

Gestão de Projeto com Scrum: Um Estudo de Caso

Antonio José F. Andrade¹, Jany C. de Oliveira², Paulo Alberto M. Barbosa³,
Francisca Raquel de V. Silveira¹

¹Informática – Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
CEP 62.800-000 – Aracati – CE – Brasil

²Ciências da Computação – Universidade Estadual Vale do Acaraú (UVA)
Sobral – CE – Brasil

³Ciências da Computação – Universidade Estadual do Ceará (UECE)
Fortaleza – CE – Brasil

{a_jot@hotmail.com, dye.nny, albertobmap}@hotmail.com,
raquel_silveira@ifce.edu.br

Abstract. *A new approach to software development has aroused great interest among the organizations around the world. These are agile methodologies that have become popular because it has the characteristic of a simplified approach focused on finding a way of work for team member to produce the software in a flexible way and in a constantly changing environment. This paper aims to present the challenges posed to the project manager and others members and the indicators of software project in the use of the Scrum agile methodology.*

Resumo. *Uma nova abordagem para desenvolvimento de software tem despertado grande interesse entre as organizações de todo o mundo. Trata-se das metodologias ágeis que vêm se tornando bastante popular por apresentar como característica uma abordagem simplificada com foco em encontrar uma forma de trabalho para os membros da equipe produzirem o software de forma flexível e em um ambiente em constantes mudanças. Este trabalho tem como objetivo apresentar os desafios proporcionados ao gerente de projeto e aos demais integrantes e os indicadores de desempenho de um projeto de software na utilização da metodologia ágil Scrum.*

1. Introdução

No atual ambiente de desenvolvimento de software, os requisitos estão sujeitos a frequentes alterações durante o ciclo de desenvolvimento do produto para atender as alterações da demanda [Rising & Janoff 2000]. Este fato torna o desenvolvimento de software um desafio, principalmente para as empresas tendo em vista seus recursos restritos [Carvalho & Mello 2009].

Em virtude desse cenário, alguns fatores críticos ou exigências para o sucesso se destacam: a agilidade, a capacidade de adaptação, o poder de inovar de forma rápida e eficiente, e o potencial de aprimoramento contínuo sob grandes restrições de recursos. Em resposta a essas exigências, fortalecem-se os sistemas de gerenciamento de projetos, como forma de gerir os empreendimentos temporários, únicos e multifuncionais, que

caracterizam o processo de implementação de estratégias, inovação, adaptação e aprimoramento [Gerenciamento 2008].

As metodologias ágeis de desenvolvimento de software se propõem a construir software com maior produtividade e, sobretudo, com qualidade garantida. Para isso elas encaram os projetos sobre um novo paradigma e defendem a adoção de uma série de princípios e práticas [Tavares 2008].

Nesse contexto, destaca-se o Scrum, uma abordagem enxuta de desenvolvimento de produtos. Ele é um processo ágil de desenvolvimento de produto ou administração de qualquer trabalho iterativo e incremental e poder ser aplicado ao desenvolvimento de produtos de maneira geral [Rising & Janoff 2000; Abrahamsson et al 2002]. Apesar de ser uma abordagem nova, a utilização do Scrum tem aumentado bastante nos últimos anos, impulsionados pelas recentes pesquisas que mostram que seu uso aumenta a satisfação dos clientes e diminui o atraso em projetos em relação aos métodos tradicionais [Mann & Maurer 2005].

Este trabalho tem como objetivo realizar um estudo de caso em uma empresa de desenvolvimento de software na implantação do SCRUM como metodologia de gerenciamento de projeto e apresentar os principais desafios encontrados pelo gerente de projetos e pelos integrantes da equipe, assim como os indicadores de produtividade do projeto de software.

O presente trabalho está organizado em cinco seções: a seção 2 apresenta a fundamentação teórica a respeito da gerência de projetos, assim como a respeito das metodologias ágeis no contexto do Scrum; na seção 3 é apresentado o preparo da equipe para a implantação do Scrum e as principais tarefas que foram executadas no primeiro projeto de software da empresa usando Scrum; na seção 4 consta o desempenho dos integrantes e os indicadores de produtividade apresentados pelo projeto; na seção 5 é explanado a respeito das considerações finais referentes a este trabalho.

2. Gerência de projeto

O fracasso de muitos grandes projetos de software, na década de 60 e no início da década de 70 foi a primeira indicação das dificuldades de gerenciamento de software. O software era entregue com atraso, não era confiável, custava várias vezes mais do que previam as estimativas originais e, muitas vezes, exibia características precárias de desempenho. Esses projetos não fracassaram porque os gerentes ou os programadores eram incompetentes. Ao contrário, esses projetos grandes e desafiadores atraíram pessoas de capacitação acima da média. A falha residia na abordagem de gerenciamento utilizada. Técnicas de gerenciamento provenientes de outras disciplinas da engenharia eram aplicadas e mostravam-se ineficazes para o desenvolvimento de software [Sommerville 2004].

O desenvolvimento de produtos é uma atividade bastante complexa, principalmente para as empresas que possuem grandes limitações de recursos [Carvalho & Mello 2009]. Segundo Mundin et al (2002), o desenvolvimento de produtos se relaciona com praticamente todas as demais funções de uma empresa. Isso porque, para desenvolver produtos são necessárias informações e habilidades de membros de todas as áreas funcionais, caracterizando-se como uma atividade, em princípio, multidisciplinar. Além disso, trata-se de uma atividade com uma característica *ad-hoc*, em que cada

projeto de desenvolvimento pode apresentar características específicas e um histórico particular.

A necessidade de gerenciamento é uma importante distinção entre o desenvolvimento profissional de software e a programação em nível amador [Sommerville 2004]. O gerenciamento de projetos é a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos [PMBOK, 2009].

Necessitamos do gerenciamento de projetos de software porque a engenharia de software profissional está sempre sujeita a restrições de orçamento e prazo. Essas restrições são estabelecidas pela organização que desenvolve o software. O trabalho do gerente de projeto de software é garantir que o projeto cumpra essas restrições e entregar um produto de software que contribua para as metas da empresa [Sommerville 2004]. O gerente de projetos é a pessoa designada pela organização executora para atingir os objetivos do projeto [PMBOK, 2009].

2.1. Metodologias Ágeis

Ao longo dos anos, várias metodologias de desenvolvimento de produtos foram apresentadas. Entre elas, existem as chamadas metodologias ágeis ou leves [Ambler 2002].

Metodologia ágil teve seu surgimento durante os anos 90 como reação aos métodos de desenvolvimento burocráticos e lentos da época. Inicialmente, as metodologias ágeis, eram denominadas como métodos de desenvolvimento “leves”. O termo metodologias ágeis tornou-se popular em 2001, quando 17 especialistas em processo de desenvolvimento de software estabeleceram princípios comuns compartilhados por todos esses métodos. O resultado foi a criação do “*The Agile Manifesto*” no qual abordaram o nome Metodologia Ágil. Este manifesto foi criado com o intuito de fazer a união entre diferentes metodologias ágeis [Agile 2011].

As metodologias ágeis tratam-se de metodologias de desenvolvimento adaptativas e flexíveis, e que são indicadas para cenários onde a mudança de requisitos é constante e os resultados precisam ser entregues ao cliente em curtos espaços de tempo. A proposta destas metodologias é dividir o desenvolvimento em ciclos curtos, ou iterações, de apenas algumas semanas, de modo que, ao final de cada ciclo, o cliente (interno ou externo) receba uma versão que agregue valor ao seu negócio [Dantas 2003].

Enquanto as metodologias tradicionais de desenvolvimento mantêm o foco na geração de documentação sobre o projeto e no cumprimento rígido de processos, a proposta ágil é concentrar as atenções no desenvolvimento em si e nas relações entre os participantes [Mundin et al 2002].

Agilidade quer dizer a habilidade de criar e responder a mudanças, buscando a obtenção de lucro em um ambiente de negócio turbulento; ou ainda, a capacidade de balancear a flexibilidade e a estabilidade. A ausência de estrutura ou estabilidade pode levar ao caos, mas que a estrutura em demasia gera rigidez [Highsmith 2004].

A agilidade pode ser aplicada a qualquer processo de software. Entretanto, para conseguir isso, é essencial que o processo seja projetado de modo que permita à equipe de projeto adaptar tarefas e aperfeiçoá-las, conduzir o planejamento para que se entenda

a fluidez de uma abordagem de desenvolvimento ágil, eliminar tudo menos os produtos de trabalho mais essenciais e mantê-los simples, e enfatizar uma estratégia de entrega incremental que forneça o software funcionando ao cliente o mais rápido possível para o tipo de produto e ambiente operacional [Pressman 2006].

O manifesto ágil não rejeita processos e ferramentas, documentação, negociação de contratos nem planejamento, mas simplesmente mostra que estes têm importância secundária quando comparado com os indivíduos, com o software executável, com a colaboração dos clientes e as respostas rápidas às mudanças. Esses conceitos aproximam-se melhor da forma como as pequenas e médias empresas trabalham e respondem às mudanças [Koscianski 2007].

Segundo Highsmith & Cockburn (2001), o que há de novo sobre os métodos ágeis não são as práticas que eles usam, mas o reconhecimento de pessoas como os principais responsáveis pelo sucesso do projeto, juntamente com um intenso foco na eficácia e na capacidade de gerenciar. Isto produz uma nova combinação de valores e princípios que definem uma visão do mundo ágil.

Podemos citar várias metodologias ágeis: Scrum, Agile Modeling, AUP (Agile Unified Process), Agile Data Method, DSDM (Dynamic Systems Development Method), EssUP (Essential Unified Process), XP (Extreme programming), FDD (Feature Driven Development), Getting Real e OpenUP (Open Unified Process). Todos eles têm seu nicho de aplicação e particularidades, mas todos são interativos e incrementais, seguindo os princípios ágeis [Larman & Basili 2003]. Alguns deles podem ser usados simultaneamente, como na recorrente parceria entre o XP e o Scrum [Vriens 2003].

2.1.1. Scrum

O Scrum (nome derivado de uma atividade que ocorre durante um jogo de *rugby*) é um modelo ágil de processo que foi desenvolvido por Jeff Sutherland e por sua equipe no início da década de 1990 [Pressman 2006]. Originalmente, o Scrum foi desenvolvido para ser implementado em equipes de desenvolvimento de produtos de software. Porém, pode ser utilizado por qualquer empresa que necessite implementar processos de gerenciamento de projetos, tais como agências de publicidade, projetos de arquitetura, bancos etc [Silva et al 2010].

O Scrum baseia-se em seis características: flexibilidade dos resultados; flexibilidade dos prazos; times pequenos; revisões frequentes; colaboração; orientação a objetos [Schwaber 1995]. Este método não requer ou fornece qualquer técnica específica para a fase de desenvolvimento, apenas estabelece conjuntos de regras e práticas gerenciais que devem ser adotadas para o sucesso de um projeto [Carvalho & Mello 2009].

O Scrum não é um processo previsível, ele não define o que fazer em toda circunstância. Ele é usado em trabalhos complexos nos quais não é possível prever tudo o que irá ocorrer e oferece um framework e um conjunto de práticas que torna tudo visível. Isso permite aos praticantes do Scrum saber exatamente o que está acontecendo ao longo do projeto e fazer os devidos ajustes para manter o projeto se movendo ao longo do tempo visando alcançar os seus objetivos [Schwaber 2004].

Segundo Schwaber & Beedle (2007), o Scrum é um *framework* dentro do qual pode-se empregar diversos processos e técnicas para desenvolver produtos complexos.

Pereira et al (2007) que relatam que o uso da agilidade traz vantagens como:

- Cria um ambiente propício para definição de mudanças de requisitos e inovação durante o ciclo de desenvolvimento do produto, assim como mais colaborativo e produtivo entre desenvolvedores e cliente, resultando em entregas mais rápidas de produto, melhor adaptados à realidade do cliente e com a qualidade esperada.
- Facilita o gerenciamento do projeto, uma vez que existem maior integração e comprometimento da equipe do projeto, que conseqüentemente se sente mais motivada: a moral da equipe é elevada.
- Reforça o planejamento constante do projeto, o que minimiza os riscos, considerando que o planejamento é mais importante do que o plano. Não se deve parar de planejar até que se tenha encontrado a satisfação do cliente com a entrega do produto.
- Valoriza a satisfação do cliente em primeiro lugar.

A Figura ilustra o ciclo de desenvolvimento do Scrum de forma simplificada.

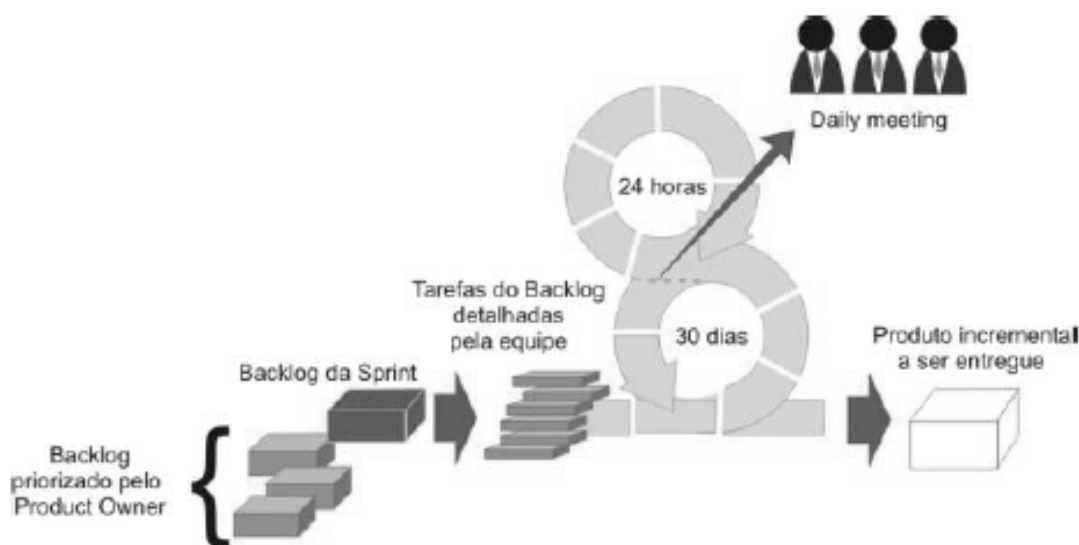


Figura 1. Ciclo do Scrum [Pereira et al 2007].

O ciclo do Scrum tem o seu progresso baseado em um série de iterações bem definidas, cada uma com duração de 2 a 4 semanas, chamadas *Sprints*. Antes de cada *Sprint*, realiza-se uma Reunião de planejamento (*Sprint Planning Meeting*) onde o time (equipe) de desenvolvedores tem contato com o cliente (*Product Owner*) para priorizar o trabalho que precisa ser feito, selecionar e estimar as tarefas que o time pode realizar dentro da *Sprint* [Pereira et al 2007].

A próxima fase é a Execução da *Sprint*. Durante a execução da *Sprint*, o time controla o andamento do desenvolvimento realizando Reuniões Diárias Rápidas (*Daily Meeting*), não mais que 15 minutos de duração, e observando o seu progresso usando um gráfico chamado *Sprint Burndown*. [Pereira et al 2007].

Ao final de cada Sprint, é feita uma revisão no produto entregue para verificar se tudo realmente foi implementado com a realização uma Reunião de Revisão (*Sprint Review*), onde o time demonstra o produto gerado na *Sprint* e valida se o objetivo foi atingido. Logo em seguida, realiza-se a Reunião de Retrospectiva (*Sprint Retrospective*), uma reunião de lições aprendidas, com o objetivo de melhorar o processo/time e/ou produto para a próxima *Sprint* [Pereira et al 2007].

Há seis papéis identificáveis no Scrum que têm tarefas e finalidades diferentes durante o processo e suas práticas: *Scrum Master*, *Product Owner*, *Scrum Team*, o cliente, o usuário e o gerente. A seguir estes papéis são apresentados de acordo com as definições de Schwaber & Beedle (2007):

- *Scrum Master*: É um novo gerente introduzido ao Scrum. É responsável por garantir que o projeto é realizado de acordo com as práticas, valores e regras ao Scrum e que avança conforme o planejado.
- *Product Owner*: É oficialmente responsável pelo projeto, pelo gerenciamento, controle e por tornar visível a lista de *Product Backlog*. Ele é selecionado pelo *Scrum Master*, cliente e gerente.
- *Scrum Team*: É o time do projeto que tem a autoridade para decidir as ações necessárias e organizá-las em ordem para atingir os objetivos de cada Sprint.
- Cliente: Participa nas tarefas relacionadas aos itens do *Product Backlog* para o sistema ser desenvolvido ou aprimorado.
- Usuário: É responsável por utilizar o produto quando este estiver em produção.
- Gerente: É responsável pela decisão final, juntamente com as normas e convenções a serem seguidas no projeto. O gerente também participa da definição das metas e dos requisitos.

Durante a *Sprint*, o time, de forma organizada, controla como as tarefas devem ser executadas. Durante a *Sprint* não deve existir interferência externa, esse é um dos principais papéis do *Scrum Master*, blindar o time de qualquer desvio do objetivo traçado. O acompanhamento do progresso é feito realizando reuniões diárias (*daily meeting*). Todos participam, o *Scrum Master* e o time. Visitantes são bem vindos, mas devem ser apenas ouvintes, pois o *daily meeting* resume-se ao time [Pereira et al 2007].

3. Execução de um Projeto com Scrum

A empresa em que foi aplicado o estudo de caso se enquadra no ramo de desenvolvimento de software. É uma entidade civil sem fins lucrativos que se orgulha por ser uma das referências nacionais em qualidade e eficiência na área de tecnologia da informação e comunicação. Possui ISO 9001:2008 e CMMI nível 5 e busca a melhoria contínua dos seus processos.

O projeto na qual foi desenvolvido era para a plataforma de hardware Windows XP profissional, a ferramenta de desenvolvimento era o Microsoft Visual Studio 2008, a ferramenta de banco de dados SQL Server 2005, a ferramenta de análise e projeto o Enterprise Architect, ferramenta de acompanhamento de bugs JIRA, sendo o mesmo produzido para a plataforma desktop. O projeto possuía 221 pontos de caso de uso técnicos (TUCP) e era enquadrado na modalidade de Fábrica de Solução, na qual são projetos em que a equipe do projeto trabalha com a equipe do cliente desde a etapa inicial de avaliação de necessidades e concepção de requisitos, ajudando o cliente a gerar uma solução de alto valor agregado para seu negócio.

Visando a necessidade da melhoria no gerenciamento dos projetos resolveu-se aplicar Scrum em um projeto e analisar o desempenho de produtividade do projeto com o emprego desta metodologia.

Aplicar Scrum traz várias mudanças, principalmente culturais na empresa. Para o início da utilização do Scrum, como primeiro passo aplicou-se um treinamento para todos os colaboradores para que todos pudessem conhecer as atividades a serem desempenhadas na nova metodologia de gestão de projeto e assim nivelar o conhecimento adquirido.

No treinamento foram repassados os conhecimentos a cerca do ciclo do Scrum e mostrado em detalhes cada evento a ser executado com o emprego da metodologia ágil, assim como as vantagens e facilidades proporcionadas pelo Scrum.

O segundo passo foi realizar o planejamento inicial do projeto. Cada *Sprint* teve sua duração definida em três semanas, assim a cada rodada tinha que ser entregue uma parte incremental do produto testado e funcionando. No total, foram definidas 5 *Sprints* para a conclusão do projeto.

A princípio, foram definidas as seguintes atividades a serem realizadas no projeto e estas foram enquadradas no *Backlog*:

- Levantamento dos requisitos.
- Especificação dos requisitos.
- Análise e Projeto.
- Especificação de testes.
- Revisão técnica.
- Implementação.
- Testes.
- Correção.
- Entrega.

Em seguida, o plano do projeto foi montado e apresentado ao cliente. O cliente foi envolvido inicialmente com participação ativa de forma remota para a definição das prioridades das atividades.

Para cada *Sprint* realizou-se um *Sprint Planning Meeting*, ou seja, uma reunião para planejamento da *Sprint*, de modo que pudesse definir dentre as atividades do *Backlog* aqueles que seriam executadas na *Sprint*. Além disso, a cada início do dia o gerente do projeto realizava a *Daily Meeting*, ou seja, a reunião diária para acompanhamento das atividades do projeto (atividades a fazer, atividades finalizadas, atividades em andamento), assim como para a identificação dos impedimentos ocorridos no dia anterior para que fossem resolvidos o mais rápido possível. Estas reuniões tinham duração de no máximo 15 minutos.

O próximo passo foi executar cada *Sprint*. A cada entrega, ou seja, decorridos 3 semanas, o time realizou a reunião de revisão (*Sprint Review*) para apresentação do produto realizado na *Sprint*. Após esta reunião fazia-se uma reunião de retrospectiva (*Sprint Retrospective*) para demonstrar as lições aprendidas. Nestas reuniões, foi possível identificar os principais desafios enfrentados pela equipe. Além disso, ao final de cada *Sprint*, o gerente realizou as medições dos indicadores de desempenho de produtividade do projeto.

4. Resultados Práticos do Uso do Scrum

4.1. Desempenho dos Integrantes

Os resultados foram satisfatórios, obteve-se uma satisfação por todos os integrantes do projeto na aplicabilidade do Scrum. O projeto foi entregue dentro do prazo e do orçamento melhor do que os previstos.

Constatou-se uma maior participação do cliente no processo de desenvolvimento do software proporcionando um acompanhamento em alto nível do andamento das atividades realizadas. Além disso, observou-se a satisfação do cliente na solicitação das modificações dentro de prazo hábil para realizá-las, além do recebimento de funcionalidades totalmente implementadas no final das *Sprints*. Um fator característico do Scrum que apresentou-se satisfatório para o cliente trata-se do tempo fixo estimado para as *Sprints*.

A equipe evoluiu profissionalmente se tornando mais segura com relação à capacidade de estimativa e auto-gerenciamento, descartando a necessidade de atribuição de tarefas pelo gerente. Esse crescimento foi gradativo no decorrer das *Sprints*. Aumentou também a segurança no que estava desenvolvendo e no conhecimento dos requisitos. Isto proporcionou um menor retrabalho por não desperdiçar tempo no desenvolvimento de requisito confuso. O aumento da segurança aumentou o comprometimento e o foco com o projeto. Além do mais, a equipe, depois de experimentar o Scrum, quer sempre que possível, seguir esta prática nos novos projetos.

O gerente apontou a facilidade em solucionar os impedimentos do projeto, haja vista que os mesmos eram identificados precocemente e não apresentava impactos nas demais atividades. Todos os integrantes tinham conhecimento do impedimento e através de uma ação em conjunto o impedimento era solucionado o mais rápido possível. Além disso, o gerente relatou a facilidade de extrair informações gerenciais do projeto através dos quadros adotados pela metodologia ágil, pois conforme é relatado por Schawber (2004) o Scrum oferece um framework e um conjunto de práticas que guardam tudo visível. Isto permite aos participantes do Scrum saber exatamente o que está

acontecendo e fazer no local os ajustes para manter o projeto na direção dos objetivos desejados.

4.2. Indicadores de Desempenho de Produtividade do Projeto

Como mencionado anteriormente, o gerente realizou o acompanhamento do desempenho da produtividade do projeto. A Tabela 1 a seguir apresenta a produtividade da equipe no projeto.

Tabela 1. Produtividade da equipe

Processo	Produtividade	
	Planejada	Realizada
Requisitos	20,00	11,97
Análise e Projeto	20,00	8,65
Codificação	20,00	14,11
Testes	20,00	19,80
Produtividade dos Processos com Retrabalho	23,15	13,32
Produtividade dos Processos sem Retrabalho	23,68	12,37
Produtividade Geral	26,70	16,72

A Tabela 1 ilustra os principais processos de engenharia executados no desenvolvimento do software e a produtividade planejada e realizada para cada um desses processos. Além disso, são expressos valores de produtividade para “Produtividade dos processos com retrabalho” que retrata a produtividade do projeto contando o esforço gasto com o retrabalho de alguma atividade; “Produtividade dos processos sem retrabalho” que representa a produtividade do projeto sem considerar o esforço gasto com o retrabalho das atividades; e “Produtividade geral” que significa a produtividade do projeto expressa em um único valor.

Por normas de sigilo, a forma como é calculada a produtividade não foi divulgada pela empresa. No entanto, alguns fatores nos permitem a análise destes dados. A produtividade planejada representa o nível de produtividade planejado pelo gerente de acordo com as características do projeto e representa o limite máximo a ser atingido. Quando a produtividade realizada é expressa abaixo da produtividade planejada, significa dizer que a equipe apresenta boa produtividade e que está realizando suas atividades com um esforço inferior ao planejado. Deste modo, quando menor a produtividade realizada, melhor está sendo a produtividade da equipe.

Como se pode constatar pela análise da Tabela 1 acima representada, em todos os processos a equipe apresenta uma boa produtividade, haja vista que todos os valores expressos na “Produtividade realizada” são inferiores aos valores da “Produtividade planejada”.

Uma análise a respeito da diferença entre “Produtividade planejada” e “Produtividade realizada” é apresentada na Figura 2.

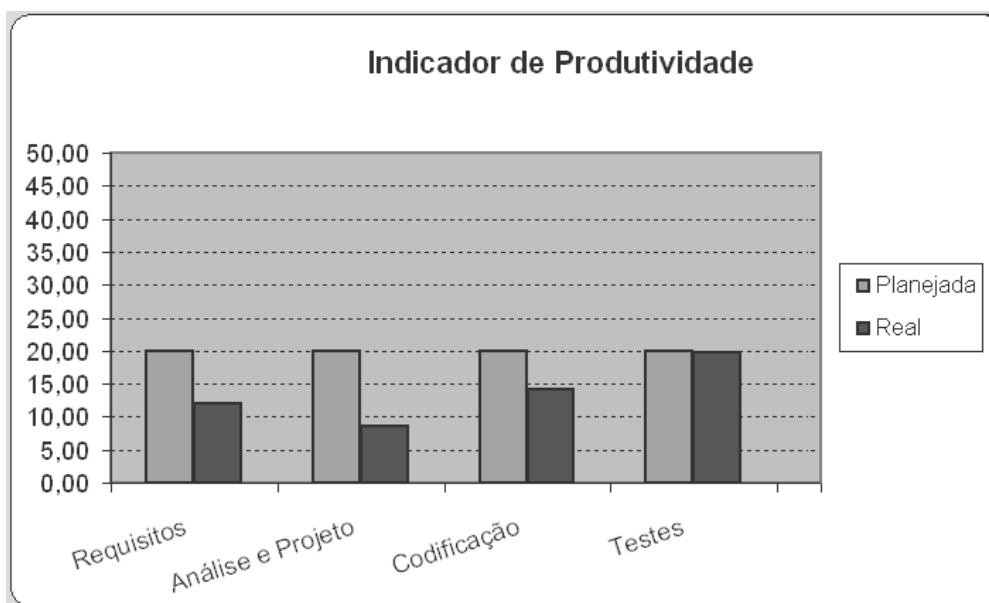


Figura 2. Gráfico da Produtividade da equipe

Na Figura 2 acima representada, é possível verificar que além da produtividade realizada apresentar-se inferior a produtividade planejada em todos os processos, o processo de “Análise e Projeto” apresenta uma melhor produtividade se comparado o valor aos demais processos. Em contrapartida, o processo de “Testes” apresenta uma boa produtividade, no entanto, bem próxima a produtividade planejada.

Com o aumento da produtividade da equipe, ou seja, a execução das atividades em menor tempo que o estimado, o projeto foi entregue em tempo menor que o planejado. No geral, foram planejados 5 meses para a execução total do projeto e o projeto foi realizado em 4 meses. Isto proporcionou uma maior satisfação do cliente assim como o aumento do lucro da empresa.

5. Considerações Finais

O gerenciamento de projetos é uma área de extrema importância para determinar o sucesso de um projeto que tem como objetivo otimizar custos e prazo, levando em conta a satisfação de todos os integrantes, controlando o escopo e a qualidade do produto.

Em virtude da necessidade de melhoria contínua, característica de uma empresa que possui maturidade de processo CMMI 5, com o emprego de uma nova forma de gerenciar os projetos de desenvolvimento de software, utilizando Scrum, consegue-se ter a otimização dos recursos e a maximização dos lucros.

Através da utilização da metodologia ágil Scrum, é possível produzir o software em um menor tempo, sem perder a qualidade do produto produzido, tendo um ganho de tempo em relação ao tempo total planejado do projeto, razão essa que levou a satisfação dos integrantes, desde a equipe até o cliente.

6. Referências Bibliográficas

Abrahamsson, P.; Salo, O.; Ronkainen, J. (2002) Agile Software Development Methods – Review and Analysis. In: VTT Publications 478, Espoo.

- Agile Manifesto. (2011) “Manifesto for Agile Software Development”. <http://www.agilemanifesto.org/>, Agosto.
- Ambler, S. (2002) Agile Modeling, Wiley Computer Publishing. New York.
- Carvalho, B.V.; Mello, C.H.P. (2009) Revisão, Análise e Classificação da Literatura sobre o Método de Desenvolvimento de Produtos Ágil Scrum. XII Simpósio de Administração da Produção, Logística e Operações Internacionais (SIMPOI), São Paulo.
- Dantas, V. F. (2003) Uma Metodologia para o Desenvolvimento de Aplicações Web num Cenário Global. Dissertação. Universidade Federal de Campina Grande. Campina Grande, Julho.
- Gerenciamento de Projetos. (2008) In: *Revista Eletrônica Promon – Business & Technology Review*. http://www.promon.com.br/portugues/noticias/download/PBTR%20GE_para%20web.pdf, Agosto.
- Highsmith, J. (2004) Agile Project Management, Creating innovative products. Addison Wesley.
- Highsmith, J. & Cockburn, A. (2001) Agile Software Development: The Business of Innovation. Computer, p. 120-122.
- Koscianski, A. Qualidade de Software: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2ª ed. São Paulo: Novatec Editora, 2007.
- Larman, C.; Basili, V. R. (2003) Iterative and Incremental Development: A Brief History. In: *Computer* 36. nº 6, p. 47-56, Junho.
- Mann, C. & Maurer, F. (2005) A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction. Agile Development Conference, p. 70-79. IEEE Computer Society.
- Mundim, A.P.F.; Rozenfeld, H.; Amaral, D.C.; Silva, S.L.; Guerreiro, V.; Horta, L.C. (2002) Aplicando o cenário de desenvolvimento de produtos em um caso prático de capacitação profissional. In: *Gestão & Produção*. v. 9, nº. 1, p. 1-16, Abril.
- PMBOK, GUIDE. (2009) Um Guia do Conhecimento em Gerenciamento de Projetos. 4ª ed. Versão em Português. Pennsylvania: Project Management Institute, Four Campus Boulevard, Newtown Square.
- Rising, L.; Janoff, N. (2000) The Scrum software development process for small teams. In *IEEE*, v. 17, nº 4, p. 26-32.
- Pereira, P.; Torreão, P.; Maçal, A. S. (2007) Entendendo Scrum para Gerenciar Projetos de Forma Ágil. In: *Mundo PM*.
- Pressman, R. S. (2006) Engenharia de Software. 6ª. ed. São Paulo: MCGrawHill.
- Schwaber, K. (1995). “Development Process”. <http://jeffsutherland.com/>, Agosto.
- Schwaber K. (2004) Agile Project Management With Scrum, Microsoft.
- Schwaber, K.; Beedle, M. (2007) Agile Software Development with Scrum. Editora Prentice Hall.

Silva, M. A. C.; Roriz Filho, H.; Silva, H. F. N. (2010) Análise do BA durante o Processo Scrum. XVII Simpósio de Engenharia de Produção. Bauru – SP, Novembro.

Sommerville, I. Engenharia de Software. São Paulo: Addison Wesley, 2004.

Tavares, A. (2008) Gerência de Projetos com PMBOK e SCRUM – Um estudo de caso. Faculdade Cenecista Senhora dos Anjos. Gravataí - RS.

Vriens, C. (2003) Certifying for CMM Level 2 and ISO 9001 with XP & Scrum. Agile Development Conference, p. 120-124.