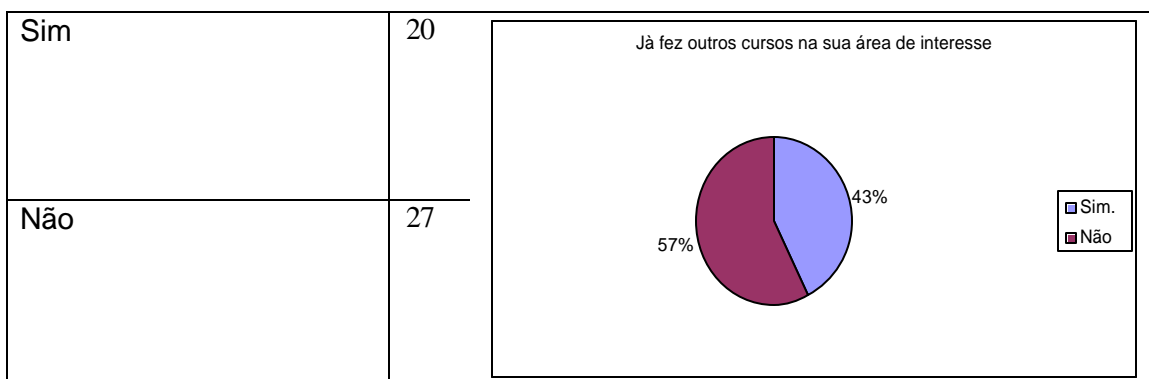
4) Em que período você realizou seus estudos de Ensino Médio?



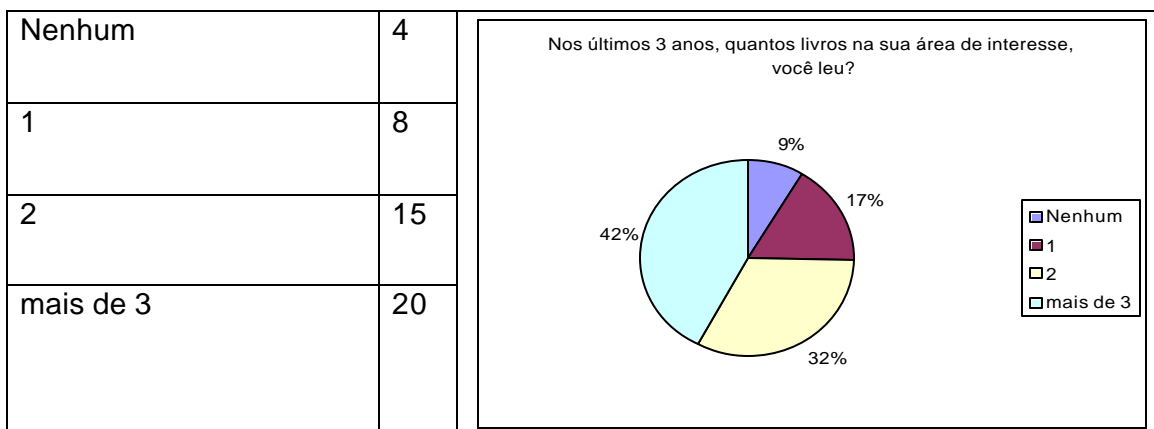
5) Você já começou outro curso superior?



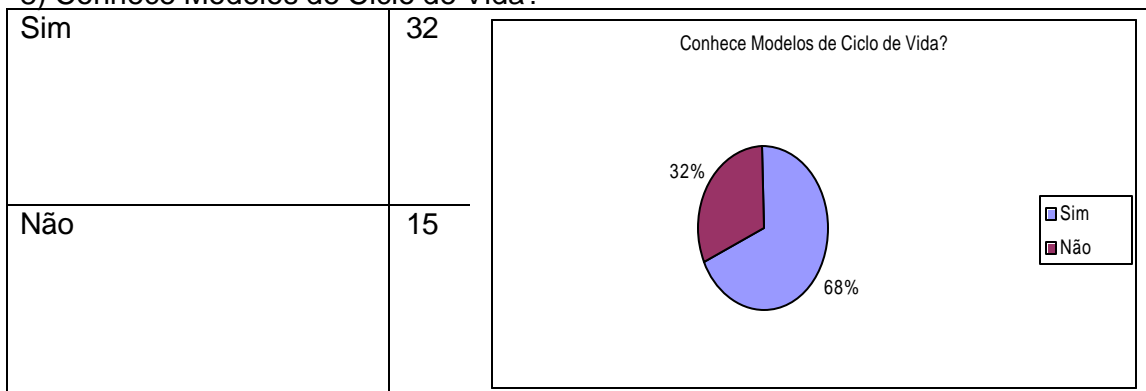
6) Já fez outros cursos na sua área de interesse?



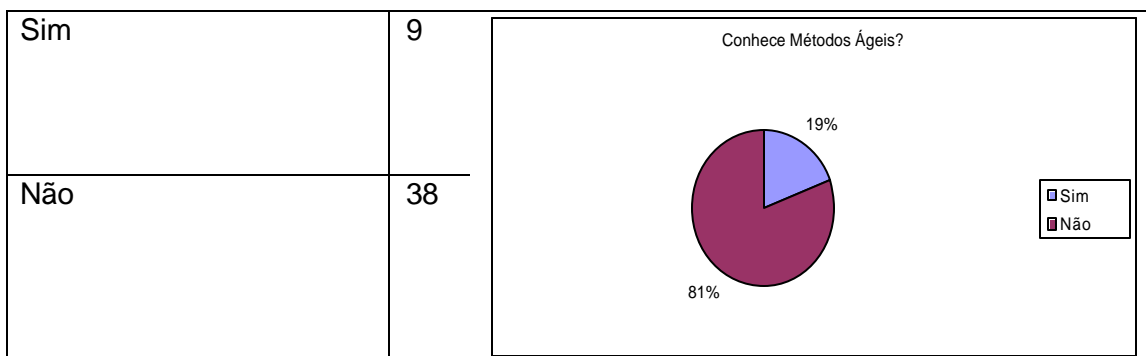
7) Nos últimos 3 anos, quantos livros na sua área de interesse você leu?



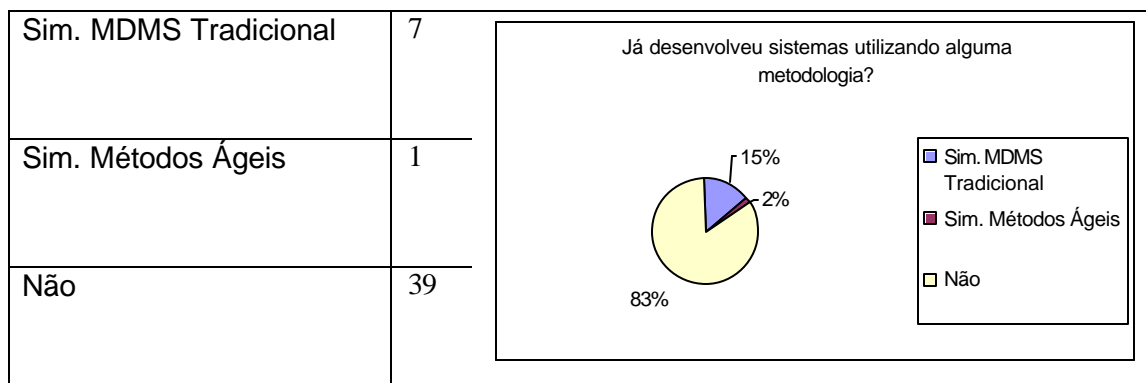
8) Conhece Modelos de Ciclo de Vida?



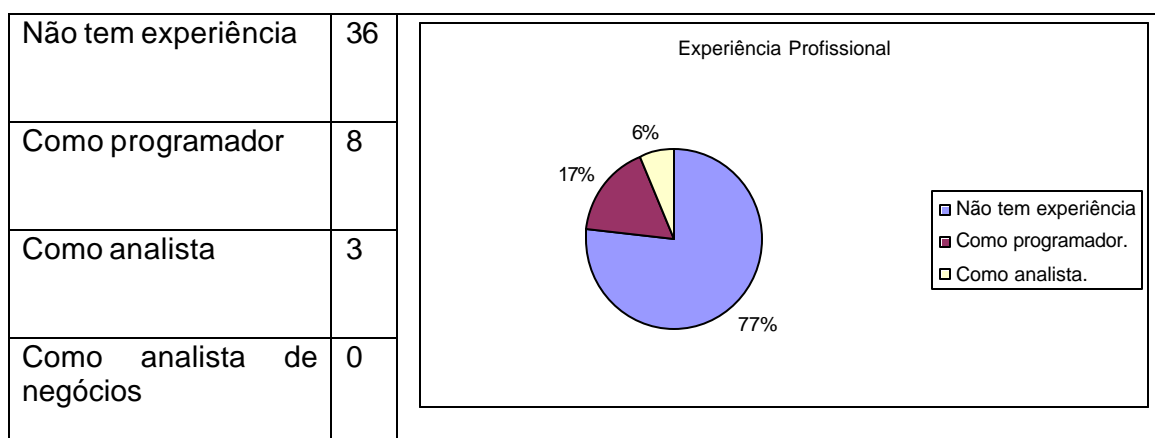
9) Conhece Métodos Ágeis?



10) Já desenvolveu sistemas utilizando alguma metodologia?



11) Em relação à experiência profissional em Desenvolvimento de Sistemas



Apêndice H

Considerações do Questionário de Avaliação I

Este apêndice relaciona as considerações que os participantes do projeto Central Fast Rádio Táxi efetuaram sobre as abordagens.

1. Quais aspectos você considerou negativos, sob a perspectiva da aprendizagem, na abordagem tradicional?

- Muito tempo fazendo documentação.
- Demanda muito tempo até ser produzido um modelo físico para ser validado pelo cliente.
- A abordagem tradicional é bastante burocrática, porém, necessária em um projeto que tenha que ter qualidade; e é mais confiável, assim como o profissional que tenha os conceitos tradicionais.
- Demora no tempo de desenvolvimento.
- Além de trabalhosa é muito lenta, com etapas dependentes umas das outras; se houver um erro, que for descoberto no final ou uma especificação mudada, vai gastar muito tempo e recursos materiais, financeiros, humanos, etc, tornando-se o aprendizado mais lento também devido a todas as fases do processo.
- Geração de muitos documentos.
- Demora-se muito para chegar ao desenvolvimento do sistema (a programação); fica muita teoria e pouca prática, prejudicando na hora de montar o projeto final. Seria bom se, junto com as teorias da abordagem tradicional, já fossem colocados projetos para o desenvolvimento no método ágil.

2. Quais aspectos você considerou positivos, sob a perspectiva da aprendizagem, na abordagem tradicional?

- A seqüência nas atividades está mais clara.
- A possibilidade de estar entendendo o sistema com um todo, antes de estar fazendo a modelagem dos requisitos.
- Melhor qualidade do produto de software, facilidade de manutenção, processos de desenvolvimento mais previsíveis e o projeto mais organizado.
- Maior controle no processo de desenvolvimento, que é melhor para quem está começando a aprender.

- Apesar de lenta e trabalhosa, mostra tudo o que precisa ser feito, como e porque, e a dependência de certas fases em relação às outras.
- O encaminhamento gradual do entendimento do sistema
- Maior tempo para ver cuidadosamente cada passo do processo de desenvolvimento e como, da união desses passos, construímos o projeto final.

3. Quais aspectos você considerou negativos, sob a perspectiva da aprendizagem, na abordagem ágil?

- Como a documentação não tem tanta prioridade como na tradicional, acredito que, para fazer uma manutenção, em um sistema de que você não tem tanto conhecimento fica mais difícil.
- Em certas ocasiões, era necessário estar falando com o cliente muitas vezes. Isso foi um ponto negativo, pois a disponibilidade dele não era igual à nossa necessidade.
- Como pode existir a falta de coordenação entre um processo de desenvolvimento e outro, a possibilidade de falta de documentação, a possibilidade de falta de estruturação do software, as possibilidades de erros são maiores.
- Acredito que seja bom aprender como tudo funciona, tentando e errando para aprender; algumas partes da análise, como elaboração do DER, desenvolvimento, conexão com banco de dados foram automatizadas, e você perde um pouco com isso. Acredito que restrinja um pouco a criatividade.
- Se você não tem noção nenhuma, fica desorientado e faz de qualquer jeito.
- A alta possibilidade de erros não detectados no início do desenvolvimento e a mudança relativamente constante de escopo.
- Se não fosse a base, nós não teríamos conseguido desenvolver o projeto do táxi facilmente, devido ao tempo e à quantidade de informações que o projeto disponibilizava.

4. Quais aspectos você considerou positivos, sob a perspectiva da aprendizagem, na abordagem ágil?

- Mais rápida, é um método mais moderno, facilita o trabalho com o cliente, pois já estamos no desenvolvimento, e o cliente já vai vendo se é isto mesmo que ele quer, enquanto que, na tradicional, acabamos levando o sistema pronto para ser testado.

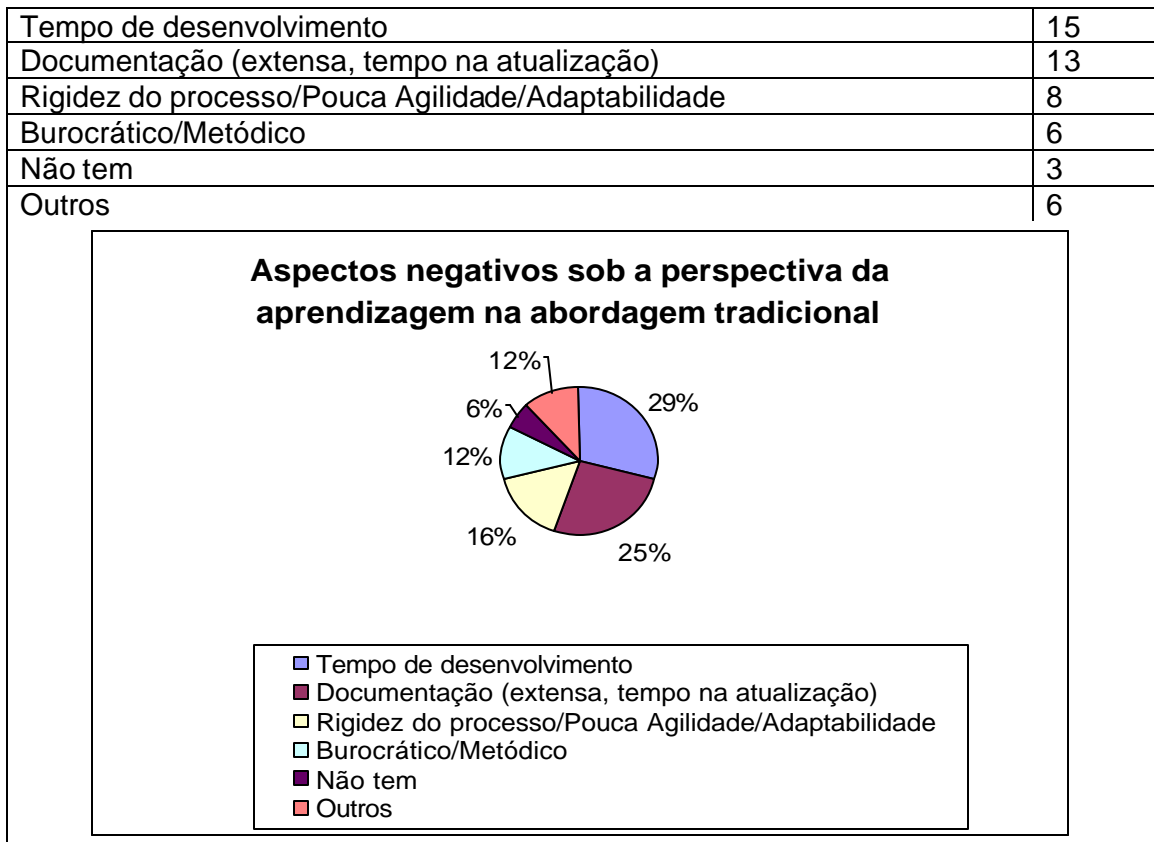
- A aprendizagem foi facilitada, ao passo que os erros de entendimento, no levantamento de dados e requisitos, já eram discutidos no momento da modelagem.
- A flexibilidade no desenvolvimento e o menor tempo de desenvolvimento.
- É rápida, por ser mais automática, tem menos erros de desenvolvimento, e podem-se perceber mais rapidamente os erros de requisitos.
- O método ágil é mais dinâmico, rápido, apesar de incompleto para uma aprendizagem, mas, para a execução, economizam tempo, recursos materiais, financeiros e humanos.
- A facilidade de desenvolvimento e o curto espaço de tempo que separa a definição do que é para ser feito da produção do projeto.
- O desenvolvimento dinâmico sem a necessidade de ficar preso a MER, DD e outros, que demandam tempo para serem feitos, analisados e, se for o caso, o que ocorre muitas vezes, serem corrigidos.

Apêndice I

Resultados Consolidados do Questionário de Avaliação II

Este apêndice contém os resultados consolidados do processo de avaliação adotado, para verificação da adequação das abordagens sob a perspectiva dos alunos.

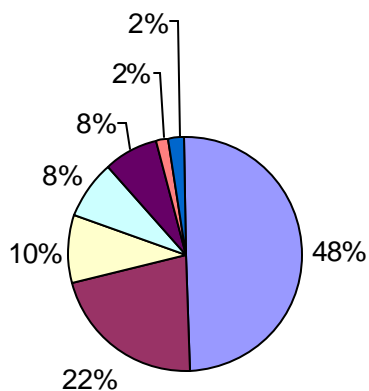
1) Quais aspectos você considerou negativos, sob a perspectiva da aprendizagem, na abordagem tradicional?



2) Quais aspectos você considerou positivos, sob a perspectiva da aprendizagem, na abordagem tradicional?

Processo bem definido, preditivo, seguro, estável	25
Facilidade de aprendizagem/compreensão do processo	11
Confiabilidade	5
Documentação detalhada dando apoio	4
Planejamento/Organização do processo	4
Requisitos estáveis e estabelecidos antecipadamente	1
Padronização e organização para trabalhar em equipe	1

Aspectos positivos sob a perspectiva da aprendizagem na abordagem tradicional

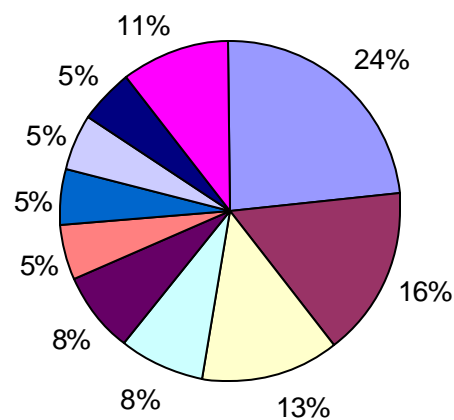


- Processo bem definido, preditivo, seguro, estável
- Facilidade de aprendizagem/compreensão do processo
- Confiabilidade
- Documentação detalhada dando apoio
- Planejamento/Organização do processo
- Requisitos estáveis e estabelecidos antecipadamente
- Padronização e organização para trabalhar em equipe

3) Quais aspectos você considerou negativos, sob a perspectiva da aprendizagem, na abordagem ágil?

Falta de documentação mais detalhada pode prejudicar a implantação ou manutenção	9
Mudanças de requisitos freqüentes	6
Necessidade de comunicação intensa (comprometimento do usuário)	5
Não especifica claramente os processos de desenvolvimento	3
Suscetível a erros (falta de um planejamento, embasamento teórico)	3
Não dá suporte ao desenvolvimento de aplicação complexa	2
Não tem	2
Difícil gerenciamento	2
Possibilidade de diminuir/pular as atividades do ciclo de vida	2
Outros	4

Aspectos negativos sob a perspectiva da aprendizagem na abordagem ágil

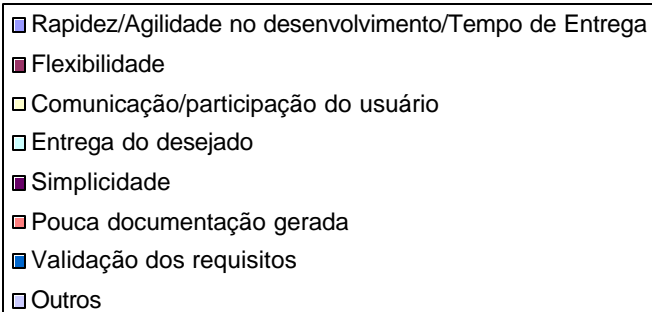
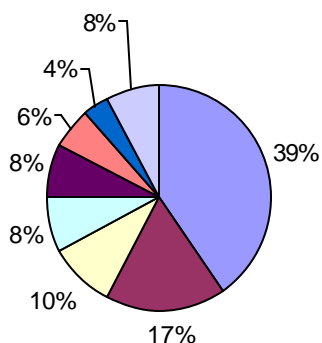


- Falta de documentação mais detalhada pode prejudicar a implantação ou manutenção
- Mudanças de requisitos freqüentes
- Necessidade de comunicação intensa (comprometimento do usuário)
- Não especifica claramente os processos de desenvolvimento
- Suscetível a erros (falta de um planejamento, embasamento teórico)
- Não dá suporte ao desenvolvimento de aplicação complexa
- Não tem
- Difícil gerenciamento
- Possibilidade de diminuir/pular as atividades do ciclo de vida
- Outros

4) Quais aspectos você considerou positivos, sob a perspectiva da aprendizagem, na abordagem ágil?

Rapidez/Agilidade no desenvolvimento/Tempo de Entrega	21
Flexibilidade	9
Comunicação/participação do usuário	5
Entrega do desejado	4
Simplicidade	4
Pouca documentação gerada	3
Validação dos requisitos	2
Outros	4

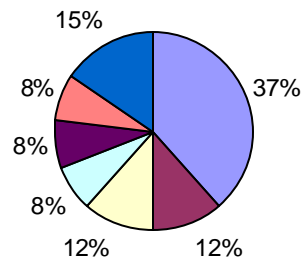
Aspectos positivos sob a perspectiva da aprendizagem na abordagem ágil



5) Liste os pontos negativos constatados a partir do uso dessa ferramenta.

Manutenção difícil (detecção e correção de erros)	10
Consumo de espaço (executável pesado, fontes grandes)	3
Limitações da ferramenta	3
Padronização	2
Conhecimento de uma nova linguagem	2
Código gerado (confuso, extenso)	2
Outros	4

Aspectos negativos em relação ao uso de software gerador de aplicação

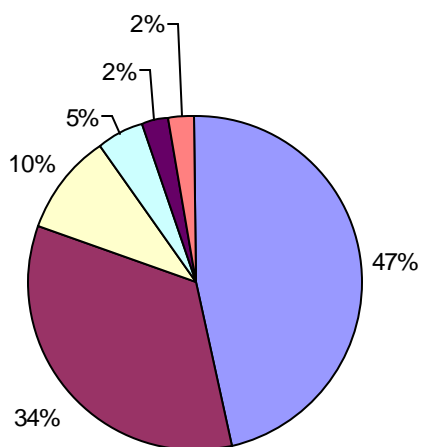


■ Manutenção difícil (detecção e correção de erros)
■ Consumo de espaço (executável pesado, fontes grandes)
■ Limitações da ferramenta
■ Padronização
■ Conhecimento de uma nova linguagem
■ Código gerado (confuso, extenso)
■ Outros

6) Liste os pontos positivos constatados a partir do uso dessa ferramenta.

Rapidez no desenvolvimento (geração de código)	19
Facilidade de desenvolvimento (programação, estrutura das tabelas, telas)	14
Não é necessário o conhecimento efetivo de alguma linguagem de programação	04
Conhecimento da ferramenta	02
Permite focar nas regras de negócio	01
Padronização	01

Aspectos positivos em relação ao uso de software gerador de aplicação

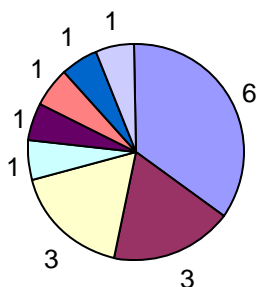


- Rapidez no desenvolvimento (geração de código)
- Facilidade de desenvolvimento (programação, estrutura das tabelas, telas)
- Não é necessário o conhecimento efetivo de alguma linguagem de programação
- Conhecimento da ferramenta
- Permite focar nas regras de negócio
- Padronização

7) Aspectos negativos em relação ao desenvolvimento auto-organizado

Falta de organização	6
Falta de comprometimento, comunicação pode prejudicar o projeto	3
Falta de delegação de tarefas	3
Dispersão de foco	1
Descentralização do grupo	1
Conflitos na realização das tarefas	1
Falta de diálogo entre os participantes	1
Falta de cooperação	1

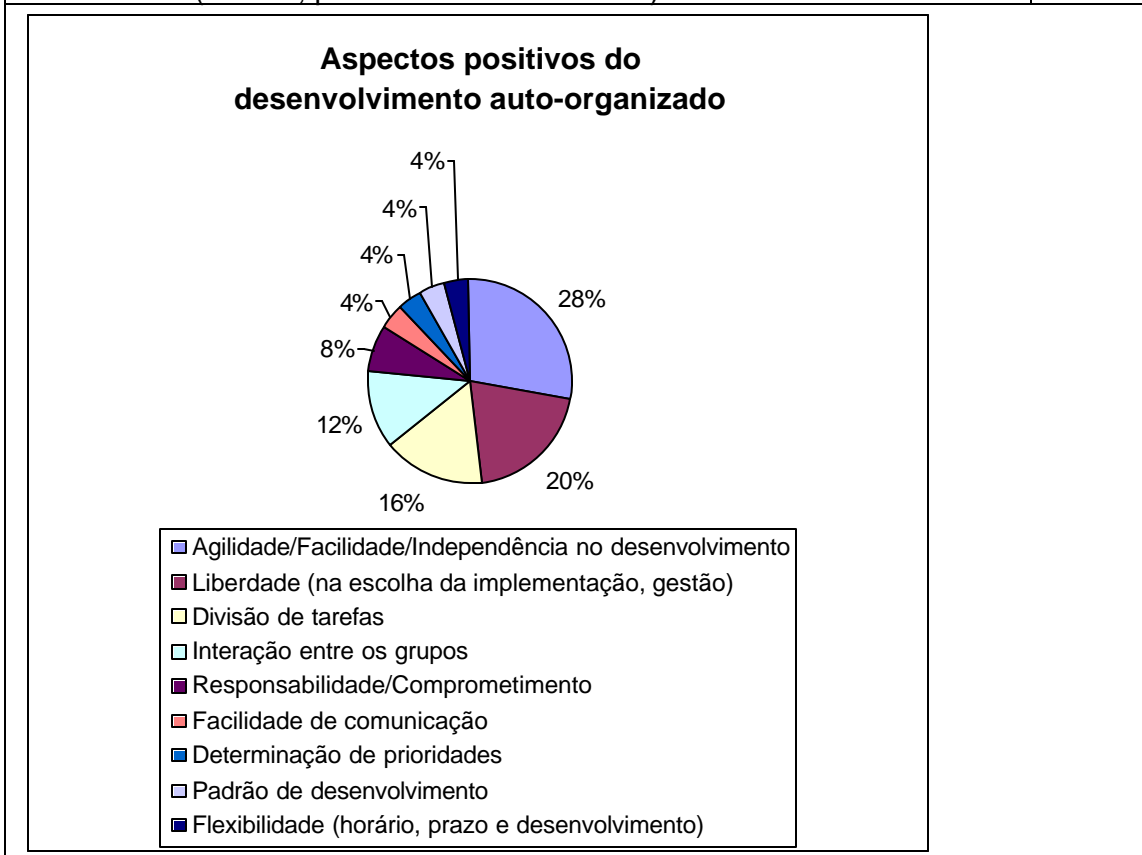
Aspectos negativos do desenvolvimento auto-organizado



- Falta de organização
- Falta de comprometimento, comunicação pode prejudicar o projeto
- Falta de delegação de tarefas
- Dispersão de foco
- Descentralização do grupo
- Conflitos na realização das tarefas
- Falta de diálogo entre os participantes
- Falta de cooperação

8) Aspectos positivos em relação ao desenvolvimento auto-organizado

Agilidade/Facilidade/Independência no desenvolvimento	7
Liberdade (na escolha da implementação, gestão)	5
Divisão de tarefas	4
Interação entre os grupos	3
Responsabilidade/Comprometimento	2
Facilidade de comunicação	1
Determinação de prioridades	1
Padrão de desenvolvimento	1
Flexibilidade (horário, prazo e desenvolvimento)	1



9) Considerações sobre as reuniões semanais.

Importantes para a definição do projeto (escopo, prazos, foco)	11
Melhorar o acompanhamento do projeto	07
Definição de objetivos/tarefas de cada membro	07
Integração entre a equipe	03
Manter a equipe envolvida	01
Não foram significativas no resultado final do projeto	01
Incentivo para a realização do projeto	01
Comprometimento entre as equipes	01
Planejamento das atividades	01
Permite avaliar o desempenho do grupo	01

