Metodologias Ágeis: Conceitos e Práticas no Gerenciamento de Projetos de Software

Ana Liddy Cenni de Castro Magalhães[1], Ivo Márcio Michalick Vasconcelos[2]

[1,2]FITec Inovações Tecnológicas
Av. Afonso Pena, 4001 11º andar – 30.130-008 – Belo Horizonte – MG – Brasil

analiddy,imichalick@fitec.org.br

Abstract. *Advantages and disadvantages of using the agile paradigm in software development have been largely discussed. One of the main disadvantages pointed out has been how project management should be performed, which can lead to a chaotic development from a managerial point of view. This work presents and compares values, principles and practices established for agile software development and management, regarding processes and practices described by the traditional approach for project planning, monitoring and control, aiming at increasing the credibility of the agile paradigm and encouraging the adoption of an agile culture in organizations.*

Resumo. *As vantagens e desvantagens da adoção do paradigma ágil de desenvolvimento de software têm sido largamente debatidas. Uma das principais desvantagens abordadas tem sido a indefinição de como deve ser conduzida a gerência de projetos, o que poderia levar a um desenvolvimento caótico sob o ponto de vista gerencial. Este trabalho apresenta e discute valores, princípios e práticas estabelecidos para as metodologias ágeis de desenvolvimento e gerenciamento de software em relação aos processos e práticas descritos pela abordagem tradicional de planejamento, monitoramento e controle de projetos, visando aumentar a credibilidade deste paradigma e incentivar a adoção de uma cultura ágil nas organizações.*

1. Introdução

Ao longo do tempo, o desenvolvimento de software tem sido rotulado como uma atividade caótica e marcada por fracassos: prazos e orçamentos não cumpridos, expectativas não satisfeitas, retorno de investimento muito menor que o esperado. Uma das principais razões apontadas para isto tem sido a dificuldade de se satisfazer, ao mesmo tempo, custo, prazo, escopo e qualidade. Em busca por uma solução, procurou-se empregar nos projetos de software a mesma abordagem dos projetos de engenharia, acreditando-se que a receita para o sucesso seria investir muito tempo e recursos em uma fase detalhada de planejamento e *design*, e garantir o sucesso da execução com gerenciamento e processos bem definidos. Nesta direção, diversas organizações imaturas e despreparadas partiram em busca de uma solução final para seus problemas: utilizando normas e modelos existentes, definiram processos organizacionais e técnicos complexos, suportados por muita documentação escrita e contratos muitas vezes confusos, que embasavam negociações posteriores complicadas e dispensavam a participação do cliente no decorrer do desenvolvimento. Insistindo em erros comuns de muitos projetos de engenharia e sem ter a coragem necessária para mudar, muitas organizações acabaram por ampliar o tamanho do problema com tentativas de solução.

Por mais que processos e controles tenham sido definidos, os resultados acabavam ficando longe da expectativa, pois a construção de software é diferente de outras construções da engenharia tradicional: um software é, pela sua própria natureza, intangível; é impossível se antever todas as suas funcionalidades; as necessidades emergem durante todo o seu desenvolvimento, e vão amadurecendo até a sua implantação; além disso, a própria utilização do software é que geralmente impulsiona o aprimoramento de seus recursos. A mudança é, portanto, inevitável, e o não reconhecimento desta realidade leva ao desperdício de recursos em planejamento e *design* desnecessários, que acabam sendo descartados posteriormente.

Ao procurar satisfazer custo, prazo, escopo e qualidade, o caminho adotado pela gerência tradicional de projetos tem sido partir de um escopo fechado, a partir do qual define-se custo e prazo e, em função do andamento do projeto, manipula-se a qualidade. Mas será que o melhor não seria ter um prazo predefinido, um custo fixo (definido em função do prazo), manter os níveis de qualidade e manipular o escopo? Por que não fazer o mais simples primeiro, e refinar depois, mudando somente e quando for necessário? Deixar de gerar documentação desnecessária para a execução do escopo? É importante notar que clientes pagam por software, não por diagramas e documentação, e no término do projeto é a qualidade final do produto que fará a grande diferença.

Em meio a tantos questionamentos, este trabalho discute os princípios e práticas das metodologias ágeis de desenvolvimento e gerenciamento de software em relação aos processos e práticas descritos pela abordagem tradicional de planejamento, monitoramento e controle de projetos, buscando mostrar que a adoção da abordagem ágil pode não só aumentar a credibilidade deste paradigma, mas também ajudar a elevar a qualidade dos produtos em conseqüência da melhoria da qualidade derivada das "boas práticas" empregadas.

2. Princípios e Práticas da Abordagem Ágil

A dificuldade de uso das metodologias de desenvolvimento de software tradicionais e a alta freqüência com que os projetos de software deixavam de cumprir seus cronogramas e extrapolavam seus orçamentos levaram, ao final da década de 1990, à elaboração do Manifesto pela Agilidade [Beck 2001] e à criação de uma organização sem fins lucrativos denominada Aliança Ágil [Agile Alliance 2005], com a finalidade de divulgar seus princípios [Beck 2001a] – apresentados na Tabela 1 – em busca de uma forma mais objetiva de se desenvolver software.

Surgiu assim um novo paradigma para o desenvolvimento de software – as metodologias "leves" (*lightweight methodologies*), ou ágeis. Ao invés de estabelecer processos, as metodologias ágeis combinam um pequeno número de regras e práticas, sendo especialmente adequadas para projetos pequenos ou médios (2-10 pessoas), com requisitos imprecisos ou em constante mudança. Entre as metodologias de desenvolvimento ágil destaca-se a *Extreme Programming* (XP). Dando suporte ao desenvolvimento, a *Extreme Project Management* (XPM) estabelece regras e ferramentas voltadas para o gerenciamento de projetos e a *Agile Project Management* (APM) define práticas de gerenciamento de projetos baseadas em sistemas adaptativos complexos.

Tabela 1. Os doze princípios da agilidade

	Descrição – Princípios da Agilidade
1	A prioridade é a satisfação do cliente, por meio da liberação rápida e contínua de software de valor.
2	Receba bem as mudanças de requisitos, mesmo em estágios tardios do desenvolvimento. Processos ágeis devem admitir mudanças que trazem vantagens competitivas para o cliente.
3	Libere software freqüentemente, dando preferência para uma escala de tempo mais curta.
4	Mantenha *stakeholders* e desenvolvedores trabalhando juntos a maior parte do tempo do projeto.
5	Construa projetos com indivíduos motivados, dê a eles o ambiente e suporte que precisam e confie neles.
6	O método mais eficiente e efetivo para repassar informação é pela comunicação direta.
7	Software funcionando é a principal medida de progresso de um projeto de software.
8	Processos ágeis promovem desenvolvimento sustentado. Assim, patrocinadores, desenvolvedores e usuários devem ser capazes de manter conversação pacífica indefinidamente.
9	A atenção contínua para a excelência técnica e um bom projeto (*design*) aprimoram a agilidade.
10	Simplicidade – a arte de maximizar a quantidade de trabalho não feito – é essencial, devendo ser sempre assumida em todos os aspectos do projeto.
11	As melhores arquiteturas, requisitos e projetos emergem de equipes auto-organizadas.
12	Em intervalos regulares, as equipes devem refletir sobre como se tornarem mais efetivas, para então refinarem e ajustarem seu comportamento.

2.1. Abordagem Ágil para Desenvolvimento de Software

A abordagem ágil propõe a construção de software de forma evolutiva e adaptativa. A idéia é começar da forma mais simples possível – apenas com o planejamento e *design* necessários – e resolver as necessidades mais claras e críticas, agregando valor ao produto e entregando algum resultado rapidamente. Durante todo o desenvolvimento, o objetivo é ter um software em operação o mais rápido possível, para que ele tenha condições de evoluir. Deve-se investir ao máximo em simplicidade, de forma que a mudança deixe de ser traumática e passe a ser natural.

2.1.1. *Extreme Programming*

A XP – *Extreme Programming* é a metodologia de desenvolvimento ágil mais conhecida e utilizada. De domínio público, possui este nome porque emprega "ao extremo" algumas "boas práticas" da engenharia de software. Ela é descrita por meio de valores, princípios e práticas [Beck 1999, 2001a] apresentados na Tabela 2. Os valores descrevem os objetivos de longo prazo ao aplicar XP e definem critérios de sucesso. Os princípios proporcionam a pequenas e médias equipes um ambiente de desenvolvimento cooperativo, capaz de atingir altos níveis de produtividade e um elevado grau de confiança. As práticas estruturam as atividades básicas do desenvolvimento XP e seguem seus princípios, sustentando os valores definidos. Os papéis e responsabilidades definidos nesta metodologia são apresentados na Tabela 3.

O ciclo de vida de um projeto XP é iterativo e incremental, como apresentado na Figura 1. Existe uma etapa de planejamento inicial, na qual o *coach* e o cliente analisam a viabilidade do projeto, identificam um escopo geral – as macro funcionalidades da solução – e elaboram o plano inicial, denominado *Big Plan*. O desenvolvimento ocorre em *releases*, que são divididas em iterações. No planejamento da próxima *release*, equipe técnica e cliente detalham requisitos e estimam esforço, utilizando *Story Cards* – cartões que descrevem estórias (requisitos de usuário) de forma sucinta, possibilitando estimar o esforço envolvido. Os *Story Cards* constituem uma ferramenta efetiva para guiar o desenvolvimento, sendo de fácil manipulação e armazenamento. As estimativas são realizadas em grupo e baseadas em histórico: atribui-se pontos às estórias mais simples e, a partir daí, estima-se de forma comparativa. Quando não se tem idéia da

complexidade de uma estória, alguns experimentos ou implementações são realizados, em busca de mais dados para realizar a estimativa. A unidade utilizada é *ideal weeks*. Após estimar todos os *Story Cards*, cliente e equipe técnica definem o escopo da próxima *release*, que deverá entregar um software de valor, de forma que usuários possam utilizá-lo, perceber benefícios de seu uso e realimentar os desenvolvedores com pontos positivos e negativos. O planejamento das iterações é então realizado, refinando o plano da *release*. Os desenvolvedores identificam as tarefas contidas nas estórias, estimam sua duração em *perfect days* e escolhem o que irão implementar – o que gera maior comprometimento e motivação. Durante todo o desenvolvimento é aplicada, constantemente, a prática de "jogo do planejamento".

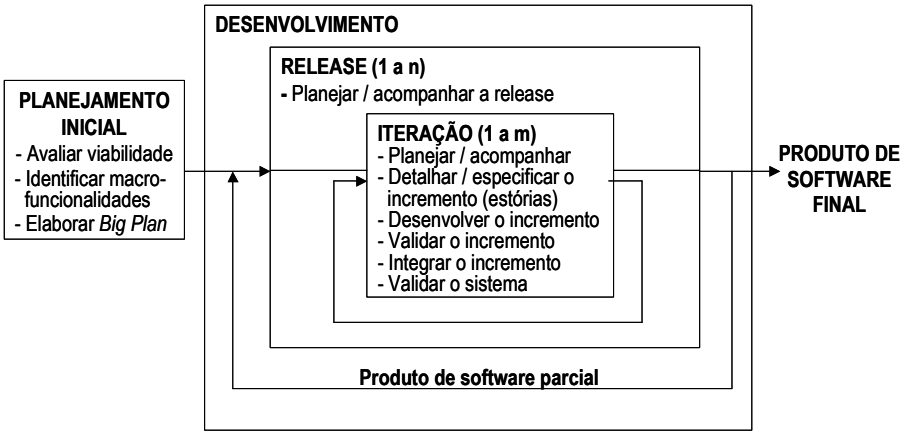

Figura 1. Ciclo de vida típico de um projeto XP

O desenvolvimento propriamente dito corresponde à implementação das estórias selecionadas. A arquitetura é definida e remodelada em seções rápidas de *design*, sempre buscando a simplicidade – deve-se resolver o problema atual, usando *metáforas*, e gerar código preparado para mudar e evoluir. A programação em pares é uma constante: o *driver* fica no teclado, focado no que está sendo implementado, enquanto o *partner* atua como inspetor, acompanhando a produção do *driver*, com trocas ocorrendo várias vezes ao dia. Desenvolvido utilizando um padrão de codificação, o código gerado é comunitário, sendo freqüentemente reestruturado em busca de um melhor desempenho e simplicidade. Testes unitários automáticos são gerados junto com o código, o que motiva e dá mais confiança à equipe. *Builds* são gerados a cada integração do código, ocasião em que todos os testes são refeitos. Se ocorrer erro, qualquer integrante da equipe poderá alterá-lo. Os testes de aceitação são especificados pelo cliente, que define com o apoio dos *testers* como quer o produto. As práticas XP utilizadas em conjunto promovem um processo eficiente de desenvolvimento do software.

O *tracker* é o responsável por acompanhar o progresso da iteração, verificando periodicamente com o programador quantos *perfect days* este já trabalhou na tarefa e quantos ainda faltam para concluí-la. Reuniões diárias de acompanhamento (*stand-up meetings*) são realizadas com o objetivo de verificar a evolução, comunicar problemas e providenciar soluções. Caso exista algum desvio ou problema, o *coach* é alertado para intervir junto à equipe e envolver o cliente, se necessário. Gráficos visíveis são disponibilizados para acompanhar progresso do projeto, apresentando em geral estimativas, testes executados, densidade de *bugs* e progresso de estórias.

Tabela 2. Valores, princípios e práticas da *Extreme Programming*

Valores		
Comunicação	Fundamental para a compreensão do trabalho a ser feito e entrosamento da equipe. A falta de comunicação é tida como um dos motivos para o insucesso de alguns projetos de software.	
Simplicidade	Deve-se procurar desenvolver a solução mais simples e que funcione, pensando apenas no presente. É preferível gastar um pouco mais para alterar depois, se necessário, do que fazer algo complicado e não utilizar.	
Realimentação (*feedback*)	Deve ser alta e ocorrer o mais rápido possível. É muito importante ter uma idéia clara sobre o andamento e a qualidade do software. Quanto maior a realimentação, melhor a comunicação e o aprendizado sobre o sistema.	
Coragem	Uma virtude que os praticantes da metodologia devem ter. Algo que não esteja funcionando adequadamente deve ser descartado e refeito, de forma mais simples.	
Princípios		
Realimentar rapidamente	Ensinar aprendendo	Não gerar sobrecarga desnecessária
Assumir a simplicidade	Aceitar responsabilidades	Investir pouco no início
Aceitar a mudança	Fazer experimentos concretos	Medir honestamente
Fazer mudanças incrementais	Jogar para vencer	Comunicar de forma aberta e honesta
Executar o trabalho com qualidade	Adaptar-se à cultura e condição local	Trabalhar a favor do instinto humano
Práticas		
Cliente sempre disponível	O cliente deve não só auxiliar, mas fazer parte da equipe, resolvendo dúvidas, definindo prioridades e alterando o escopo de iterações	
Uso de metáforas	Metáforas facilitam a comunicação sobre características do software, tanto com o código quanto com o processo ou domínio do negócio a ser desenvolvido, em busca de uma visão comum	
Jogo do Planejamento	O planejamento deve ser avaliado constantemente, envolvendo o cliente e a equipe técnica, visando colocar em produção as funcionalidades de maior valor possível no decorrer do projeto	
Versões em ciclos pequenos	O software deve ser desenvolvido e entregue em pequenas versões (*releases*), garantindo uma evolução constante do produto e um retorno rápido do investimento, além de minimizar riscos	
Desenvolvimento baseado em testes	Testes são definidos juntamente com o usuário. A seguir é realizada a implementação, e por último trabalha-se a arquitetura do software	
Integração contínua	A integração dos módulos do software deve ocorrer diversas vezes por dia, após aprovação completa de cada teste unitário	
Arquitetura simples	Deve-se simplificar a arquitetura, de forma que o código possa, a qualquer momento e na forma mais simples possível, passar por todos os testes	
Reestruturação constante	A cada nova funcionalidade adicionada, a estrutura do código deve ser retrabalhada, até ficar na sua forma mais simples	
Programação em pares	Todo o código deve ser produzido em duplas que trabalham em um mesmo teclado, *mouse* e monitor. Os membros das duplas de programação se revezam, em média, a cada duas horas	
Propriedade coletiva	A equipe toda é responsável por cada arquivo de código, não sendo necessário pedir autorização para alterar qualquer arquivo	
Padrão para codificação	Todo o código é desenvolvido segundo um padrão predefinido, utilizado por todos os integrantes da equipe	
40 horas de trabalho semanal	Cada membro deverá ter uma carga máxima de trabalho semanal de 40 horas, uma vez que trabalhar por longos períodos é contraproducente	

Tabela 3. Papéis e responsabilidades da *Extreme Programming*

Papel	**Responsabilidades**
Programador	Projetar, codificar, implementar testes e estimar tarefas – é o "coração" da XP
Cliente	Estabelecer prioridades e escopo, escrever estórias e testes funcionais, esclarecer dúvidas
Testador (*tester*)	Apoiar o cliente na escolha e definição de testes funcionais, além de assegurar a execução destes testes e relatar problemas por eles identificados
Acompanhador (*tracker*)	Reportar as métricas do projeto, promovendo visibilidade sobre a precisão das estimativas e progresso do projeto – é a "consciência da equipe"
Técnico (*coach*)	Identificar problemas e resolvê-los para que a equipe possa trabalhar da melhor forma. Não requer conhecimento técnico profundo – papel semelhante a um gerente de projeto de software

2.2. A Abordagem Ágil para o Gerenciamento de Projetos de Software

A principal característica da abordagem ágil de gerenciamento de projetos está na sua atitude em relação à mudança: diferentemente da abordagem tradicional, na qual o planejamento direciona os resultados, na abordagem ágil os resultados é que direcionam o planejamento, sendo necessário facilitar a mudança, e não desencorajá-la por meio de processos complexos, que restringem e diminuem a criatividade.

2.2.1. *Extreme Project Management*

A XPM (*Extreme Project Management*) visa melhorar o gerenciamento de projetos desenvolvidos segundo o paradigma ágil, com ênfase no XP, e em especial os *e-projects* – projetos de software para os quais o tempo e o custo para tornar o produto disponível no mercado são críticos, não sendo possível elaborar um cronograma detalhado e uma especificação de requisitos em um estágio preliminar do processo, e mostra-se necessária uma avaliação diária do projeto para adequá-lo à situação de mercado [Jeffries 2004]. A meta é entregar o resultado desejado, e não necessariamente o resultado inicialmente planejado. Ela é definida por um conjunto de onze regras e cinco ferramentas, descritas a seguir, que possuem como missão dar suporte à mudança, planejando critérios de sucesso para os *stakeholders*, descrevendo eventos e cenários principais, definindo benefícios esperados e como atingi-los e estabelecendo acordos com parceiros e em relação à qualidade exigida [Thomsett 2005].

Regra 1: A gerência de pessoas e processos criativos demanda processos de gerenciamento criativos

Não existe uma forma única e ideal para gerenciar projetos na dinâmica atual do mercado. Tanto gerente quanto equipe precisam ser criativos no desenvolvimento de um produto inovador, com alto valor para o negócio e maior qualidade. É necessário considerar não só as expectativas e os requisitos dos clientes, mas também o contexto e as estratégias da própria organização, em busca de um planejamento, monitoramento e controle mais tangível do projeto.

Regra 2: Quanto menos o gerente souber sobre questões técnicas do projeto, melhor

O avanço da tecnologia e da complexidade das soluções técnicas incitou a condução gerencial e técnica de um projeto por profissionais distintos, em função do conjunto de habilidades requeridas. Enquanto o gerenciamento técnico compreende o domínio de tecnologias para análise, modelagem, desenho, programação e teste, o gerenciamento do projeto busca controlar o processo – custos, riscos, estimativas, cronogramas – bem como integrar o pessoal qualificado exigido para dar a contribuição intelectual específica ao produto. Os aspectos gerenciais e técnicos são integrados pelo escopo, objetivo, estratégias e requisitos de qualidade. Mudar um gerente técnico para a posição de um gerente de projeto significa retirar um especialista de sua zona de domínio, pois experiências específicas devem ser usadas em áreas específicas.

Regra 3: O que ocorre depois do projeto é mais importante do que o que ocorre durante o projeto

Em muitos projetos, os mecanismos de rastreamento e informação são interrompidos logo após a implantação, deixando os encarregados do negócio com uma evidência muito pequena do valor agregado pelo produto gerado, o que faz com que a área de TI pareça gerar um custo enorme à empresa. Um acompanhamento posterior permitiria verificar se o projeto realmente cumpriu suas metas de sucesso e gerou retorno para o investimento.

Regra 4: Um planejamento de projeto desenvolvido sem a participação completa dos *stakeholders*[1] não é mais que a fantasia de uma única pessoa

O gerente de projeto deve facilitar o processo de planejamento, envolvendo os *stakeholders* fundamentais e empreendendo o planejamento de maneira aberta e colaborativa. O XPM usa o Planejamento Rápido (RAP) para produzir planos de projeto – um conceito semelhante ao Desenvolvimento Rápido de Aplicação (RAD). O RAP assegura uma fundamentação sólida para o projeto, uma vez que engloba um processo intensivo e participativo, no qual pontos de discórdia entre *stakeholders* são discutidos na busca do consenso. Devido à impossibilidade de se prever o futuro, o plano de projeto deve ser dividido em porções maiores e estar baseado em dados realistas, obtidos de pessoas responsáveis por tarefas específicas.

Regra 5: Quanto mais tempo o gerente de projeto permanecer com os *stakeholders*, melhor

O XPM está mais relacionado ao contexto do projeto – com a informação gerencial do negócio – do que com o seu conteúdo – especificações técnicas e entregas. O gerente deve investir boa parte do seu tempo em entender melhor as pessoas envolvidas no projeto. Os líderes empresariais empregam a compreensão humana como forma de obter relação longa e duradoura com a equipe de projeto e os *stakeholders*.

Regra 6: Se o sucesso de projeto não foi definido no começo, ele nunca será alcançado no final

O sucesso em um projeto está geralmente associado a: 1) satisfazer os *stakeholders*; 2) atingir as exigências de escopo; 3) permanecer dentro do orçamento e dos prazos estabelecidos; 4) agregar valor ao negócio; 5) assegurar uma boa qualidade ao produto; 6) deixar os membros da equipe satisfeitos. Os critérios de sucesso devem ser definidos logo de início e acompanhados durante todo o desenvolvimento. A ferramenta sugerida para o acompanhamento é composta por *sliders* de sucesso, um indicador que representa o grau de atendimento aos critérios de sucesso.

Regra 7: Mostre-lhes o lucro: nada mais importa

Mais do que facilitadora de processos, a força atual da TI reside em tornar o negócio mais lucrativo e competitivo. Uma análise cuidadosa do negócio permitirá identificar o valor agregado pelo produto e priorizar funcionalidades a serem desenvolvidas, visando empregar menos esforço na obtenção de mais benefícios. As ferramentas utilizadas são:

- O modelo O3 (*Objective, Output, Outcome*), que cria uma cadeia de valores para o projeto, permitindo modelar e perceber os benefícios associados aos objetivos. Ele parte do princípio de que a organização só será capaz de alcançar seus resultados se o projeto tiver sucesso na produção de resultados relevantes. Ele está baseado no modelo IRACIS (*Increase Revenue, Avoid Costs and Improve Services*) [Thomsett 2002], que possibilita identificar se os benefícios propostos são realistas ou não.
- Um Acordo de Qualidade que, a partir dos objetivos do produto, estabelece os critérios para definir sua qualidade e os processos presentes no desenvolvimento que irão assegurá-la, uma vez que a melhoria de um atributo de qualidade pode causar impacto negativo em outro.

Regra 8: Os *stakeholders* do projeto podem ser seus melhores aliados ou seus piores inimigos

---

[1] *Stakeholders* são os indivíduos ou as organizações que estão ativamente envolvidos em um projeto cujos interesses podem afetar positivamente ou negativamente o resultado da execução do projeto

Existe risco de um *stakeholder* (como um fornecedor de recursos) ser deslocado para outro projeto, o que geralmente causa impacto negativo. Na tentativa de evitar tais situações, o XPM sugere ao gerente a utilização de Acordos de Parceria, uma ferramenta que documenta os serviços requeridos, o tempo e o custo estimado para sua realização e a identificação de pessoas em condições de fornecer um retorno sobre os serviços para o *stakeholder*. Manter os *stakeholders* informados sobre os resultados ajuda a manter uma boa relação no projeto.

Regra 9: Se você não pode prever o futuro, não planeje em detalhe

O XPM abrange planejamento e re-planejamento diários, como parte do processo da equipe e dos *stakeholders*. Todas as alterações relativas a contexto, externas e internas – riscos, escopo, objetivos, valor agregado, etc. – são identificadas e avaliadas diariamente, o que se ajusta extremamente bem aos conceitos de XP. Na sessão de RAP os principais eventos e cenários relacionados à entrega do produto são identificados, e somente tarefas que visem atingir um evento específico são detalhadas e incluídas no cronograma. A ferramenta de planejamento em tempo real de eventos e cenário é uma extensão simples das práticas de desenvolvimento utilizadas em XP.

Regra 10: Se o seu projeto não mudou, fique apreensivo, muito apreensivo

Gerente e equipe devem reunir-se diariamente para avaliar se houve alteração em expectativas de sucesso, escopo, objetivos, riscos, qualidade, *stakeholders* ou projetos relacionados, bem como verificar se suposições referentes a custo e benefício continuam pertinentes.

Regra 11: Em *e-projects*, um dia é um tempo muito longo

O gerenciamento efetivo de *e-projects* demanda uma abordagem nova e radical para o gerenciamento de projetos.

2.2.2. *Agile Project Management*

Elaborada a partir de princípios que exploram a compreensão do comportamento humano autônomo, a abordagem APM (*Agile Project Management*) inclui em suas suposições e práticas para o gerenciamento de projetos a noção de sistemas adaptativos complexos (*Complex Adaptive Systems* – CAS) [Kelly 1994], embasada em um estudo dos sistemas vivos existentes na natureza – como revoadas, cardumes e enxames. É interessante notar que, apesar destes sistemas vivos possuírem somente regras e capacidades estratégicas locais, seu comportamento coletivo é caracterizado por uma superposição de ordem, auto-organização e uma inteligência coletiva que é maior que a soma das partes, além de exibirem uma habilidade notável para se adaptarem a ambientes complexos e dinâmicos.

Em uma equipe ágil, o gerente de projeto também precisa de um conjunto de práticas simples que o guie e forneça um *framework* dentro do qual possa atuar, e não de um conjunto de instruções rígidas. Seguindo estas práticas, o gerente torna-se um líder com capacidade de adaptação, capaz de fixar uma direção, estabelecer regras simples e geradoras do sistema, bem como encorajar uma constante avaliação (*feedback*), adaptação e colaboração.

Segundo seus autores, a APM coloca em destaque as habilidades de liderança, combinando visão de negócio, habilidades de comunicação, gerência flexível e compreensão técnica com a habilidade para planejar, coordenar e executar [Augustine

2005]. No desenvolvimento ágil há pouca necessidade de um gerente de projeto atuando de forma tradicional, uma vez que o cliente define e prioriza as funcionalidades considerando seu valor para o negócio, as equipes criam e monitoram seu próprio planejamento de iterações junto com o cliente, e os desenvolvedores dividem as tarefas entre si, enquanto trabalham e avaliam o progresso obtido em cada iteração, ajustando o planejamento com o cliente, quando necessário. As metodologias ágeis livram o gerente de projeto da atividade de ser um mestre das tarefas, permitindo-lhe atuar como líder.

O *framework* para gerência de projeto ágil baseado em CAS é composto pelas seis práticas-chave detalhadas a seguir. Aplicadas em conjunto, ajudam a administrar equipes de desenvolvimento como sistemas adaptáveis complexos, ao mesmo tempo em que proporcionam liberdade para sobrepor estilos próprios de liderança pessoal.

Prática 1: Visão Direcionadora – Estabeleça uma visão direcionadora para o projeto e reforce-a continuamente, por meio de palavras e ações

Ao coordenar uma equipe, o gerente ágil deve influenciar continuamente seu comportamento, definindo, disseminando e sustentando uma visão direcionada. No início do projeto o gerente deve trabalhar juntamente com o cliente para entender sua visão do projeto, ou seja, como é esperado que o projeto atenda às metas de negócio e como o produto resultante deverá ser usado. O gerente deverá promover a propriedade coletiva desta visão, facilitando uma discussão conjunta da equipe para construir uma visão comum do projeto. Uma definição consistente da visão ajudará a equipe a manter o foco e a tomar decisões difíceis em relação a valor e prioridade para o negócio. Ao invés do processo tradicional de dividir as tarefas do projeto em componentes cada vez menores para atribuição e rastreamento freqüentes, é preferível manter o foco no conjunto e promover um processo de planejamento que mantenha as tarefas em um nível de detalhe que defina a intenção e o resultado desejados, preservando a flexibilidade para a inovação e a autonomia da equipe.

Prática 2: Trabalho e Colaboração em Equipe – Facilite a colaboração e o trabalho em equipe reforçando relacionamentos

Quando o trabalho conjunto aprimora forças individuais complementares, os resultados obtidos podem ser excepcionais. Conseguir que as pessoas trabalhem de forma colaborativa, porém, pode ser um desafio e não pode acontecer por imposição. O papel do gerente de projeto é facilitar a colaboração ativa da equipe e estabelecer condições para se manterem bons relacionamentos. A definição de papéis na equipe, o conhecimento das características, habilidades complementares e interesses de cada membro, aliadas a um relacionamento próximo, cordial e respeitoso entre todos proporcionam interações ricas e aprimoram o trabalho colaborativo. Um espaço físico adequado também é importante, com áreas individuais e comuns, organizado de forma a facilitar atividades colaborativas – como programação em pares e resolução de problemas pela equipe – ao mesmo tempo em que possibilita manter a privacidade.

As sessões de planejamento são situações férteis para desenvolver uma compreensão e respeito comum entre os desenvolvedores e o cliente. À medida que o projeto progride, e com uma liderança adequada, estas sessões podem se tornar altamente colaborativas e criativas, resultando em melhoria do moral da equipe e em um produto melhor. Técnicas de facilitação básicas, como garantir que todas as partes tenham oportunidade para falar, enquanto outros opinam, confirmam pontos de vista e sanam preocupações, ajudam a construir a equipe. Por outro lado, existem muitas

situações que impedem a colaboração, como tratamento desrespeitoso, egoísmo e membros que não apresentam desempenho satisfatório. Cabe ao gerente monitorar a dinâmica da equipe e decidir quando intervir, não se esquecendo de celebrar sucessos e marcos conquistados. Valorizar a autonomia da equipe é fundamental para todas as outras práticas.

Prática 3: Regras Simples – Estabeleça e apóie um conjunto de práticas-chave (guias)

Em uma revoada, os pássaros seguem regras simples – como evitar objetos, manter o ritmo e ficar próximo de outros pássaros – mas exibem um comportamento complexo, coletivo, voando em formação por distâncias longas, e adaptando-se no percurso a condições variáveis. Seguindo esta mesma idéia, regras simples como as práticas XP podem fornecer suporte para um comportamento complexo, permitindo à equipe trabalhar dentro de uma estrutura flexível. Essas regras devem ser suficientes para fornecer limites claros, mas não complexas em demasia, para não restringir a autonomia e criatividade de cada membro da equipe. Para isso, as práticas devem ser declaradas explicitamente, compreendidas e acordadas por todos os membros da equipe logo no início do projeto – embora a equipe deva ter a habilidade para modificar práticas que não estão funcionando como esperado ou adicionar novas práticas. Ao longo do projeto, o gerente tem como papel principal avaliar junto à equipe se as práticas estão sendo seguidas e se os membros possuem dúvidas ou estão inseguros na sua utilização, sempre atento a oportunidades para encorajar o uso das práticas estabelecidas ou aperfeiçoá-las.

Prática 4: Informação Aberta – Forneça acesso aberto à informação

Para que uma equipe ágil possa adaptar-se, a informação deve estar disponível de forma aberta e livre. Durante muito tempo, gerentes tradicionais evitavam esta liberdade com medo de resultar em caos, controlando a informação e repassando somente aquilo que os membros "precisavam saber". Desta forma, as equipes freqüentemente ficavam alheias à situação do projeto – só o gerente possuía o "plano mestre" e só ele interagia com o patrocinador do projeto. Na abordagem ágil, a informação é liberada e acaba possuindo um poder maior. As práticas de XP, por exemplo, promovem acesso aberto à informação – a propriedade coletiva dos cartões com as estórias, da documentação gerada e do código encorajam a equipe a contribuir para o projeto, enquanto a prática cliente no local promove uma troca aberta de informação.

Uma variedade de técnicas podem ser utilizadas para promover a informação aberta: posicionamento próximo dos membros; uso de "radiadores" para disseminar informação, como quadros e *flipcharts*; uso de um *wiki* [Wiki 2005] (um *site* na web, gratuito, desenvolvido e mantido pelos usuários) para compartilhar informação; reuniões diárias rápidas para promover o fluxo e troca de informação; troca de informação entre equipe, área de negócios e direção; encontros informais com o patrocinador no local de desenvolvimento para obtenção de relatos públicos e demonstrações do que está sendo efetivamente feito.

Prática 5: Toque leve – Aplique somente o controle suficiente para manter a ordem emergente

O controle inteligente de equipes requer uma mistura delicada de ordem imposta e emergente. No gerenciamento tradicional tudo é visto sob o prisma de controle: controle de mudança, de risco, de pessoas, etc. Diversas metodologias sofisticadas, ferramentas e práticas evoluíram visando "administrar um mundo fora de controle", acreditando que mais controle resultaria em mais ordem. Boa parte delas, porém, não manipula facilmente processos cíclicos, atualizações freqüentes e situações variáveis presentes em um mundo real e incerto. No zelo de impor um controle cada vez maior, alguns gerentes parecem se esquecer de que o propósito original de controle é apenas criar ordem e facilitar a execução dos trabalhos do projeto.

Ao reconhecer que é impossível saber tudo com antecedência, o gerente ágil renuncia a parte do controle e aborda o gerenciamento com coragem, aplicando somente o controle suficiente para manter a ordem. Assim, sincronizados com a visão e baseados nas funcionalidades a serem entregues, os planos de projeto são elaborados em um nível bastante alto para fornecer espaço à equipe para inovação, criatividade e resposta rápida para ambientes dinâmicos. A equipe deve ter autonomia suficiente para se auto-organizar, ou seja, para adaptar-se rapidamente a situações e adotar soluções alternativas, quando necessário. À visão convencional, esta ordem emergente pode parecer-se com uma desordem ou caos. Mas para um gerente corajoso, disposto a renunciar a um pouco deste controle, as recompensas desta prática envolvem uma equipe dinâmica e comprometida, soluções inovadoras e possibilidade de adaptação contínua.

Prática 6: Vigilância Ágil – Aplique um contínuo monitoramento, aprendizado e adaptação ao ambiente

O encadeamento entre todas as práticas está na vigilância ágil. O trabalho mais criativo e ágil de uma equipe acontece no limiar entre a ordem e o caos – imprevisível o bastante para ser interessante e ordenado o suficiente para não entrar em caos. A operação nesta extremidade requer, porém, aprendizagem e adaptação contínuas para intervir nas condições ambientais. Conduzir uma equipe estabelecendo uma visão direcionada, estimulando o trabalho e a cooperação, estabelecendo regras simples, patrocinando informação aberta e administrando com um toque leve, tudo isso requer do gerente ágil uma vigilância contínua. Isto significa confiar nas pessoas e no processo, ser observador, manter o pulso da equipe, realizar avaliação contínua, monitorar o sucesso ou o fracasso e adaptar-se conforme a necessidade. A vigilância ágil consiste em: examinar as decisões de projeto para verificar seu alinhamento com a visão estabelecida; encorajar continuamente o trabalho e a colaboração em equipe, conversando com os membros, percebendo sinais de tensão e intervindo quando necessário; encorajar a auto-organização, desaprovando "panelinhas"; manter-se à frente da tecnologia para entender a linguagem da equipe; estabelecer regras simples, aproveitando toda oportunidade para refletir sobre o que funciona e o que precisa de melhoria; agir com coragem para fazer as mudanças quando perceber que são necessárias; demolir barreiras para compartilhamento da informação; intervir depressa, porém sabiamente, para resolver assuntos pessoais; motivar e recompensar iniciativas, administrando expectativas.

3. Princípios e Técnicas do Gerenciamento Tradicional de Projetos

Na busca pela qualidade de produtos de software e cumprimento de metas físicas e financeiras, tornou-se imperativa a necessidade de dominar os princípios e técnicas para a gestão eficiente de projetos. Nesta direção, muitos gerentes passaram a empregar o conhecimento disponibilizado no PMBOK [PMI 2004], um guia que descreve o conjunto de conhecimentos e as melhores práticas de gerenciamento de projetos. O processo de gerenciamento de grandes projetos, porém, é geralmente mais complexo e burocrático, devido à necessidade de formalizar ações, o que muitas vezes dificulta a adaptação destas técnicas para pequenas empresas, que necessitam um gerenciamento mais informal.

3.1. A Gerência Informal de Projetos

No início da década de 80, à medida que o gerenciamento de projetos tornava-se um processo reconhecido, muita documentação formal passou a ser produzida, visando principalmente tranqüilizar o cliente. Considerando o tempo consumido na elaboração, digitação, leitura, cópia, distribuição e arquivamento de documentos, essa formalização resultava em um custo muito alto. Mais grave do que isto, os gerentes de projeto acabaram ficando tão absorvidos com a documentação a ser gerada que pouco tempo lhes restava para gerenciar efetivamente o projeto.

Nos últimos 20 anos, porém, esse cenário mudou. Segundo Kerzner [Kerzner 2002], a mudança mais significativa no campo do gerenciamento de projetos foi a comprovação de que o gerenciamento informal dá resultados. Com o gerenciamento informal, a necessidade de documentação foi reduzida para níveis minimamente aceitáveis, e diretrizes formais foram substituídas por listas de verificação menos detalhadas e mais genéricas. Mudar da formalidade para a informalidade exige, porém, uma alteração na cultura da organização. Kerzner aponta quatro elementos chave para o sucesso da implementação da gestão informal de projetos:

- confiança: fundamental na consolidação de uma relação efetiva entre o fornecedor/ terceirizado e o cliente, a confiança traz inúmeros benefícios para ambas as partes. Sem ela, gerentes e responsáveis por projetos precisariam de uma vasta documentação, apenas para terem a certeza de que todos os encarregados estão cumprindo suas tarefas da maneira que lhes foi determinada.
- comunicação: em geral, embora os executivos prefiram comunicar-se verbal ou informalmente, existe a crença de que aquilo que não foi escrito não foi dito. Um dos pré-requisitos para o gerenciamento informal de projetos é que os funcionários entendam a estrutura, as funções e as responsabilidades que terão no âmbito da empresa e do projeto. Metodologias eficientes de gerenciamento de projetos promovem não apenas o gerenciamento informal, mas igualmente a comunicação eficiente, tanto lateral quanto verticalmente. Dois grandes obstáculos internos a serem superados para desenvolver-se uma cultura informal são os relatórios e as reuniões longas e desnecessárias, decorrentes da intervenção dos gerentes em atividades rotineiras ou do mal direcionamento de informações. Quando necessária, a comunicação formal deve ser breve e focada em três questões básicas: qual a situação atual, qual a situação desejada e se existe algum problema exigindo a interferência da administração. Nenhum planejamento, por melhor que seja, irá muito longe sem uma comunicação eficiente.

- cooperação: mais relacionada com a atitude em relação ao trabalho do que com o trabalho propriamente dito, consiste em ações voluntárias das pessoas para trabalhar em benefício do todo, buscando um resultado favorável. Sua efetivação não depende da intervenção formal da autoridade, pois os integrantes geralmente sabem o que devem fazer e o que fazem. As pessoas aprendem a cooperar à medida que se conhecem melhor, o que leva tempo – um recurso quase sempre escasso em projetos.
- Trabalho em equipe: desenvolvido por pessoas atuando juntas com um espírito de cooperação, sob os limites de uma coordenação, possibilitando: troca de idéias e informações por iniciativa própria; estabelecimento de altos índices de inovação e criatividade; confiança e lealdade entre os membros e para com a empresa; dedicação ao trabalho realizado e aos compromissos assumidos; franqueza e honestidade em seu relacionamento.

3.2. O Gerenciamento Tradicional de Projetos segundo o PMBOK

Mundialmente reconhecido e aceito desde 1999 como padrão de gerenciamento de projetos pelo ANSI (*American National Standards Institute*), o PMBOK [PMI 2004] descreve o conjunto de conhecimentos e as melhores práticas para o gerenciamento de projetos, podendo ser utilizado para orientar a definição de um processo padrão para gerenciamento de projetos de software, pois descreve "o que deve ser feito" e não "como fazer". Diversas empresas no mundo utilizam com sucesso o PMBOK como base para o estabelecimento de uma cultura em gerenciamento de projetos.

Os 44 processos que integram a terceira edição do PMBOK [PMI 2004] estão organizados em nove áreas de conhecimento ou de atuação gerencial: Integração, Escopo, Tempo, Custos, Qualidade, Recursos Humanos, Comunicações, Riscos e Aquisições. Tais processos encontram-se divididos em cinco grandes grupos: Iniciação (que autoriza o início do projeto ou de uma fase), Planejamento (que define, refina e seleciona as melhores alternativas), Execução (que coordena recursos a partir do que foi planejado), Controle (que monitora e mede o progresso do que foi executado) e Encerramento (formaliza o aceite e a finalização do projeto ou de uma fase). Detalhes sobre as áreas de conhecimento do PMBOK serão apresentados mais à frente, no quadro comparativo da Tabela 4.

4. Comparativo entre os Gerenciamentos de Projetos Tradicional e Ágil

O gerenciamento tradicional de projetos enfatiza um forte planejamento e muita disciplina no processo, partindo do princípio de que a aplicação de controle suficiente possibilita alcançar o objetivo planejado dentro de limites preestabelecidos de tempo, orçamento e escopo. Muito deste formalismo provém de necessidades inerentes a grandes projetos ou a restrições severas de confiabilidade, como ocorre em sistemas relacionados a atividades que apresentam risco de vida. Para projetos desta natureza, o formalismo da abordagem tradicional demonstra-se ainda adequado. Para a maioria dos projetos, porém, o gerenciamento tradicional pode acabar acrescentando somente custo e complexidade, enquanto fornece um falso senso de segurança, uma vez que o gerente está sempre ocupado – exaustivamente planejando, medindo e controlando. Muitas organizações desperdiçaram cifras enormes com planejamento prematuro, sem desenvolver de forma rápida e iterativa e sem envolver o cliente, fatores que são hoje considerados fundamentais para o sucesso de um projeto de software.

Como o PMBOK foi concebido para ser um guia genérico e completo para o gerenciamento tradicional de projetos, ele desconsidera as peculiaridades da execução de produtos e serviços de software. Sua aplicação prática na área de software exige um investimento importante na concepção de um processo adequado, que venha realmente alavancar o negócio e facilitar o trabalho das equipes de projeto, sem requerer trabalho adicional somente para atender às suas definições de processos.

Consideradas por alguns como um antídoto ao excesso de burocracia e uma solução para os problemas das pequenas equipes, as metodologias ágeis vêm se firmando e atraindo novos adeptos. Elas se apresentam como uma coleção de doutrinas boas, básicas e familiares, especialmente adequadas para projetos de natureza exploratória desenvolvidos por equipes pequenas.

Em essência, o ciclo de vida ágil é semelhante a qualquer outra abordagem de desenvolvimento: descobrir o que o cliente quer, certificar-se exatamente sobre o que ele quer, calcular o esforço, construir a solução e entregá-la ao cliente. O que muda na abordagem ágil, porém, é que o ciclo fica reduzido em algumas semanas para cada componente funcional desenvolvido, tornando-se muito mais rápido e sendo executado diversas vezes. Por este prisma, Ludwig considera que as áreas de conhecimento definidas no PMBOK aplicam-se ao desenvolvimento ágil por meio da "elaboração progressiva" [Ludwig 2005].

Em contraposição ao gerenciamento tradicional de projetos, que parte do princípio de que o planejamento direciona os resultados e que entregar o planejado é sinônimo de sucesso em um projeto, a abordagem ágil considera que os resultados é que devem direcionar o planejamento, e que o sucesso advém de entregar o resultado desejado – e não necessariamente o resultado planejado [Jacobsen 2005]. A grande diferença entre estas duas abordagens está na atitude em relação às mudanças no planejamento: a abordagem ágil considera necessário incentivar a mudança, e não desencorajá-la por meio da aplicação de procedimentos que restringem e diminuem a criatividade. A abordagem ágil esforça-se para reduzir o custo de mudança ao longo do processo de desenvolvimento de software. Como exemplo, o XP – visando entregar funcionalidades de maior valor para o negócio tão rápido quanto possível – usa ciclos de planejamento e desenvolvimento rápidos e iterativos, nos quais o "jogo do planejamento" atua como um processo evolutivo, continuamente refinado e ajustado. Além disso, testes sistemáticos e constantes, que são inerentes ao XP, asseguram mais qualidade pela descoberta e resolução antecipada de defeitos.

Existem algumas objeções importantes ao uso da abordagem ágil: a falta de documentação e a forma superficial com a qual questões gerenciais e organizacionais são tratadas – duas questões alarmantes do ponto de vista de um gerente de projetos. Uma vez que a abordagem ágil transmite muita confiança na comunicação oral, a documentação muitas vezes é relegada a segundo plano em prol do conhecimento implícito, tácito. É necessário buscar um ponto de equilíbrio, de forma a não gerar documentação além do necessário.

Como a abordagem ágil não impõe uma estrutura de gerenciamento no estilo "comando-e-controle", ela não enfatiza o papel do gerente, mas destaca a necessidade de líderes para resolução de problemas e atendimento das necessidades de clientes. Para a equipe, é preferível seguir um líder do que ser limitada por um gerente. A abordagem ágil introduz o papel de um orientador (*coach*), que facilita a comunicação e direciona a

equipe. Ao conduzir o projeto, ele verifica se os valores e práticas estão sendo adotados, se não existe conflito técnico ou pessoal na equipe, se há escassez ou excesso de recursos. Além de identificar as expectativas e requisitos do cliente, avaliar custo, benefício e risco necessários para atender aos requisitos, o planejamento deve também enfocar e entender o contexto organizacional no qual o projeto está sendo desenvolvido, incluindo cultura, política e diretrizes financeiras.

As atividades fundamentais de planejamento são necessárias, porém um plano de projeto muito detalhado é geralmente inadequado. Boehm explica que tanto a abordagem ágil quanto a dirigida por planejamento possuem aspectos positivos e negativos, e considera que a grande diferença entre as duas abordagens está no conjunto de pressupostos de cada uma. Também compara as duas em relação a uma série de fatores e discute formas de aproximá-las – usando o planejamento para aperfeiçoar métodos ágeis e a agilidade para dar maior eficiência aos métodos dirigidos pelo planejamento – e propõe métodos híbridos, que podem considerar os riscos do projeto como fator de ajuste na busca do ponto de equilíbrio entre elas [Boehm 2005].

Quanto ao Manifesto pela Agilidade, Boehm comenta que em cada par de alternativas uma pode reforçar a outra: processos e ferramentas podem melhor capacitar os indivíduos e interações; a documentação ajuda as pessoas a entenderem um software complexo; a negociação de contrato pode ser uma parte integrante da colaboração do cliente; seguir um plano pode ser o melhor modo para responder a uma mudança, quando esta é previsível. Neste contexto, as pessoas ativamente envolvidas e suas proposições de valor possuem grande importância.

O ponto-chave para amenizar a discordância entre o uso das abordagens tradicional e ágil está em considerar, na escolha da abordagem, as características do projeto a ser desenvolvido, buscando aplicar a metodologia correta para o trabalho a ser realizado. A dinâmica requerida em projetos de natureza exploratória e que primam pela inovação inviabiliza o uso da abordagem tradicional, pois o risco de ser necessário alterar um produto depois da conclusão de uma fase de seu ciclo de vida é bastante alto. Por outro lado, a aplicação da abordagem ágil para planejamento estratégico ou priorização de projetos não será eficiente, uma vez que uma das premissas principais da abordagem ágil é entregar incrementos de funcionalidade específicos em curto espaço de tempo.

A abordagem ágil é mais adequada para projetos definidos em alto nível, com objetivos claros e foco nos valores empresariais – principalmente em relação a custos e prazos –, necessitando grande participação do cliente durante todo o processo. Também é adequada para o detalhamento do trabalho tático necessário para completar um projeto, bem como para mudanças mínimas requeridas na manutenção de sistemas ou outros produtos de trabalho discretos.

A Tabela 4 apresenta uma comparação sucinta entre os processos definidos no PMBOK e a abordagem ágil de desenvolvimento de projetos. O objetivo principal da abordagem ágil é simplificar o processo de desenvolvimento, minimizando e dinamizando tarefas e artefatos que são importantes no processo, além do software funcionando.

Tabela 4. Gerenciamento de projetos: comparativo PMBOK x abordagem ágil

Área de Conhecimento	
Processos Definidos no PMBOK	**Princípios e Práticas da Abordagem Ágil**
Gerenciamento de Integração Identificar, definir, combinar, unificar e coordenar os diversos processos e atividades de gerenciamento de projetos	
Desenvolver o termo de abertura do projeto: gerar uma autorização formal de seu início Desenvolver a declaração de escopo preliminar do projeto: fornecer uma descrição de alto nível do escopo Desenvolver o plano de gerenciamento do projeto: definir, preparar, integrar e coordenar planos auxiliares Orientar e gerenciar a execução do projeto: executar o trabalho definido para atingir os requisitos definidos na declaração de escopo Monitorar e controlar o trabalho do projeto: atender aos objetivos de desempenho definidos no plano de projeto Controlar mudanças: aprovar solicitações e coordenar mudanças em produtos e processos, de forma integrada Encerrar o projeto: finalizar todas as atividades para encerrar formalmente o projeto	Big Plan: identifica a visão e missão do sistema, dá ao cliente e à equipe uma noção geral para as *releases* Planos de release e iteração: detalham etapas específicas e possibilitam o acompanhamento Jogo do planejamento: acompanha toda a execução, avaliando prazo, escopo, risco e custo, com pouca carga adicional, ainda que de maneira informal Cartões com estórias: indexam os casos de uso, que são selecionados pelo cliente para implementação a cada iteração Desenvolvimento por iteração: entrega um conjunto de funcionalidades, agregando valor ao produto e realimentando o planejamento Reuniões de acompanhamento: possibilitam identificar com rapidez a situação atual e a necessidade de mudança
Gerenciamento de Escopo Assegurar que o projeto inclua todo o trabalho requerido – e somente ele–, visando concluí-lo com sucesso	
Planejar o escopo: criar um plano que documenta como definir, verificar e controlar escopo e estrutura analítica Definir escopo: desenvolver uma declaração detalhada do escopo que servirá de base para decisões futuras Criar estrutura analítica: subdividir principais entregas e trabalho em componentes menores e mais gerenciáveis Verificar o escopo: formalizar a aceitação das entregas que foram concluídas no projeto Controlar escopo: gerir mudanças de escopo ocorridas	Big Plan: distribuição geral de etapas e produtos gerados Cartões com estórias: definem e formalizam o escopo – o cliente enumera os principais casos de uso e define o valor agregado ao negócio por cada um Jogo do planejamento: Equipe e cliente avaliam casos de uso, considerando o lado técnico e interesses do negócio, em busca de consenso quanto ao escopo de cada *release*. Se ocorrerem mudanças no conteúdo da *release* que não possam ser absorvidas, o conjunto de estórias é redefinido
Gerenciamento de Custos Assegurar que o projeto seja concluído dentro do orçamento previsto	
Estimar custos: desenvolver uma aproximação dos custos dos recursos necessários para concluir as atividades Fazer orçamentação: alocar estimativas de custos globais a itens individuais, estabelecer linha de base dos custos Controlar custos: acompanhar fatores que acarretam variações no custo, acompanhar mudanças no orçamento	Desenvolvimento por iteração: de duração fixa, facilita controlar custos – se necessário, negocia-se seu escopo Planejamento de custo amarrado ao de tempo: como cada iteração possui duração fixa, uma estimativa de custo da *release* pode ser obtida a partir do número de pessoas alocadas e de iterações
Gerenciamento de Riscos Cuidar da identificação, análise, monitoramento e resposta a riscos, visando aumentar a probabilidade e o impacto dos eventos positivos e diminuir a probabilidade e o impacto de eventos adversos nos objetivos do projeto	
Planejar o gerenciamento de riscos: decidir como abordar, planejar e executar atividades para gerenciar riscos Identificar riscos: determinar riscos que podem afetar o projeto e documentar suas características Fazer análise quantitativa de riscos: priorizar riscos para análise ou ação, avaliando sua probabilidade e impacto Fazer análise qualitativa de riscos: fazer uma análise numérica do efeito dos riscos nos objetivos do projeto Planejar respostas a riscos: desenvolver opções e ações para aumentar oportunidades e reduzir ameaças Monitorar e controlar riscos: executar planos de resposta a riscos e avaliar sua eficácia durante todo o projeto	Valores ajudam a controlar e mitigar riscos: a busca da simplicidade diminui a complexidade; a realimentação antecipa a detecção de erros; a comunicação aberta minimiza problemas relacionados à falta de informação Práticas ajudam a controlar e mitigar riscos: a quebra em iterações e o planejamento constante ajudam a controlar prazo e custos; cliente disponível e entrega em *releases* diminuem o risco de se obter produtos inadequados Reuniões diárias de acompanhamento: possibilitam identificar com antecedência a iminência de um risco, permitindo atuar a tempo e minimizando suas possíveis conseqüências

Tabela 4. Gerenciamento de projetos: comparativo PMBOK x abordagem ágil
(continuação)

Gerenciamento de Tempo Assegurar a conclusão do projeto no prazo previsto	
Definir atividades: identificar atividades específicas a realizar visando produzir os diversos produtos previstos Seqüenciar atividades: identificar e documentar as relações de dependência entre atividades do cronograma Estimar recursos: identificar tipo e quantidade de recursos para cada atividade do cronograma Estimar duração: identificar períodos de trabalho para concluir as atividades do cronograma Desenvolver cronograma: analisar recursos, restrições, duração e seqüência das atividades e distribuir no tempo Controlar cronograma: acompanhar mudanças	É contra um planejamento inicial detalhado: no início, o cliente explica os casos de uso, em alto nível, e a equipe calcula o tempo para implementação, o que resulta em uma programação grosseira (*releases* e iterações), a ser refinada à medida que o projeto evolui. Se necessário, gerar protótipos e pesquisar soluções, em busca de uma estimativa mais precisa *Releases* decompostas em iterações: o número de iterações pode variar, mas cada iteração possui duração fixa, o que permite acompanhar melhor o progresso obtido
Gerenciamento de Qualidade Determinar responsabilidades, objetivos e políticas para atender às necessidades que motivaram o projeto	
Planejar a qualidade: identificar padrões de qualidade relevantes e determinar a forma de satisfazê-los Garantir a qualidade: realizar atividades de qualidade planejadas, para garantir que o projeto empregue todos os processos necessários para atender aos requisitos Controle da qualidade: monitorar resultados específicos do projeto para determinar se eles estão de acordo com padrões de qualidade relevantes e identificar como eliminar as causas de desempenhos insatisfatórios	Não formaliza a área de qualidade: não define grupo de SQA nem atividades de revisão e auditoria de processos Diversas práticas relacionadas: desenvolvimento dirigido por testes, uso de padrões de codificação, realimentação constante, programação aos pares, integração contínua com suporte de *frameworks* para testes automatizados fornecem métricas reais da situação do desenvolvimento Ao final de cada *release*: o produto é verificado e validado com o cliente. Após aceito, entra em produção
Gerenciamento de Recursos Humanos Organizar e gerenciar a equipe do projeto, fazendo uso mais efetivo de competências e habilidades	
Planejar recursos humanos: identificar e documentar funções, responsabilidades e hierarquia no projeto, além de criar plano de gerenciamento de pessoal Contratar e mobilizar equipe do projeto: conseguir os recursos humanos necessários para trabalhar no projeto Desenvolver a equipe: aperfeiçoar competências e interação da equipe, melhorar o desempenho do projeto Gerenciar a equipe: acompanhar desempenho, resolver problemas, obter realimentação, e coordenar mudanças	Programação aos pares: permite que uns aprendam com os outros, favorecendo o aprendizado Informalidade da documentação: leva à necessidade de um conhecimento tácito da equipe, para não ter que explicitá-lo e formalizá-lo em documento – pode ter vários perfis profissionais, mas pelo menos alguns ágeis Visão direcionada, comprometimento, confiança e cooperação: ajudam a acompanhar e elevar o desempenho *Coach* ajuda a identificar problemas e resolvê-los
Gerenciamento de Comunicações Garantir a geração, coleta, distribuição, armazenamento e recuperação de informação, de forma oportuna e adequada	
Planejar as comunicações: determinar as necessidades de informação e comunicação das partes interessadas Distribuir informações: disponibilizar as informações necessárias às partes interessadas, no momento oportuno Gerar relatórios de desempenho: distribuir informações de desempenho, andamento, medições e previsões Gerenciar partes interessadas: gerir a comunicação para satisfazer requisitos e resolver problemas dos envolvidos	Comunicação: considerada fundamental – confia-se muito na comunicação oral e aberta, sendo desnecessário definir canais e estruturas formais Informalidade da documentação: informações e resultados se tornam de conhecimento tácito e interpessoal, ao invés de documentado e explícito Realimentação constante: junto com a comunicação aberta, permite que o desempenho seja acompanhado
Gerenciamento de Aquisições Gerenciar contratos ou pedidos de compra emitidos pela equipe do projeto, controlando possíveis mudanças	
Planejar compras e aquisições: o que, quando e como Planejar contratações: requisitos / fornecedores potenciais Solicitar resposta de fornecedores: cotação, preço, ofertas Selecionar fornecedores: analisar, negociar, formalizar Administrar contrato: avaliar desempenho, tomar ações Encerrar contrato: liquidar contrato e pendências	Voltada para o desenvolvimento: não faz referência à aquisição de produtos e serviços

5. Considerações Finais

As metodologias ágeis surgem como um novo paradigma para o desenvolvimento de software. Elas apresentam diferenças marcantes em relação às metodologias tradicionais, resumidas na Tabela 5, principalmente por partirem de pressupostos diferentes. Ambas possuem pontos positivos e negativos, e acabam de certa forma se complementando: processos e ferramentas podem melhor capacitar os indivíduos e suas interações; a documentação ajuda as pessoas a entenderem um software complexo; a negociação de contrato pode ser uma parte integrante da colaboração do cliente; seguir um plano pode ser o melhor modo para responder a uma mudança, quando esta é previsível [Boehm 2005]. Desta forma, é necessário buscar o ponto de equilíbrio, avaliando riscos: o planejamento pode aperfeiçoar a agilidade, enquanto esta pode dar maior eficiência ao planejamento. Em qualquer abordagem, porém, as pessoas ativamente envolvidas e suas proposições de valor possuem grande importância.

Tabela 5. Comparativo geral – abordagem tradicional e ágil

Abordagem Tradicional	Abordagem Ágil (XP)
Preditivo: detalhar o que ainda não é bem conhecido	Adaptativo: conhecer o problema, resolver o crítico antes
Rígido: seguir especificação predefinida, a qualquer custo	Flexível: adaptar-se a requisitos atuais, que podem mudar
Burocrático: controlar sempre, para alcançar o objetivo	Simplista: fazer algo simples agora e alterar se necessário
Orientado a processos: podem garantir a qualidade	Orientado a pessoas: motivadas e comprometidas
Documentação gera confiança	Comunicação gera confiança
Sucesso corresponde a entregar o planejado	Sucesso corresponde a entregar o desejado
Gerenciamento no estilo "comando-e-controle", voltada para o trabalho em massa	Gerenciamento no estilo liderança / orientação, voltada para trabalhadores do conhecimento
Ênfase: papel do gerente, planejamento e disciplina fortes	Ênfase: criatividade,flexibilidade, atenção às pessoas
Desenvolvedor hábil (variedade)	Desenvolvedor ágil (colaborador)
Cliente pouco envolvido	Cliente comprometido (com autonomia para decidir)
Requisitos conhecidos, estáveis	Requisitos emergentes, mutáveis
Retrabalho / reestruturação caros	Retrabalho / reestruturação baratos
Planejamento direciona resultados (incentiva controlar)	Resultados direcionam planejamento (incentiva mudar)
Conjunto de processos, com metodologia definida	Conjunto de valores, com atitudes e princípios definidos
Premia a garantia da qualidade	Premia o valor rápido obtido
Foco: projetos grandes ou que envolvam restrição de confiabilidade (exigem mais formalismo)	Foco: projetos de natureza exploratória e inovadores, com equipes pequenas/médias (exigem maior adaptação)
Objetivo: controlar, em busca de alcançar o objetivo planejado (em termos de tempo, custo, escopo)	Objetivo: simplificar processo de desenvolvimento, minimizando e dinamizando tarefas e artefatos
Responsabilidade recai sobre o processo da organização (menos suscetível a falhas?)	Responsabilidade recai sobre o envolvimento e a experiência dos membros da equipe
Foco na maturidade, decorrente da definição e uso de processos e modelos de maturidade	Foco na disciplina, seguindo valores, princípios e "boas práticas" documentados na literatura
Foca em questões ligadas ao gerenciamento, tanto de projeto quanto de processo	Foca em questões ligadas ao trabalho técnico e valor agregado ao produto (resultado)
Institucionalização de processos é crucial – definidos, escritos, treinados, praticados, controlados e cobrados	Utilização das práticas é crucial – princípios e "boas práticas" devem ser levadas ao extremo
Abordagem mais profunda para gerência de projetos	Abordagem ainda superficial para gerência de projetos

É possível conviver, em uma mesma organização, com o formalismo requerido pelos modelos de qualidade e o informalismo pregado pelas metodologias ágeis, obtendo benefícios de ambos. As fases básicas de um projeto de desenvolvimento ágil não são diferentes das fases de qualquer outro projeto, sendo ainda necessário definir e iniciar o projeto, elaborar um plano, executar este plano, e monitorar e controlar os resultados. A maneira pela qual estes passos são realizados na abordagem ágil, porém, é diferente, e requer uma reavaliação em cima dos valores, princípios e práticas ágeis, visando incorporá-los de forma a agregar valor aos projetos.

É importante salientar que a abordagem ágil não é estritamente uma metodologia, mas uma forma de visão e um conjunto de atitudes, valores e princípios. Há processo, mas não no senso rigoroso de um processo baseado em papel ou uma metodologia centrada em controle. Por sua natureza adaptativa e pelo seu foco em resultados, ela demonstra suprir muito bem as necessidades de cada uma das partes envolvidas no fornecimento de software: os usuários têm a oportunidade de obter um sistema mais próximo de suas necessidades; o cliente tem um retorno mais rápido do seu investimento; os desenvolvedores têm a oportunidade de trabalhar em um ambiente melhor e o fornecedor se beneficia com o trabalho de equipes mais eficientes e produtivas. Para colocar estas idéias em prática, porém, é preciso que os desenvolvedores adequem sua forma de desenvolver software, e que o cliente reveja também a sua forma de conduzir e comprar um projeto de software.

Para atender aos anseios do mercado, porém, não basta melhorar apenas o processo de desenvolvimento de software: é necessário definir objetivos claros para a qualidade, comprometer toda a equipe com a melhoria contínua, orientar atividades para resultados de curto prazo, bem como focar na satisfação do cliente e nas necessidades do mercado. A melhoria de processos deve ser implementada não apenas com foco na qualidade do produto, mas também na saúde e longevidade da empresa, considerando fatores como competência, produtividade e inovação. Neste sentido, a cultura da agilidade pode e deve permear todos os processos de uma organização. Seu informalismo não pressupõe desorganização, e suas práticas estão embasadas em muita disciplina. De maneira geral, processos de gestão e realização do produto podem ser totalmente concebidos e implementados com base nos princípios e práticas ágeis.

Por serem mais centradas no desenvolvimento, as metodologias ágeis aparentam minimizar o papel do gerenciamento em assegurar sucesso. Abordagens gerenciais como APM e XPM vieram complementar as práticas de metodologias ágeis existentes e preencher esta lacuna, com diretrizes claras para a liderança de projetos e introdução de práticas efetivas de gerenciamento que requerem criatividade, flexibilidade e atenção somadas às qualidades específicas e interações entre os membros da equipe. Além disso, muitas das práticas de gerenciamento de projeto tradicionais ainda se aplicam a projetos de desenvolvimento ágeis – com alguma adaptação e com uma forte dose de liderança.

Na realidade, independentemente das metodologias ágeis, outras tendências em gerenciamento de projetos já indicavam uma certa convergência entre a comunidade de gerenciamento e a comunidade técnica em relação à agilidade. Muitas das práticas ágeis não são novidade advindas do XP. Por exemplo, o gerenciamento informal já sinalizava a confiança, a comunicação, a cooperação e o trabalho em equipe como elementos essenciais para o sucesso de um projeto, mesmo antes do Manifesto pela Agilidade. Já a abordagem de planejamento ininterrupto, a confiança no processo de desenho evolutivo e o desenvolvimento dirigido por testes são práticas mais recentes, trazidas pelo XP.

A definição de um processo ideal – que seja produtivo, gere menos carga que as metodologias tradicionais e que proporcione garantia de qualidade ao software produzido – ainda é um desafio a ser enfrentado e vencido pelos pesquisadores da Engenharia de Software. Mais do que isso, existe um longo caminho a ser percorrido na busca de processos de software adequados à realidade brasileira, composta principalmente por empresas de software de pequeno porte [MCT 2001].

A ênfase, porém, não deve ficar só nos processos. A necessidade de adaptação muda a perspectiva de uma visão mecanicista, que segue um plano e prevê o futuro, para uma visão mais orgânica, na qual criatividade, inovação, aprendizado rápido, resolução de problemas específicos e efetiva tomada de decisão são determinantes do sucesso [Highsmith 2005]. A visão mecanicista leva a pensar em processos – limitados a um conjunto discreto de entradas e saídas aceitáveis, podendo ser medidos e controlados – enquanto a visão orgânica leva a pensar em padrões – que lidam com a variedade e a ambigüidade, desafiam a criatividade e a habilidade para resolver problemas, podendo ser aprendidos, adaptados e avaliados.

Referências

Agile Alliance (2005) http://www.agilealliance.org.

Augustine, Sanjiv. (2005) "Agile Project Management Explained" http://www.ccpace.com/TechnologySolutions/TechnologySolutions_ProjectManagement.htm

Beck, Kent. et. al. (1999) "Exteme Programming Explained", Addison-Wesley.

Beck, Kent (2001) "Manifesto for Agile Software Development" http://www.agilemanifesto.org

Beck, Kent. et. al. (2001a) "Planning Extreme Programming", Addison-Wesley.

Boehm, Barry. (2005) "Agile and Plan-Driven Methods: Oil and Water?" http://www.xpuniverse.com/pdfs/agileAndPlanDrivenMethods

Highsmith, Jim. (2005) "Does Agility Work?" http://www.agilealliance.com/articles

Jacobsen, Catrine M. (2005) "XPM – from idea to realization - critical approach to the concept of XPM". http://www.glyn.dk/download/synopsisXPM.pdf.

Jeffries, Ron. (2004) "Extreme Programming and the Capability Maturity Model". http://www.agilealliance.com/articles/searchResults?topic=CMM

Kelly, Kevin. (1994) "Out of Control: The New Biology of Machines, Social Systems, and the Economic World", Addison-Wesley.

Kerzner, Harold. (2002) "Gestão de projetos: as melhores práticas". Porto Alegre: Bookman.

Ludwig, Charles. (2005) "Extreme Project Management". http://www.stickyminds.com/sitewide.asp?Function=edetail&ObjectType=ART&ObjectId=6661.

Ministério da Ciência e Tecnologia. (2001) Secretaria da Política de Informática. Sociedade SOFTEX. "Levantamento do universo das associadas SOFTEX", Brasília.

PMI - Project Management Institute (2004) "A Guide to the Project Management Body of Knowledge", Pennsylvania.

Thomsett, Rob. (2002) "Radical Project Management", Prentice Hall.

Thomsett, Rob. (2005) "Extreme Project Management - Executive Report", http://www.cutter.com/research/freestuff/epmr0102.pdf.

Wiki. (2005) http://c2.com/cgi-bin/wiki.