WORKSHOP DE AVALIAÇÃO DE DESEMPENHO DA FASETE

Paulo Afonso, 20 de Junho de 2008.
FASETE – Faculdade Sete de Setembro

Avaliação de Desempenho do Impacto de Metodologias Ágeis no Desenvolvimento de Software

Juliana M. C. dos Santos[1], Edson A. da Silva[1], Filipe R. V. da Silva[1]

[1]Faculdade Sete de Setembro - FASETE
Cep: 48.601-180 – Paulo Afonso – BA – Brasil

(julaimaciel@yahoo.com.br, edson.vicious@gmail.com, lio.bassman@hotmail.com)

Resumo: *Este artigo faz uma comparação entre as metodologias tradicionais para desenvolvimento de software e as metodologias ágeis. Em particular é feita a comparação das pesquisas realizadas pelo Chao Report, em relação ao desenvolvimento de softwares na America do Norte mais especificamente nos USA e no Canadá, e a pesquisa a nível Brasil comparando com a pesquisa a Norte Americana. O foco do artigo e o desenvolvimento ágil e a engenharia de software, como eles podem ajudar os alunos da área de TI da Faculdade Sete de Setembro – FASETE a ter um maior índice de qualidade e sucessos em seus software, enfatizando que suas características são ideais para projetos que devem ter um desenvolvimento rápido e que podem ter requisitos alterados constantemente pelos clientes, e possibilitando um sucesso pleno tanto para o desenvolvedor quanto para o cliente, adaptando os alunos ao ambiente de desenvolvimento ágil e conciso não só na programação como na engenharia.*

Palavras-Chave: *Metodologias de Desenvolvimento, Desenvolvimento Ágil, Engenharia de Softwares.*

1. Introdução

Atualmente o setor de Tecnologia da Informação - TI é o que mais tem desenvolvido em comparação com outras áreas, em especial houve acréscimo exponencial na área de desenvolvimento de *software*, nessa área a busca do aperfeiçoamento e da agilidade do *software*, junto à necessidade de se elaborar e implementar *software* que satisfaçam ao máximo as partes interessadas, são cada vez mais requisitadas. No caso, por parte dos clientes, há maior preocupação em relação ao respeito do contrato.

Sabendo que a produção e o desenvolvimento de *software* é uma vertente em crescimento, é importante se ater na qualidade, tendo como base o respeito às métricas do tempo, assim como os recursos e de funcionalidades propostas e o contrato firmado com o cliente dentro do proposto pela partes.

Com base na pesquisa realizada pelo *Chaos Report* em ralação a maturidade e desenvolvimento de software nacionais e internacionais, se reconhece que a qualidade e o desempenho são fatores decisivos de um produto final satisfatório.

Assim para provar a necessidade dos cuidados com a qualidade de software é proposto um estudo do caso com a instituição de ensino FASETE, com fim de avaliar a evolução da maturidade dos softwares em relação ao conhecimento do aluno sobre engenharia de software.

Neste artigo será tratada na seção 2 a qualidade de softwares, analisando como a qualidade e a engenharia de software andam sempre juntas, uma diretamente influenciada pela outra, na seção 2.2 é tratada a qualidade de software no presente, e fala dos problemas que perduram da origem até os dias atuais, explana também sobre a pesquisa realizada pelo *Chaos Reporta* na America do Norte, na seção 2.3. Analisa a pesquisa feita pelo *Chaos Report* no Brasil e fazendo com comparativo com a pesquisa Norte Americana. Na seção 3 são tratados os modelos de desenvolvimento mostrado, na seção 3.1 é tratado o modelo clássico e como ele funciona e focando seus defeitos e problemas, na seção 3.2 falaremos da modelo ágil colocando os pontos mais importantes do modelo ágil e como e o que é o manifesto ágil, 3.3 e coloca a situação atual de desenvolvimento. Na seção 4 falaremos do nosso estudo de caso, o desenvolvimento de softwares no ambiente de TI da FASETE, explanaremos sobre a pesquisa realizada e os índices encontrados na mesma, e por ultimo a seção 5 é feita a conclusão e avaliação individual dos períodos em seus desenvolvimentos e aplicação da engenharia.

2. Qualidade de Software

Na produção de *software* há vários pontos a serem ressaltados de sua origem a Qualidade de Software e um ponto central que se confunde com a da Engenharia de Software, já que no início não havia a definição clara das subáreas da Engenharia de Software. Inclusive ainda hoje, a Qualidade de Software é ubíqua na Engenharia de Software [1], podemos notar a necessidade de se conhecer a historia da Engenharia de

Software para a percepção de onde se origina os principais problemas vivenciados.

A Engenharia de Software, como aceita por muitos, iniciou-se em 1968, quando aproximadamente 50 especialistas em computação de 11 países se reuniram em Garmisch, Alemanha, para discutir os problemas de desenvolvimento de software da época [2]. Foram identificados os seguintes problemas atribuídos ao desenvolvimento de Software:

- Projetos realizados acima do orçamento;
- Projetos finalizados acima do tempo esperado;
- Produtos de *software* de baixa qualidade;
- Produtos de software sem atender aos requisitos do cliente;
- Projetos ingerenciáveis e com código difícil de se manter.

Hoje em dia muitos desses focos de problemas ainda são detectados no desenvolvimento de *Software*, mas o que realmente foi notado ao decorrer dessas décadas foi que com a observação do amadurecimento de *Software* tais pontos de problemas eram reduzidos daí a importância do gerenciamento de amadurecimento de Software, ou seja, a produção do Software esta ligada a muito mais que puramente implementação do mesmo, mas sim a fatores externos a programação tais como o amadurecimento, ou seja, conhecimento adquirido com os demais *software* produzidos, isso caracteriza a evolução obtida na Engenharia de *Software*.

Os modelos aplicados na garantia da qualidade de software focam no processo de desenvolvimento como forma de garantir a qualidade final do produto.Compreender de fato o que é Qualidade Software é essencial para entender como alcançá-la. Ao longo dos anos vários autores e organizações têm tentado definir o significado do termo Qualidade de *Software*. Para Crosby [3] ela pode ser definida como "Conformidade com os requisitos do usuário". *Baldrige National Quality Program* [4] referem-se à qualidade como "Qualidade dirigida ao cliente" e declaram a satisfação do cliente como ponto mais relevante. Segundo a norma ISO 9000, a qualidade é o grau em que um conjunto de características inerentes a um produto, processo ou sistema cumpre os requisitos inicialmente estipulados para estes.

Quando se fala de qualidade de software é indispensável referenciar o estudo realizado pela empresa Norte Americana *Standish Group*, esse estudo tem como objetivo descobrir o real estado dos projetos, fazendo uma analise com um grupo de várias empresas de softwares, avaliando os prazos, os custos e se o projeto foi entregue de acordo com o que foi planejado.

Esse estudo vem a confirmar o que [5] comenta: "a situação atual do desenvolvimento de software encontra-se aquém da ideal". Sendo que os dados obtidos pelo *Chaos Report* são considerados abaixo do esperado.

2.2. Qualidade de Software no presente

A evolução obtida nas décadas após os problemas descritos em 1968 não garantiram estabilidade e segurança com ao decorrer do tempo ao contrario do que se pensa os Softwares desenvolvidos atualmente ainda permanecem nos mesmos problemas do passado.

O *Chaos Report* foi publicado algumas décadas depois a evolução *Standish Group* [6] com resultados de uma pesquisa mundial com mais de 175.000 projetos de software. A pesquisa mostra que somente 16% dos projetos são finalizados com sucesso. E analisado por metas a serem atingidas para se caracterizar que um projeto obteve sucesso metas tais como o cumprimento do prazo,custo e escopo. Os projetos que não cumpriram pelo menos uma dessas cariáveis são considerados "Desafiados" como mostra a Figura 1. Como podemos ver a seguir 53% dos projetos não cumpriram algumas das variáveis supracitadas e 31% dos projetos foram denominados "Cancelados" por serem cancelados por completo não chegando ao fim do projeto.

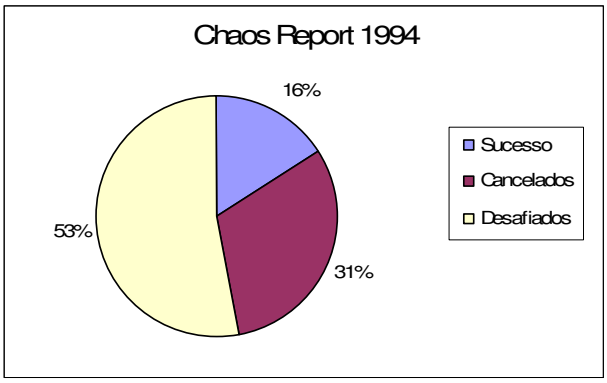


Figura 1 - Resultado do Chaos Report de 1994

A conclusão obtida com os resultados obtidos assusta a comunidade de software, depois de 25 anos vividos após a discussão sobre o problema da Crise de Software que levou a detectarem vários pontos fracos e pertinentes no desenvolvimento de software que resultou em vários aprimoramentos nos métodos, técnicas e teoremas, a nova analise após tais evoluções e que os softwares apresentam os mesmos problemas descobertos na conferencia de Garmish em 1968.

Com essa nova analise notamos o crescimento de software produzidos mas tais relatórios vistos [6] não mostram uma melhoria nos projetos que são entregues fora de prazo, acima do orçamento ou menor de acordo com o que foi proposto, ou seja, os "Desafiados " não obterão alguma melhoria. Podemos notar que ouve uma redução no numero de projetos cancelados e aumento nos que obtiveram sucesso, muitos projetos ainda são "Desafiados", assim mostra a Figura 2 a seguir. Com isso podemos perceber que a evolução não fez com que os problemas relatados antes foram totalmente superados pelo contrario os mesmo problemas persistem.

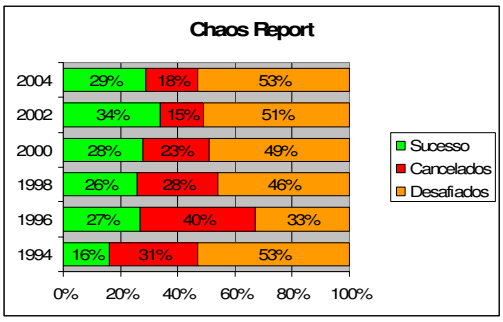


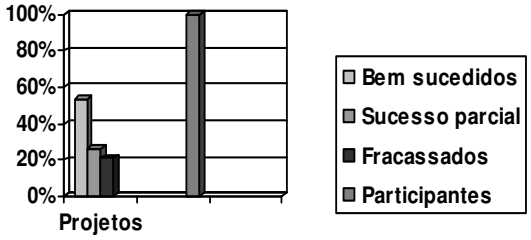
Figura 2 - Resultado do Chaos Report entre 1994 e 2004

Apesar deste diagnóstico negativo dos projetos de software, muitas empresas reportam dados de projetos de sucesso e demonstram os resultados positivos de investimento em iniciativas de melhoria de qualidade. Isso demonstra que se os investimentos em qualidade de *software*, em particular a implantação de modelos de maturidade de software, quando conduzido da maneira correta, podem traz benefícios substanciais para as organizações envolvidas.

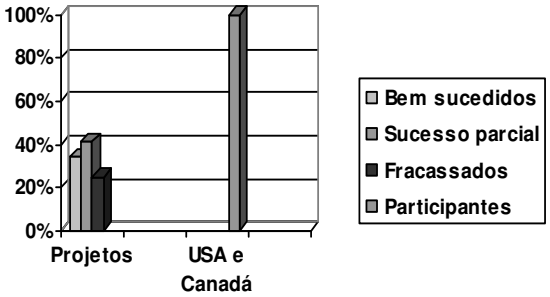
2.3. A Pesquisa de Qualidade de Software no Brasil.

A importância de *Chaos Report* e notável pois através dele podemos ter acesso a informações para avaliar a maturidade e organização do softwares produzidos, ou seja, fazer comparações para elevar o nível de maturidade dos softwares. No Brasil essa pesquisa de maturidade e pode ser encontrada pelo MPCM (*Maturity by Project Category Model*) eles disponibilizam o questionário aplicado para analise da maturidade do software e também da acesso aos demais resultados Brasileiros para que se possa fazer comparativos com a media nacional.

O gráfico a seguir mostra a comparação do resultados dos projetos Brasileiros em 2006 e do *Chaos Report* em 2003:



Fonte: ***Chaos Report*** **Norte Americano**



Fonte: ***Chaos Report*** **Brasil Vs USA e Canadá**

Essa comparação não e feita com fim de especular quem tem mais maturidade na produção de software mas sim mostrar que os índices Brasileiros estão crescendo perante o cenário mundial , daí a importância da analise da maturidade dos softwares. A pesquisa é constituída de um questionário, através do qual é possível avaliar a maturidade de um setor de uma organização com um grau de acerto muito satisfatório. A idéia para o Brasil já vem sendo adotada pela MPCM e que seja feito anualmente no site para que seja possível a analise para que seja traçado um plano de crescimento anualmente. Na produção do software a complexidade do projeto,motivação da equipe,nível de competência técnica da equipe, o cenário de clientes,concorrência, pressão dos negócios fatores externos e o gerenciamento do projeto são fatores que agregam valor ao resultado final do mesmo, ou seja, implicam no sucesso do projeto.

Esse tipo de analise possibilita um padronização de procedimentos para serem utilizados em todos os projetos a principal característica de fracasso e os desvios da meta dos projetos e medidas erradas aplicadas com a analise dos mesmos a otimização na execução dos projetos com base na larga experiência adquirida na pesquisa e eminente. Otimização significa melhoria de resultados, tais como redução de prazos e custos e melhoria da qualidade, ou seja, a melhoria nos processos de gerenciamento de projetos.

Neste Artigo viabilizaremos a importância da analise da maturidade dos projetos desde já na faculdade para termos o conhecimento do nível de produção e com base nas analises nacionais do MPCM em conjunto com o nível mundial disponível pelo *Chaos Report* elevarmos o nível de maturidade dos softwares para que haja maior adequação do profissional de SI no mercado de trabalho.

3. Modelos de Desenvolvimento

As metodologias tradicionais surgiram em um contexto de desenvolvimento de software muito diferente do atual, baseado apenas em um mainframe e terminais burros [7]. Na época, o custo de fazer alterações e correções era muito alto, uma vez que o acesso aos computadores era limitado e não existiam modernas ferramentas de apoio ao desenvolvimento do software, como depuradores e analisadores de código. Por isso

o software era todo planejado e documentado antes de ser implementado. A principal metodologia tradicional e muito utilizada até hoje é o modelo Clássico.

3.1. Modelo Clássico

O modelo Clássico ou Seqüencial segundo [8] foi o primeiro processo publicado de desenvolvimento de software. Desde sua introdução tem sido muito utilizado. É um modelo em que existe uma seqüência a ser seguida de uma etapa a outra. Um exemplo da modelo clássico pode ser visto na figura a seguir:

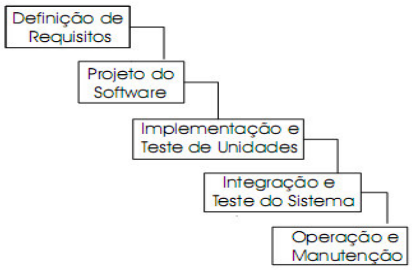


Figura 1: Modelo Clássico

Fonte: Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software

Cada etapa tem associada ao seu término uma documentação padrão que deve ser aprovada para que se inicie a etapa posterior. De uma forma geral fazem parte do modelo Clássico as etapas de definição de requisitos, projeto do software, implementação e teste unitário, integração e teste do sistema, operação e manutenção.

O problema do modelo em Cascata é sua inflexível divisão do projeto em fases distintas, o que dificulta possíveis alterações que são comuns no desenvolvimento de um projeto. [9] desencoraja o uso do modelo Clássico em grandes softwares, estimulando o desenvolvimento incremental como um modelo que apresenta menores riscos e maiores possibilidades de sucesso.

3.2. Desenvolvimento Ágil

O termo Metodologias Ágeis tornou-se popular em 2001 quando dezessete especialistas em processos de desenvolvimento de software representando os métodos Scrum [10], e outros, estabeleceram princípios comuns compartilhados por todos esses métodos. Foi então criada a Aliança Ágil e o estabelecimento do Manifesto Ágil. Os conceitos chave do "Manifesto Ágil" são: Indivíduos e interações ao invés de processos e ferramentas. Software executável ao invés de documentação. Colaboração do cliente ao invés de negociação de contratos. Respostas rápidas a mudanças ao invés de seguir planos.

O Manifesto Ágil não rejeita os processos e ferramentas, a documentação, a negociação de contratos ou o planejamento, mas simplesmente mostra que eles têm importância secundária quando comparado com os indivíduos e interações, com o software, com a colaboração do cliente e as respostas rápidas a mudanças e alterações. Esses conceitos aproximam-se melhor com a forma que pequenas e médias organizações trabalham e respondem a mudanças.

3.3. A Situação Atual

Segundo [11] a situação atual do desenvolvimento de software encontra-se aquém da ideal. Os sistemas são invariavelmente entregues com atraso ou com orçamento estourado, isso quando são efetivamente entregues. Freqüentemente, eles não atendem aos requisitos dos nossos clientes e precisam ser desenvolvidos novamente.

Analisando os dados do *Chaos Report* do *Standish Group* em (1995), [12] apóia a opinião de [5]. Em 1995 ocorreu um dos maiores sucesso de projetos de desenvolvimento de software nos USA. Mais os mesmo relatório que apontavam o sucesso do desenvolvimento, apontavam as falhas ocorridas, em relação as falas ocorridas nos projetos tiveram como causa fundamental a adoção equivocada de metodologias tradicionais de desenvolvimento de software.

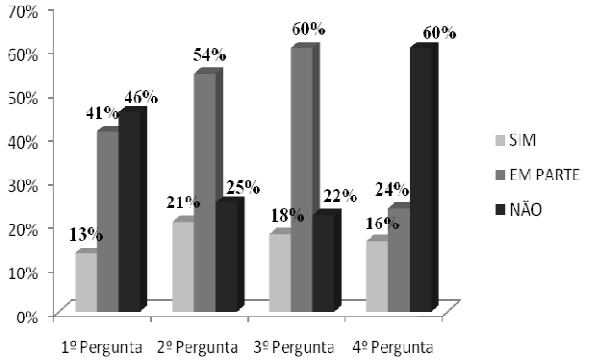
Analisando os dados do *Chaos Report* de 2000 referentes às funcionalidades produzidas pelos projetos, constatou-se que 45% "nunca são utilizados pelos seus usuários e 19% delas raramente são usadas, totalizando 64% de funcionalidade que poderiam deixar de ser produzidas, contrastando com apenas 20% de funcionalidades que são realmente utilizadas de modo freqüente, o que demonstra um inaceitável desperdício de tempo dos projetos de desenvolvimento de software na produção de funcionalidades que os usuários quase ou muito raramente vão utilizar. De um modo geral, os métodos ágeis têm obtido melhores resultados que as metodologias tradicionais.

4. O Desenvolvimento de Softwares no Ambiente de TI da FASETE

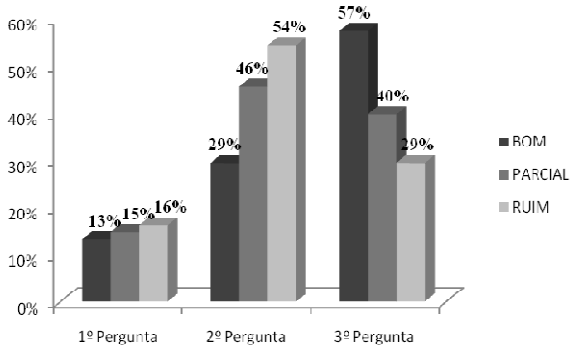
A presente pesquisa teve o intuito de caracterizar a produtividade das atividades relacionadas à implementação e o desenvolvimento de *software* na área de TI na Faculdade Sete de Setembro – FASETE localizada no município de Paulo Afonso – Bahia, entre os alunos das turmas de Sistema de Informação do 3º período ao 7º período, com a finalidade de identificar os déficits dos alunos nos seus desenvolvimentos e no modo de aplicar à engenharia software, para se levantar os dados, foi usada uma pesquisa qualitativa através de um questionário, sendo necessário o contato direto dos pesquisadores com as turmas, e com base nesses dados da início a proposta de desenvolvimento ágil no ambiente FASETE.

O questionário foi composto parcialmente com base nas pesquisas realizadas pelo *Chaos Report* a nível internacional e nacional, na pesquisa realizada no ambiente FASETE, quando

os alunos foram questionados sobre seu projeto nos seguintes dados relacionados: a 1º pergunta foi se cliente ficou satisfeito, 41% dos alunos responderam que em partes isso foi cumprido, 46% dos alunos responderam que não e 13% dos alunos responderam que sim; na 2º pergunta quando questionados se executou o escopo e seu software esta sendo utilizado com previsto 54% dos alunos responderam que em partes isso foi cumprido, 25% responderam que não conseguiram atender o proposto e 21% responderam que sim; a 3º pergunta foi se o software atendendo as especificações qualitativas e de desempenho 60% dos alunos responderam que em partes isso foi atendido, 22% responderam que não conseguiram atender e 18% responderam que sim; na 4º pergunta foi abordado se o software atendeu as restrições e o prazo: 24% do alunos responderam que em partes isso foi cumprido, 60% dos alunos responderam que não conseguiram atender os aspectos colocados e apenas 16% dos alunos responderam que sim o projeto seguiu e atendeu corretamente os aspectos relacionados.

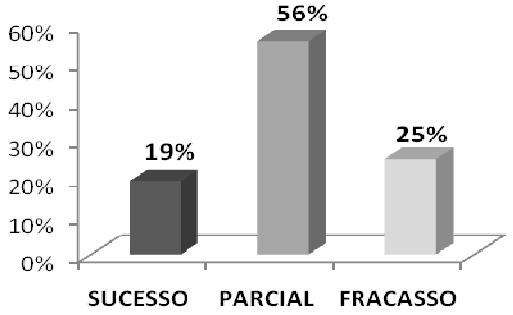


Analisando esse gráfico constatamos que, quando se trata do projeto em todos os aspectos tratados, o índice de alunos que conseguem atender positivamente em todas as questões é muito baixo, em relação aos que conseguem parcialmente, e os que não conseguem. Assim como na foi tratado na pesquisa realizada pelo *Chaos Report analisa nas seções anteriores*. Outros aspectos foram analisados como quando foram perguntados sobre a avaliação da sua equipe no seguinte pontos: a 1º pergunta tava a motivação da equipe 15% dos alunos responderam que parcialmente quanto a motivação, 16 % responderam que sua equipe era ruim em se tratando da motivação e 13 % responderam que sim suas equipes eram motivadas; já na 2º pergunta o questionamento era o nível de competência técnica e a complexidade dos projetos em ralação a isso 46% dos alunos responderam parcialmente aos pontos analisados, 54% responderam que sua equipe era ruim em se tratando das questão colocada e 29% responderam que sim atendiam plenamente a nível de competência e complexidade; o nível de maturidade em gerenciamento tivemos que 40% alunos responderam que parcialmente sua equipe correspondia com maturidade ao gerenciamento, 29% responderam que sua equipe era ruim se tratando em maturidade e gerenciamento e 57% responderam que sim sua equipes atendiam corretamente os ponto analisados.



Ao analisando esse gráfico, notasse que quando se trata de motivação, complexidade e participação integral de equipe o índice não é satisfatório, mais quando se trata dos aspectos maturidade e gerenciamento um nível favorável foi encontrado, mostrando que não importa só a maturidade e o gerenciamento para que a equipe participe integralmente e competentemente no projeto.

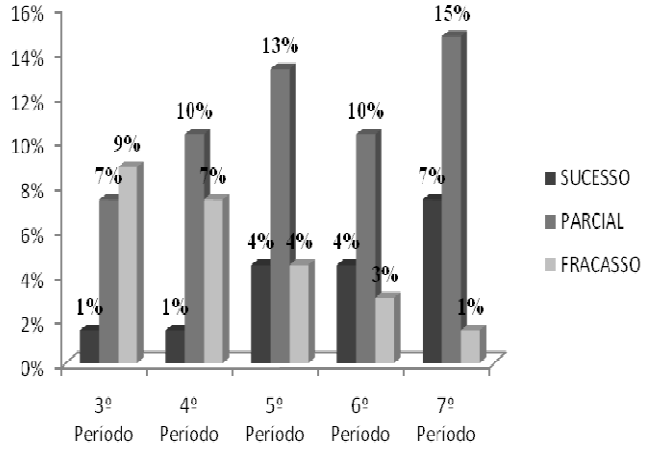
Por fim e não menos importante o ponto mais crucial de toda a pesquisa foi perguntado aos alunos como eles avaliavam o seu projeto final: 56% dos alunos responderam que os seus projetos eram parcialmente bem sucedidos, já 25% dos alunos responderam que seus projetos fracassaram apenas 19% dos alunos responderam que seus projetos foram bem sucedidos, um sucesso pleno, correspondendo integralmente ao esperado e proposto.



Com base no gráfico do projeto final notasse, que uma boa parte dos alunos entrevistados conseguiram fazer com que seus projeto fossem parcialmente bem sucedidos, mais o nível de projetos fracassados ainda é maior que os bem sucedidos. De uma forma geral, podem-se observar os métodos tradicionais de desenvolvimento de *software* e a falta de empenho na utilização da engenharia do *software* são curiosamente alguns dos fatores somados a outros aspectos não menos importantes, que podem fazer com que no decorrer do projeto até o projeto final tive-se um resultado de sucesso total. Como mostra os gráficos analisados a maioria dos projetos bem sucedidos seguiu corretamente todos os pontos analisados tanto de maturidade e agilidade quanto de engenharia, durante o desenvolvimento e o projeto final.

5. Conclusão

Diante das questões analisadas, podemos observar mais atentamente com as respostas dadas. Na aplicação do questionário houve uma grande dificuldade em relação à engenharia de *software* dos alunos do 3º período, por não se preocuparem com a engenharia e apenas com a programação, mas no decorrer dos períodos era crescente o nível de conhecimento e preocupação com a engenharia, e a aplicação da engenharia no projeto sanava as dificuldades encontradas nos períodos anteriores, aumentando assim o percentual de projetos bem sucedidos.



Podemos analisar no gráfico que o percentual de alunos com projetos bem sucedidos no 7º período é maior que o percentual dos períodos anteriores, e em principal em relação ao 3º período. Então podemos definir a engenharia de softwares e tão importante quando a programação que o sucesso do projeto. Mediante todas as respostas, tiveram alunos que não respoderam as questões relacionadas à engenharia chagando a um percentual de 6% dos alunos, que não respondeu algumas das questões, por que não entediam ou não se aplicava a sua realidade ou seus métodos de desenvolvimento.

Com as respostas dadas pelos alunos e as não respondidas, cabe-nos refletir o modo como está sendo trabalhada a engenharia na área de TI. Notamos que o aluno tende a evoluir no contexto da produção de software a medida que aprendem técnicas de desenvolvimento e paradigmas que lhe dão mais agilidade no desenvolvimento. Essas reflexões são necessárias para um melhor aprimoramento no ensino e na prática de desenvolvimento com a aplicação da engenharia, e em um contexto geral da área de TI. Para que assim, todos os desenvolvedores possam aplicar a engenharia com mais flexibilidade e freqüência em seus projetos.

Para que essa pesquisa seja mais efetiva seria necessário que ela abrangesse todo o estado da Bahia ou até mesmo toda a região nordeste do país, como não tivemos recurso e nem tempo suficiente para tal, essa pesquisa se baseará nas turmas de Sistema de Informação da FASETE.

Bibliografia

[1] Swebok, **Software Engineering Body of Knowledge ,** 2004.Qualidade de Software, **acessivel em:**http://209.85.215.104/search?q=cache:qVz_eosgBkYJ:200.17.137.110:8080/in953/lectures/slides/porquequalidadedesoftware.doc+Software+%C3%A9+ub%C3%ADqua+na+Engenharia+de+Software+(SWEBOK,+2004)&hl=ptBR&ct=clnk&cd=1&gl=br&client=firefox-a

[2] Naur & Randell, 1968. P. Naur and B. Randell, (Eds.).Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs Division, NATO (1969) 231pp.

[3] Crosby,Philip B. Quality is free: The Art of Making Quality Certain. New York: MCGraw-Hill,1979.

[4] Julio Battisti**, tutorial de TCP/IP - Introdução ao TCP /IP** , ultima atualização 13/09/2006 , http://www.juliobattisti.com.br/artigos/windows/tcpip_p11.asp.

[5] Ambler, Scott. **Modelagem Ágil**, Bookman, 2002.

[6] Standish Group, **CHAOS REPORT**, 586 Olde Kings Highway. Dennis, MA 02638, USA, 1995.

[7] Royce, W.W. **Managing the develop-ment of large software systems: concepts and techniques**. Proc. IEEE Westcon, Los Angeles – CA, 2007.

[8] Pressman, R. Engenharia de Software. McGraw-Hill,2001.

[9] Gilb, T. Principles of Software Engineering Management. Addison-Wesley,1988.

[10] Schwaber, K. and Beedle, M., Agile Software Development with Scrum, NJ, Prentice-Hall,2002.

[11] Ambler, Scott. **Modelagem Ágil**, Bookman, 2002.

[12] Soares, J. F. Qualidade e eqüidade na educação básica brasileira:a evidência do SAEB 2001. *Archivos Analíticos de Políticas Educativas*, v.12, n. 38, 2004. Disponível em:http://epaa.asu.edu/epaa/v12n38. Acesso em: abr. 2007.