



FANESE – Faculdade de Administrações e Negócios de Sergipe

Núcleo de Pós-Graduação e Extensão – NPGE
ESPECIALIZAÇÃO EM GESTÃO EM PROJETOS DE SOFTWARE

FERNANDO HENRIQUE VIEIRA TRINDADE

A EVOLUÇÃO DAS METODOLOGIAS ÁGEIS PARA O GERENCIAMENTO DE PROJETOS EM EQUIPES DE DESENVOLVIMENTO DE SOFTWARE

Aracaju - SE
14 de setembro de 2018

FERNANDO HENRIQUE VIEIRA TRINDADE

A EVOLUÇÃO DAS METODOLOGIAS ÁGEIS PARA O GERENCIAMENTO DE PROJETOS EM EQUIPES DE DESENVOLVIMENTO DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao Núcleo de Pós-Graduação e Extensão – NPGE, da Faculdade de Administração de Negócios de Sergipe – FANESE, como requisito para a obtenção do título de Especialista em Gestão em Projeto de Software

Nome completo do Avaliador

Nome completo do Coordenador de Curso

FERNANDO HENRIQUE VIEIRA TRINDADE

Aprovado (a) com média: _____

Aracaju (SE), ____ de _____ de 2018.

RESUMO

As empresas que mais crescem nos dias de hoje são as startups da área de tecnologia da informação. Cita-se a título de exemplo os seguintes casos exitosos: Paycom Software, NetEase, Facebook dentre outros. Um dos fatores cruciais para que uma startup se torne uma grande empresa é a forma como ela gerencia os projetos. Este artigo tem como objetivo, comparar os mais recentes frameworks ágeis de gestão com as metodologias de projetos ditas “tradicionais”, focando em qual seria a abordagem mais adequada para a realidade de uma corporação de desenvolvimento de software. Para isso foi utilizada a metodologia descritiva e explicativa por meio de pesquisa bibliográfica. Como resultados tem-se que as metodologias tradicionais são muito “engessadas”, baseadas em etapas complexas e possuem muitos artefatos. Já as metodologias ágeis possuem foco na interação entre stakeholders, empoderamento das equipes de trabalho e gestão mais flexível. À vista disso, as conclusões deste trabalho versam metodologias ágeis são as melhores opções para levar essas organizações ao sucesso de forma rápida e consistente.

Palavras-chave: Gerência. Desenvolvimento. Metodologia. Ágil. Startup.

LISTA DE TABELAS

Tabela 1 – Quadro Ágil x Tradicional.....	19
---	----

LISTA DE FIGURAS

Figura 1.....	10
Figura 2.....	16
Figura 3.....	18

SUMÁRIO

RESUMO	
LISTA DE FIGURAS	
LISTA DE TABELAS	
1. INTRODUÇÃO.....	6
2. DESENVOLVIMENTO.....	8
2.1. Processo X Projeto.....	8
2.2. PMBOK.....	8
2.3. Metodologias ágeis.....	11
2.4. Comparação entre PMBOK e <i>Scrum</i>.....	19
3. CONSIDERAÇÕES FINAIS.....	21
REFERÊNCIAS.....	22
ABSTRACT.....	23

1 INTRODUÇÃO

Desde o advento da criação dos computadores e da internet, grandes empresas, indústrias e comércios têm utilizado o software como ferramenta fundamental para controle e processamento de informações. No mundo corporativo atual a informação fornecida de forma segura e no momento certo é um dos ativos mais importantes para levar uma empresa ao sucesso no mercado. Dessa maneira, sistemas de computadores precisam ser construídos cada vez mais rápido respeitando diversos critérios de qualidade, pois uma falha pode gerar dano irreparável ao empreendimento. Entretanto o processo de desenvolvimento de um sistema não é nada simples e requer um gerenciamento efetivo (eficiente, eficaz e com qualidade). Por conseguinte, diversas metodologias de gestão de projetos foram desenvolvidas ao longo dos últimos anos com a finalidade de gerir a produção de softwares de maneira mais assertiva.

Por ser uma área relativamente recente, a engenharia de software, no seu surgimento aproveitou diversos conceitos de gestão de outras áreas a exemplo da engenharia civil. Sendo assim as primeiras práticas de gestão ditas “tradicionalis” eram aplicadas no processo de desenvolvimento de sistemas de forma pouco eficaz. Para resolver esta adversidade no final dos anos 90 engenheiros de softwares começaram a criar práticas que vieram a se tornar as metodologias ágeis. De acordo com Pressman (2011), as metodologias ágeis têm o propósito de sanar os problemas das metodologias tradicionais, em tempo hábil, num ambiente onde os requisitos de negócio mudam constantemente.

Empresas conhecidas como *startups* de tecnologia são criadas a partir de uma ideia de negócio que geralmente são desenvolvidas por equipes pequenas, com baixo orçamento e num lapso de tempo de mercado para atender à uma oportunidade. A utilização de métodos ágeis no gerenciamento de *startups* vem se mostrando como meio propício para conduzir equipes de trabalho que entregam resultado no tempo adequado gerando valor para os clientes. Métodos ágeis de gestão a exemplo do *Scrum* através de práticas como: energizar pessoas, empoderar times, desenvolver competências, entre outras responde aos problemas citados anteriormente. O interesse pelo tema a ser estudado surgiu, principalmente, pela necessidade de conhecer como os frameworks ágeis aumentam a velocidade

do desenvolvimento de softwares e evitam os fracassos. Para isso, faz-se necessário a comparação entre o conjuntos de boas práticas de gestão tradicional mais utilizado: PMBOK e a ¹aplicação do framework (SCRUM) em startup de desenvolvimento de software.

O objetivo deste trabalho é analisar a promoção de melhorias de gerenciamento a partir de métodos ágeis no processo de gestão de desenvolvimento de software aplicado a startups. Como objetivos específicos pretende-se identificar características do Scrum, como também o comparar com o PMBOK e analisar qual método de gestão ajusta-se melhor ao universo das startups. A metodologia adotada para este artigo será descritiva e explicativa por meio de pesquisa bibliográfica.

1 Fernando Henrique Vieira Trindade
Bacharel em Sistemas de Informação - UNIT

contato@fernandohenrique.com

2 DESENVOLVIMENTO

2.1 Processos X Projeto

Antes de apresentar os *frameworks* de gestão abordados nesse trabalho é preciso definir o que é um projeto, o que é um processo e quais as suas semelhanças e diferenças. Segundo PMI (2013) "Projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo" PMI (2013). Portanto sempre que se executa uma ou várias tarefas que tenham início, meio e fim, utiliza-se recursos humanos ou materiais com a finalidade de produzir algo único define-se um projeto. Por sua vez o conceito de processo segundo PMI (2013) é um conjunto de ações e atividades inter-relacionadas, que são executadas para criar um produto, resultado ou serviço predefinido. Cada processo é composto por entradas, ferramentas, técnicas e saídas. Embora as duas definições contenham características como: realização de tarefas e produzir resultado, o que de fato as diferenciam é que um processo é contínuo e repetitivo enquanto um projeto é temporário e único. Ainda de acordo com o PMI (2013) Gestão de projetos é aplicar conhecimentos, habilidades, ferramentas e técnicas para projetar atividades como o objetivo de atender aos requisitos do projeto.

2.2 PMBOK

O guia *Project Management Body of Knowledge (PMBOK)* é um conjunto de boas práticas em gestão de projetos agrupadas em formato de guia pelo PMI e é considerado a uma fonte sobre assuntos relacionados a gestão de projetos. A primeira edição do "*A Guide to the Project Management Body of Knowledge (PMBOK Guide)*" foi publicada em 1996 pelo *Project Management Institute (PMI)*, em 2004 foi publicada a versão que trouxe mais mudanças em relação a primeira. A publicação mais recente do guia foi em 2017, também conhecida como PMBOK sexta edição. Embora seja um guia elaborado pelo agrupamento de práticas de gestão consolidadas pelo mercado, não significa que é mandatório, não é necessário aplicar todas as práticas. Uma das vantagens de seguir este documento é a padronização de termos definidos em seu glossário. Esta uniformização de taxonomias permite trocar experiências de sucesso e fracasso com outros profissionais da área de gestão utilizando os mesmos conceitos. Este guia é descrito

em forma de processos que indicam quem deve fazer o quê, e quando durante a gestão de um projeto. Os processos descritos interagem entre eles durante o ciclo de vida do projeto e são descritos por:

- a. Entradas (artefatos, produtos, etc.).
- b. Ferramentas e técnicas (que serão aplicadas nas entradas).
- c. Saídas (produtos, artefatos gerados a partir da aplicação de ferramentas e técnicas às entradas e podem vir a ser entradas de outro processo).

2.2.1 Ciclo de vida

O PMBOK, 5ed. estabelece que todo projeto tem intrinsecamente cinco etapas durante sua execução. Cada etapa do projeto em uma linha temporal tem suas características específicas assim sendo o guia organiza grupos de trabalhos (processos) para cada uma delas. as cinco etapas (grupos de processos) são:

- a) Iniciação: nessa fase é definido o termo de abertura do projeto e identifica-se as partes interessadas..
- b) Planejamento: fase de definição e refinamento de objetivos, deve-se determinar tudo que será feito e procurar as melhores soluções para realizar os objetivos aqui definidos.
- c) Execução: fase de execução dos planos do projeto: coordenação de pessoas e gerenciar recursos para executar o que foi planejado, é a parte mais operacional do projeto.
- d) Processos de Monitoramento e Controle: medição e monitoramento do andamento do projeto. Verifica-se a conformidade dos objetivos do projeto (planejado) com o que está sendo executado através do monitoramento e medição constantes do progresso. Caso seja encontrada alguma não conformidade ações corretivas ou "mitigantes" devem ser realizadas.
- e) Processos de Encerramento: nessa fase ocorre a finalização do projeto. Deve-se analisar se o projeto foi concluído atendendo aos requisitos de escopo, prazo, custo e qualidade ou se foi encerrado por não ser mais viável.



Figura 1

Fonte: Cruz (2013).

2.2.2 Áreas de conhecimento

Com a finalidade de organizar as atividades de um projeto o PMBOK agrupou os processos por áreas de conhecimento. Cada processo é definido para atingir um ou mais objetivos de uma área específica. Dessa forma é possível reunir processos que objetivam gerenciar de forma eficiente e eficaz os riscos, custos, cronograma e assim por diante. Na 5ª edição do PMBOK essas divisões são chamadas de áreas de conhecimento. Segundo PMI (2013) as dez áreas descritas pelo guia são:

- Gestão de integração do projeto:** consiste em processos que servem para coordenar os componentes do projeto garantindo que tudo aconteça de maneira organizada. São os processos do dia a dia do gerente de projetos.
- Gerenciamento do escopo:** em conformidade com o guia esta área é constituída pelos processos que garantem o “tamanho do projeto”, ou seja, que seja feito o esforço essencial para concluir o projeto com sucesso e nada além disso. O escopo também pode definir qual trabalho não deverá ser realizado.
- Gerenciamento do tempo:** consiste em processos que visam garantir que o projeto seja finalizado consoante aos prazos previamente estabelecidos no planejamento. Esta área compila atividades que geram e controlam o cronograma do projeto.

- d. Gerenciamento de custo do projeto: área que agrega os processos de planejamento orçamentário do projeto. Reuni as atividades que definem e gerenciam os custos necessários para viabilizar o projeto.
- e. Gerenciamento da qualidade do projeto: consiste em processos que se destinam a respeitar os critérios, especificações, de qualidade estabelecidas no planejamento. Asseguram a qualidade essencial que atenda às necessidades do cliente.
- f. Gerenciamento dos recursos humanos do projeto: consiste em processos que pretendem distribuir os *stakeholders* (partes envolvidas) nas atividades certas no momento certo para que o projeto seja executado de forma eficiente e eficaz.
- g. Gerenciamento das comunicações do projeto: consiste em processos que asseguram que toda informação importante para o projeto seja coletada e distribuída para as pessoas certas.
- h. Gerenciamento dos riscos do projeto: consiste em processos tocantes aos riscos inerentes do projeto. Os riscos podem ser oportunidades ou ameaças ao projeto. No caso das ameaças é preciso analisar e elaborar planos de contingência ou mitigação dos riscos.
- i. Gerenciamento das aquisições do projeto: área responsável pelo planejamento e execução de aquisições de bens ou serviços necessários para o projeto.
- j. Gerenciamento das partes interessadas: consistem em processos relacionados a identificação de qualquer pessoa que possa impactar positivamente ou negativamente o projeto.

2.3 Metodologias ágeis

Segundo Prikladnicki, Wili e Milani (2014), por volta dos anos 90 começaram a surgir novas metodologias de desenvolvimento de software para resolver problemas constantes causados pelo uso das metodologias tradicionais. As metodologias conservadoras da época eram consideradas, “pesadas” por apresentarem características como: excessivamente regradas, lentas, burocráticas e inadequadas à atividade de desenvolvimento de software. Após 2001 as novas

metodologias que eram chamadas de “leves” passaram a ser chamadas de Ágeis. Houve uma reunião em Utah, nos Estados Unidos, para debater maneiras de resolver os problemas de desenvolvimento supracitados. Neste encontro 17 especialistas criaram o que hoje é conhecido como “Metodologias Ágeis” e a partir daí em 2001 foi publicado o Manifesto Ágil e qualquer pessoa pode ser signatária deste documento. Em 2014 algumas dessas pessoas criaram uma organização filantrópica, a Agile Alliance, para promover o desenvolvimento ágil. O manifesto segundo (WARD, C. et al. 2001) foi elaborado baseado em 4 valores:

- a. Indivíduos e interações são mais importantes que ferramentas e processos.
- b. Software em funcionamento é mais importante que documentação abrangente.
- c. Colaboração com o cliente é mais importante que negociações de contratos.
- d. Responder rápido as mudanças é mais importante que seguir um plano. E a partir desses valores foram criados os 12 princípios com a finalidade de melhorar o desenvolvimento de softwares, que são:
 - e. Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.
 - f. Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.
 - g. Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
 - h. Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.
 - i. Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.
 - j. O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face.
 - k. Software funcionando é a medida primária de progresso.
 - l. Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.

- m. Contínua atenção à excelência técnica e bom design aumenta a agilidade.
- n. Simplicidade, a arte de maximizar a quantidade de trabalho não realizado, é essencial. As melhores arquiteturas, requisitos e designs emergem de equipes auto organizáveis.
- o. Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

2.3.1 Scrum

Segundo Chon (2011), o Scrum não é um processo, muito menos um algoritmo de como se construir algo. Ele se trata de um framework, ou seja, um conjunto de conceitos e melhores práticas utilizados para resolver problemas e atingir um objetivo.

De acordo Schwaber e Stherland (2014), o Scrum é: leve, simples de entender e extremamente difícil de dominar. Cada membro dentro da equipe Scrum tem um papel definido, existem regras a serem seguidas, artefatos a serem desenvolvidos e eventos ocorrendo e que irão ocorrer. Cada parte dessas tem um objetivo específico para o sucesso da implementação do framework. Embora possua regras bem definidas sobre as práticas, bastante efetivas, de gerenciamento de projetos, não impede que sejam aplicadas novas práticas ou que as existentes sejam otimizadas. Este conjunto de processos ou técnicas está sendo utilizado desde os anos 90 para resolver problemas complexos e adaptativos de forma criativa entregando produtos ou serviços que agregam valor ao negócio.

O Scrum possui seu alicerce em teorias empíricas. De acordo com Buarque (2010), empirismo é um conjunto de conhecimentos adquiridos com a prática e experiência. Ele tem caráter iterativo e incremental, dessa maneira ocorre um aumento da experiência e conseqüentemente os riscos diminuem. Existem três pilares que apoiam o empirismo dentro do Scrum, são eles: transparência, inspeção e adaptação. A transparência discorre sobre a necessidade de os processos de trabalhos de estarem visíveis e compartilhados com os envolvidos. Por exemplo:

- a. Uma linguagem padronizada que descreve o processo deve se de conhecimento de todos os stakeholders envolvidos no projeto.

- b. O conceito de produto\serviço “Finalizado” deve ser de comum acordo entre aqueles que executam o processo e os que o recebem em forma de produto. A inspeção versa sobre a necessidade de inspecionar se o framework está sendo implementado de maneira correta, dessa maneira garante uma evolução contínua otimizando os processos sem que atrapalhe o andamento das atividades. Por fim, a adaptação, que diz que se o produto saiu do escopo padrão, ou algum processo não condiz com os padrões aceitáveis deve ser ajustado o quanto antes para evitar acúmulos de trabalho desnecessário ou desperdícios (COHN, 2011). O Scrum prescreve quatro eventos formais, contidos dentro dos limites da Sprint, para inspeção e adaptação:
- a. Reunião de planejamento da Sprint: Reunião de planejamento realizada para definir o plano de trabalho da Sprint. O plano é definido de forma colaborativa entre todos os membros do time. Deve ter duração de no máximo oito horas para uma Sprint de um mês. Procura responder as seguintes questões: o que pode ser entregue como resultado de incremento da próxima Sprint? E como será realizado o trabalho para entregar o incremento?
 - b. Reunião diária: É uma reunião feita diariamente com duração de 15 minutos, com a finalidade de inspecionar o que foi feito no dia anterior, comunicar a todos o andamento das atividades e principalmente planejar os passos do dia seguinte.
 - c. Reunião de revisão da Sprint: É feita ao final da Sprint para verificar se os objetivos da Sprint foram atingidos. Sua duração é de 4 horas para uma Sprint de um mês.
 - d. Retrospectiva da Sprint: É uma oportunidade de autoavaliação do time objetivando a melhoria contínua nos processos para ser aplicada na próxima Sprint. É feita ao final da Sprint e tem duração de três horas. Segundo Schwaber e Sutherland (2014), dentro de um projeto existem ciclos, já que o Scrum é iterativo e incremental, que são chamados de Sprints. A Sprint é o principal evento do Scrum. Assim, cada Sprint deve possuir sua execução planejada dentro de duas a quatro semanas com a finalidade de produzir um incremento “Pronto” utilizável pelo cliente. Uma Sprint contém um conjunto

específico de eventos (reunião de planejamento, reuniões diárias, trabalho de desenvolvimento, revisão e retrospectiva da Sprint) e regras:

- a. Não serão aceitas mudanças que possam modificar o objetivo da Sprint, essas serão avaliadas para uma próxima etapa;
- b. As metas de qualidade não devem ser relaxadas;
- c. O escopo pode ser melhor entendido e renegociado entre o Product Owner e o Time de Desenvolvimento. O framework define basicamente três papéis que são fundamentais para o seu funcionamento. Este grupo de papéis é denominado de Time do Scrum. O time do Scrum é auto gerenciável, multifuncional (facilita a disseminação do conhecimento) e é composto por:
 - a. Product Owner: O dono do produto é responsável por definir e gerenciar um Product Backlog (escopo com as partes entregáveis) que agregue valor ao negócio do cliente. Entre as suas responsabilidades pode-se citar: Ordenar os itens para maximizar as metas e missões, garantir que o time de desenvolvimento entenda os itens do backlog no nível necessário, garantir o valor do trabalho realizado pelo time de desenvolvimento, entre outros.
 - b. O Time de Desenvolvimento: É composto pelos profissionais que de fato realizam o trabalho para produzir um incremento pronto ao final de cada Sprint. O time de trabalho possui características das quais exemplifica-se: são auto organizáveis, são multifuncionais, a responsabilidade é de todos e devem ser compostos por pelo menos três, mas no máximo nove integrantes.
 - c. Scrum Master: Este papel é responsável por assegurar que as práticas e valores do framework serão exercidas em conformidade com guia. Dentre as atribuições do Scrum Master pode-se exemplificar: facilitar os eventos scrum conforme exigidos ou necessários, treinar o time em autogerenciamento e interdisciplinaridade, remover impedimentos para o progresso do time de desenvolvimento, etc. Além dos eventos e papéis supracitados o scrum preconiza a utilização de artefatos. Segundo Schwaber e Sutherland (2014), Os artefatos do Scrum representam o trabalho ou o valor para o fornecimento de transparência e oportunidades para inspeção e adaptação. Os três artefatos estabelecidos pelo guia são:

- a. Product Backlog: É uma lista ordenada por prioridade de todos os itens(escopo) geradores de valor necessários no produto. É a única fonte de requisitos. O Product Owner é o responsável por mantê-la.
- b. Backlog da Sprint: É uma seleção de itens retirados do product backlog para compor o trabalho e plano necessários para concretização da sprint.Torna transparente todo o trabalho que o Time de Desenvolvimento terá para finalizar um incremento.
- c. Incremento: É o resultado da Sprint. Pode-se caracterizar pelo somatório de tudo que foi realizado e definido como “Pronto” para ser entregue. Deve estar em condição de ser implantado para que o Product Owner decida qual o melhor momento de liberá-la em ambiente de produção (efetivo uso pelo cliente).

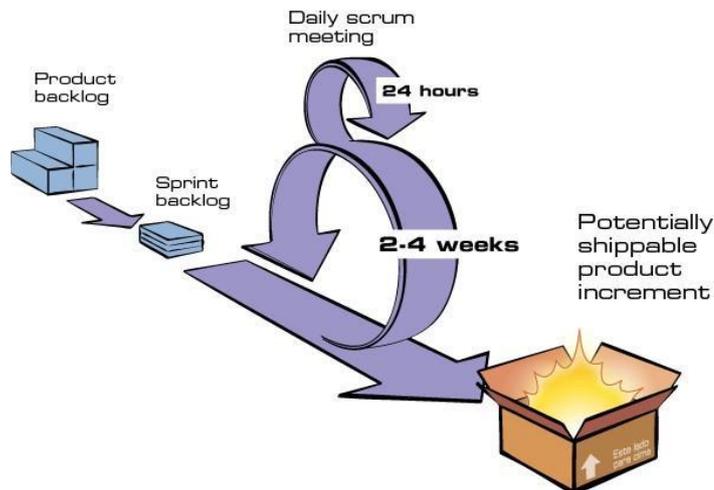


Figura 2 - Diagrama Scrum.

Fonte: Guia do Scrum (2013)

Na Figura 2, pode-se ver um diagrama que mostra como fluxo iniciado com a definição do product backlog que em seguida será fracionando em sprint backlogs, estes por sua vez serão executados durante a sprint, inspecionados através dos eventos de reuniões, e ao final produzirão um incremento.

2.3.2. Extreme Programming (XP)

Segundo Wells (2013), a grande característica do XP é a sua simplicidade e o trabalho em equipe. Diminuir a burocracia dos métodos tradicionais de desenvolvimento e focar naquilo que é realmente necessário e assim, aumentar a

produtividade. A metodologia XP possui na sua essência alguns valores e princípios básicos. Os valores são: comunicação, coragem, simplicidade, feedback e respeito. Para o XP todos os envolvidos devem conhecer o projeto da melhor maneira possível. Além disso, com um melhor feedback é possível saber com antecedência sobre erros ou saídas do escopo, as vezes por conta de uma informação errada do cliente. Isso trás benefícios, pois quanto mais tempo as mudanças demorarem para serem feitas, mais custosas elas serão. Como a própria nomenclatura em inglês diz, “programação extrema”, é preciso ter coragem e enfrentar os riscos, até porque o próprio XP utiliza meios de proteção contra esses riscos. Por fim, o respeito, seja ele dentro da equipe, para com o cliente, os usuários ou os chefes, isso reflete em uma melhor interação e em um melhor projeto.

O XP utiliza pequenas histórias que o cliente cria para o planejamento do projeto. Ela se assemelha a um Use Case e serve para elucidar sobre o que deve ser feito no projeto. Ele também é dividido em iterações e cada nova iteração passa por um planejamento.

O ciclo de vida dessa metodologia é composto por 6 fases, (AMBLER, 2012):

- Fase de exploração: é nela onde é feita uma análise de viabilidade de construção do software, ou seja, ocorre antes a construção do mesmo. Possíveis arquiteturas são pensadas e tentativas de como o software irá funcionar são feitas.
- Fase de planejamento inicial: nela é acordada com os clientes uma data marco para entrega do primeiro release do software. Além disso, juntamente com os clientes, são definidas as histórias que serão implementadas e as descrevem em cartões. Essas histórias são priorizadas e separadas por iterações (planejamento de iterações). Um release, deve durar de dois a quatro meses, e cada iteração do release de uma a três semanas.
- Fase de iterações do release: cada iteração segue uma sequência de atividades, que pode não ser seguida em toda sua completude. Primeiramente são escritos os casos de teste que serão seguidos posteriormente. Após isso, é feito um estudo de possível refatoramento de código, por motivos de mudanças. Em seguida são realizados

testes em conformidade com os casos de testes escritos, e finalmente a integração.

- Fase de produção: cada release do software é colocado em integração com as outros releases em um ambiente de produção. O foco é no comportamento desse release integrado e em produção, ou seja, ele deve atender as expectativas iniciais e serem aceitos pelo cliente.
- Fase de manutenção: uma das características do XP versa pelo fato do software ter que se adaptar às mudanças constantes. As solicitações de mudanças causam a necessidade de diversas alterações no código durante o ciclo de vida. Nesta fase preconiza-se o refatoramento, adição de novas funcionalidades, a verificação e validação.
- Fase de morte: de acordo com o Pmbok, um projeto só termina por dois motivos. A primeira, se o objetivo do projeto foi alcançado com sucesso. A segunda, se não existem possibilidades, economicamente viáveis, de se chegar nesse objetivo. Assim, essa fase versa sobre o término do projeto utilizando a metodologia ágil XP.

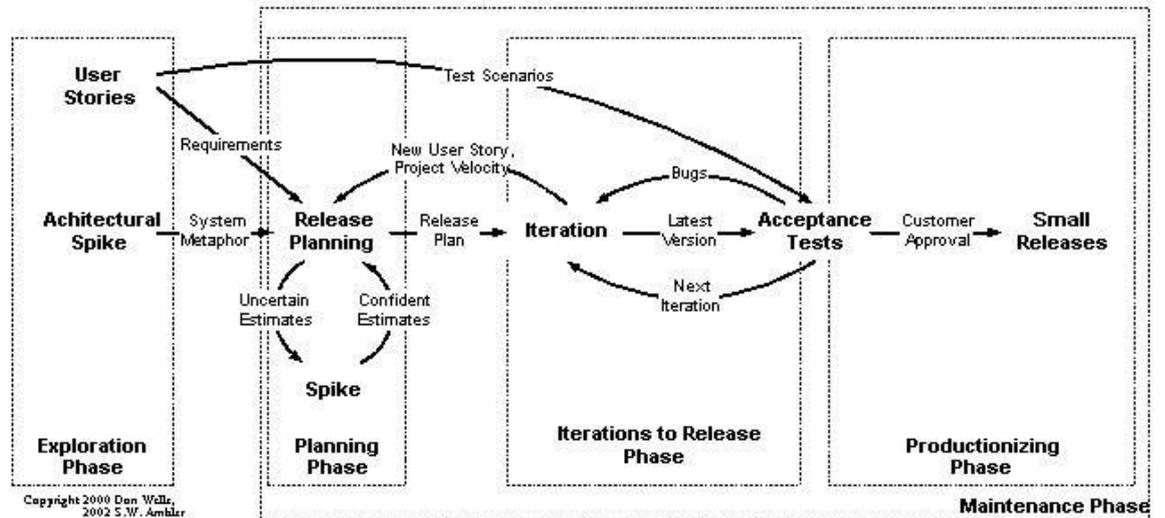


Figura 3 - Fases do XP

Fonte:Beck (2004).

A Figura 3 mostra as fases do XP e o ciclo de vida de uma pequena release, Desde as estórias do usuário, passando pelo planejamento da release, pela iteração, pela aprovação do cliente e a saída da release construída. XP também versa a necessidade de reuniões diárias e em pé, semelhante ao Scrum. A refatoração é bem vinda sempre que possível e a progressão do projeto é sempre

medida. Uma característica marcante do XP é a programação pareada, ou seja, duas pessoas programando um mesmo código. Ela se da com um só computador e duas pessoas trabalhando nele em um código. Enquanto uma está digitando a outra serve como uma espécie de verificador e validador e acaba aumentando a produtividade e a efetividade no desenvolvimento de códigos sem erros. Em adição a isso a biblioteca de teste é feita antes mesmo do código ser feito, orientação a testes é seguida por ele.

2.4 Comparações entre PMBOK e Scrum

Métodos ágeis e tradicionais possuem atributos semelhantes como também discrepantes pois tratam do mesmo assunto: gerência de projetos. O foco deste trabalho é destacar quais são os pontos conflitantes. Desta forma é possível entender os problemas causados pelo uso de métodos tradicionais de gestão durante o desenvolvimento de sistemas. É justamente na tentativa de solucionar essas dificuldades que os métodos ágeis se destacam. A Tabela 1 demonstra singularidades conflitantes entre métodos ágeis tradicionais.

Tabela 1- Quadro Ágil x Tradicional.

	TRADICIONAL	METODOLOGIAS ÁGEIS
Pressupostos fundamentais	Sistemas totalmente especificáveis, previsíveis; desenvolvidos a partir de um planejamento extensivo e meticuloso	Software adaptativo e de alta qualidade; pode ser desenvolvido por equipes pequenas utilizando os princípios da melhoria contínua do projeto e testes orientados a rápida resposta a mudanças
Controle	Orientado a processos	Orientado a pessoas
Estilo de gerenciamento	Comandar e controlar	Liderar e colaborar
Gestão do conhecimento	Explícito	Tácito
Atribuição de papéis	Individual – favorece a especialização	Times auto-organizáveis – favorece a troca de papéis
Comunicação	Formal	Informal
Ciclo do projeto	Guiado por tarefas ou atividades	Guiado por funcionalidades do produto
Modelo de desenvolvimento	Modelo de ciclo de vida (Cascata, Espiral, ou alguma variação)	Modelo iterativo e incremental de entregas
Forma/estrutura organizacional desejada	Mecânica (burocrática com muita formalização)	Orgânica (flexível e com incentivos a participação e cooperação social)

Comparando os métodos de gestão fundamentadas (*Scrum* e *PMBOK*) neste artigo, enquanto o *PMBOK* em sua 5ª edição possui 47 processos (23 somente para planejamento) é dividido em 10 áreas de conhecimento e 5 etapas ou grupos e processos, o *Scrum* preconiza poucas etapas (planejamento do backlog, planejamento dos sprints, execução dos sprints, reuniões e entrega) para execução e planejamento de projetos.

3 CONSIDERAÇÕES FINAIS

O mercado de TI está em ascensão no mundo e um fator importante para esse crescimento são as startups. Essas empresas embora iniciem suas atividades com equipes pequenas, conseguem atender de forma rápida as demandas exigidas pelo mercado. Para que isso se tornasse possível foi preciso adaptar a forma de desenvolver softwares ao longo dos anos. As necessidades de negócio das empresas que contratam serviços e produtos de tecnologia da informação mudam constantemente, e este é um ambiente extremamente competitivo que exige celeridade nas entregas. Considerando os pontos abordados nesse trabalho é possível concluir que as características como: simplicidade nos processos, pouca burocracia, foco no valor agregado e fácil adaptação à mudanças de requisitos, são de fato evoluções da gerência de processo de desenvolvimento de software. Portanto o uso de métodos tradicionais, lentos e burocráticos de gestão devem ser utilizados apenas em projetos onde o escopo é muito bem definido, por empresas que possuem uma estrutura "projetizada" para atender os papéis, artefatos e processos complexos como os do PMBOK.

REFERÊNCIAS

- BECK, K. **Extreme Programming Explained**. 2ª Edição, ISBN: 0201616416, 224p, out. 2004.
- BUARQUE, A. **Mini Dicionário Aurélio da Língua Portuguesa** 8. ed. São Paulo: Positivo, 2010.
- CRUZ, Fábio. **Scrum e PMBOK unidos no gerenciamento de projetos**. Brasport Livros e Multimida Ltda, 2013
- COHN, M. **Desenvolvimento de Software com Scrum - Aplicando Métodos Ágeis com Sucesso**. 1. ed. São Paulo: Bookman, 2011.
- FOGGETTI, CRISTIANO. **Gestão Ágil de Projetos**. 1.ed. São Paulo: Pearson, 2014.
- MIKE, B. et al. **Guia Open Source sobre Desenvolvimento Ágil feito pela comunidade de Desenvolvimento Ágil do Brasil**. 2001. Disponível em: <<http://www.desenvolvimentoagil.com.br/>>. Acesso em: 13 dez. 2014. .
- PMI. **Um guia do conhecimento em gerenciamento de projetos**. Guia PMBOK 7. ed. EUA: Project Management Institute, 2013.
- PRESSMAN, R. **Engenharia de Software: Uma abordagem Profissional**. 7. ed. São Paulo: McGrawHill, 2011.
- PRIKLADNICKI, R.; WILI, R.; MILANI F. **Métodos Ágeis para Desenvolvimento de Software**. 1. ed. São Paulo: Bookman, 2014.
- SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum. Um guia definitivo para o Scrum: As regras do jogo**. Scrum.Org and ScrumInc, 2014.
- WARD, C. et al. **The Agile Manifesto**. 2001. Disponível em: <<http://agilemanifesto.org>>. Acesso em: 22 fev Dagem Profissional. 7. ed. São Paulo: McGrawHill, 2011
- WELLS, D. **Agile Software Development: A gentle introduction**. 2009. Disponível em: <<http://www.agile-process.org/>>. Acesso em: 13 dez. 2014.

ABSTRACT

The fastest growing companies nowadays are technology information's *startups*. The following successful cases are cited as examples: Paycom Software, NetEase, Facebook among others. One of crucial factors for a startup become a big company is how it is able to manage projects. This article aims to compare the current agile frameworks with methodologies called "orthodoxy" focusing on what would be a more appropriate approach to a reality of a software development corporation. For this, the descriptive and explanatory methodology was used through bibliographic research. As a result, traditional methodologies are very "embedded", based on complex process and have a lot of artifacts. Already, agile methodologies focus on the stakeholder interaction, empowerment of work teams and more flexible management. In view of this, the conclusions of this work are agile methodologies are the best options to take these organizations to success in a fast and consistent way.

Keywords: Management. Development. Methodology. Agile. Startup.