

# Estudo da aplicação de um processo gerenciado de produção de software em MPEs

Sandra Leandro Pereira <sup>1</sup>

leandrlu@terra.com.br

Thiago Pereira de Brito Vieira <sup>2</sup>

tpbvieira@gmail.com

1 Centro de Ciências Sociais Aplicadas (CCSA), Universidade Federal da Paraíba – João Pessoa, PB, Brasil

2 Centro de Ciências Sociais Aplicadas (CCSA), Universidade Federal da Paraíba – João Pessoa, PB, Brasil

## RESUMO

*Este artigo apresenta um debate teórico sobre a gestão de projetos na produção de software em micro e pequenas empresas por meio da análise de diferentes modelos em relação a sua aplicação na produção de software em micro e pequenas empresas. Especificamente, este estudo tem como foco as metodologias de desenvolvimento de software RUP e XP, e o conjunto de conhecimento em gerenciamento de projetos formado pelo PMBOK, sendo realizada uma pesquisa descritiva e exploratória sobre uma análise indutiva sobre as premissas de cada modelo, em busca das características que melhor se aplicam ao contexto da produção de software em MPEs. Os modelos estudados apresentam características relevantes quanto à gestão de projeto de software, que permitem um melhor gerenciamento da produção de software, e estes conhecimentos podem se complementar e serem utilizados conjuntamente para formular um único processo de gerenciamento de projeto de software.*

Palavras-Chave: Gestão de projetos. Processo de desenvolvimento de software. Micro e pequenas empresas.

## 1. INTRODUÇÃO

O processo de desenvolvimento de software vem se aprimorando gradativamente ao longo do tempo, passando de um modo de produção artesanal a uma produção racional e sistematizada. Esta mudança se deu para atender à crescente demanda por softwares com uma maior qualidade, à necessidade de uma produção em curto intervalo de tempo, e às dificuldades existentes durante o processo produtivo de software.

O desenvolvimento de software está sujeito a diversas variáveis que podem influenciar no sucesso do projeto, dentre estas variáveis, pode-se destacar o escopo, o tempo e o custo do projeto (LEPPER, ESPINOZA e SILVEIRA, 2005). O mau gerenciamento destas variáveis pode causar problemas na produção de software. Dentre estes problemas, Pressman (1995, p. 23) destaca a estimação do prazo e de custos freqüentemente imprecisos, a demanda por software ser superior à capacidade de produção das pessoas da área e a qualidade dos softwares produzidos serem aquém da esperada. Outros problemas são apontados por Beck e Andres (2000 p. 10), como os elevados custos com a manutenção e modificação nos softwares e os defeitos existentes que não foram detectados previamente.

Segundo *The Standish Group International* (2004), com base numa pesquisa realizada em 50.000 projetos de tecnologia da informação, apenas 29% destes projetos cumpriram o prazo, o custo e o escopo estimado, 53% necessitaram de mais tempo que o previsto e exigiram maior investimento ou tiveram menos funcionalidades do que o especificado e 18%

foram cancelados ou devidamente concluídos mas nunca tiveram os seus resultados utilizados efetivamente.

As empresas produtoras de software devem ser capazes de gerenciar os riscos inerentes ao processo, de forma a obter os melhores resultados possíveis do processo produtivo, reduzindo os impactos negativos e maximizando os impactos positivos proporcionados pelos riscos. As micros e pequenas empresas podem ser mais vulneráveis aos riscos a que um projeto de software está sujeito, pois estas empresas apresentam uma reduzida capacidade de identificação e reação a imprevistos, o que pode acontecer por estas empresas serem comumente formadas por pessoas com muito conhecimento técnico mas com pouco embasamento gerencial, onde o quadro de funcionários é composto em sua maioria por sócios da empresa. Outra limitação das MPEs produtoras de software consiste na importância que é dada à gestão, tanto em nível de projeto para a produção de software quanto da organização como um todo, onde predomina um planejamento, execução, controle, avaliação e melhoria de forma precária.

Como forma de reagir às demandas existentes e aos riscos inerentes à produção de software, tornou-se necessário desenvolver metodologias que agrupam práticas comprovadamente eficazes no planejamento e na produção de software. Atualmente existem diversas metodologias de gestão de projeto com ênfase no desenvolvimento de software, mas estas metodologias apresentam formas distintas de abarcar determinados aspectos. Com um conhecimento amplo sobre os vários modelos de gestão de projeto e produção de software é possível delimitar quais características de cada modelo podem ser implantadas em uma determinada equipe de produção de software, e obter resultados vantajosos com a sua utilização, como: a redução de erros, a redução do retrabalho na produção, o cumprimento de prazos e custos, uma maior qualidade no produto final, e a redução das limitações impostas por cada modelo através da utilização das características que melhorem o processo produtivo da equipe.

Este artigo explora o guia generalista em gestão de projetos formado pelo *Project Management Body of Knowledge* e modelos dedicados especificamente à produção de software, que são o *Rational Unified Process* e a *Extreme Programming*, realizando uma análise a respeito da sua aplicação em MPEs produtoras de software.

## **2. PROJECT MANAGEMENT BODY OF KNOWLEDGE - PMBOK**

O PMBOK representa todo o conhecimento necessário dentro da profissão de gerência de projetos comprovado através de práticas amplamente utilizadas em todo o mundo e em vários gêneros de projetos (PMI,2004). Criado pelo *Project Management Institute*, o PMBOK forma um guia da gerência de projetos que aborda os aspectos mais importantes de serem gerenciados para assegurar o êxito de um projeto, para isto, este guia define os grupos de processos e as áreas do conhecimento em gerência de projetos.

Os grupos de processos definem atividades que podem ser executadas durante a execução do projeto, agrupando-as dentro de aspectos semelhantes que representam um determinado momento dentro do ciclo de vida do projeto. Os grupo de processos podem ser executados seqüencialmente ou iterativamente, onde alguns grupos podem ser repetidos ciclicamente até a obtenção do resultado esperado. Cada um destes grupos pode se fazer presente em diferentes momentos do ciclo de vida do projeto, podendo dois ou mais grupos serem executados no mesmo período do ciclo de vida de um projeto. Como grupos de processos, o PMBOK define o grupo de iniciação, de planejamento, de execução, de controle e de finalização.

O grupo de processos de iniciação é composto por processos que produzem a autorização formal para o início de um projeto, onde é realizado um levantamento inicial dos requisitos e da viabilidade do projeto, além de serem desenvolvidas descrições dos objetivos do projeto e da solução para alcançar o objetivo. O grupo de processos de planejamento realiza o planejamento e gerenciamento do projeto, desenvolve o plano de gerenciamento do projeto, identifica, de forma mais madura, o escopo do projeto, o custo e as atividades necessárias para a conclusão do projeto. O grupo de processos de execução define processos para realizar as atividades planejadas, onde são definidos quais são os processos específicos para o projeto da equipe, além de coordenar as pessoas, os recursos e realizar a integração das atividades necessárias. O grupo de processo de monitoramento e controle realiza um gerenciamento das atividades em andamento em relação ao plano de gerenciamento do projeto, identificando possíveis problemas e até antecipando-se e prevenindo problemas, ou mesmo corrigindo-os. O grupo de processos de encerramento verifica se o projeto ou fase atendeu a todas as necessidades esperadas para o seu encerramento, uma vez aprovadas, será realizado o fechamento do processo e do contrato.

Os processos de gerência de projetos estão organizados em nove áreas do conhecimento, estas são: a gerência de integração, a gerência do escopo, a gerência do tempo, a gerência do custo, a gerência da qualidade, a gerência dos recursos humanos, a gerência das comunicações, a gerência dos riscos, e a gerência das aquisições do projeto. Estas áreas definem os processos que podem ser utilizados para alcançar o propósito da área do conhecimento, e para cada processo são definidas as entradas necessárias para a sua realização, ferramentas para sua execução e as saídas produzidas pelo processo.

A gerência da integração do projeto descreve os processos necessários para assegurar que os diversos elementos do projeto sejam adequadamente coordenados. A gerência do escopo do projeto descreve os processos necessários para assegurar que o projeto contemple todo o trabalho requerido, e nada mais que o trabalho requerido para completar o projeto com sucesso. A gerência do tempo do projeto descreve os processos necessários para assegurar que o projeto termine dentro do prazo previsto. A gerência do custo do projeto descreve os processos necessários para assegurar que o projeto seja completado dentro do orçamento previsto. A gerência da qualidade do projeto descreve os processos necessários para assegurar que as necessidades que originaram o desenvolvimento do projeto sejam satisfeitas. A gerência dos recursos humanos do projeto descreve os processos necessários para proporcionar a melhor utilização das pessoas envolvidas no projeto. A gerência das comunicações do projeto descreve os processos necessários para assegurar que a geração, a captura, a distribuição, o armazenamento e a apresentação das informações sejam feitas de forma adequada e no tempo correto. A gerência dos riscos do projeto descreve os processos que dizem respeito à identificação, análise e resposta a riscos do projeto. A gerência das aquisições do projeto descreve os processos necessários para a aquisição de mercadorias e serviços fora da organização que desenvolve o projeto (PMI, 2004).

### **3. RATIONAL UNIFIED PROCESS – RUP**

O RUP é uma metodologia de desenvolvimento de software iterativa, centrada na arquitetura e orientada para casos de uso (KROLL e KRUCHTEN, 2003). Esta metodologia objetiva realizar um maior controle sobre os resultados obtidos, gerenciar mudanças, e fomentar um produto de qualidade e estável. O RUP possui princípios básicos que o caracterizam, Kroll e Kruchten (2003, p34) citam como princípios essenciais: atacar os riscos cedo e continuamente, entregar algo de valor ao cliente, focar em um software que possa ser

utilizado o quanto antes, realizar mudanças cedo, liberar protótipos da aplicação, utilizar componentes reutilizáveis, trabalhar como um time e fazer da qualidade um estilo de vida.

No RUP, o desenvolvimento de software é realizado em iterações, onde uma determinada atividade pode ser executada repetidas vezes, até ser obtido o resultado desejado. Isto permite evitar maiores impactos com as mudanças nos projetos, detectar erros previamente, e determinar quais as atividades serão priorizadas, de acordo com o seu risco.

O ciclo de vida definido pelo RUP é formado por quatro fases seguidas seqüencialmente. Estas fases são: a iniciação, que consiste no entendimento das necessidades do projeto, da sua viabilidade e na definição do escopo do projeto; a elaboração, que é a especificação dos pontos de maior risco, como características tecnológicas e arquiteturais a serem utilizadas no projeto; a construção, que é o desenvolvimento do sistema a partir de um código que implemente as funcionalidades e recursos definidos; e a transição, que é formada pelos ajustes, implantação e transferência do sistema para o cliente (KRUCHTEN, 2003).

Cada fase é composta de uma ou mais iterações, que são em geral curtas. Durante as iterações executadas em cada fase, são realizadas atividades denominadas disciplinas, compostas por fluxos de trabalho, que consistem em uma seqüência de tarefas encadeadas e relacionadas a um aspecto importante do projeto, tal como análise do negócio e testes. O RUP possui nove disciplinas, divididas em disciplinas do processo e de suporte. As disciplinas de processo são: modelagem de negócios, requisitos, análise e projeto, implementação, teste e distribuição. As disciplinas de suporte são: configuração e gerenciamento de mudanças, gerenciamento de projeto, e ambiente (KROLL e KRUCHTEN, 2003).

Cabe detalhar os requisitos de cada disciplina. A modelagem do negócio tem como papel listar os possíveis requisitos e entender o contexto do sistema. A definição dos requisitos deve capturar os requisitos funcionais e não-funcionais do sistema. A análise e projeto do sistema define a arquitetura do sistema, de forma a torná-lo mais eficiente, reusável e de fácil manutenção. A implementação tem como responsabilidade criar as especificações dos artefatos em forma de código de software e integrar os resultados obtidos. Ao testes aplicam-se nos componentes produzidos e nas suas funcionalidades. A distribuição visa a entrega do produto acabado ao cliente. A configuração e gerenciamento visam manter a integridade dos recursos durante a evolução do projeto. O gerenciamento de projeto tem como finalidade equilibrar objetivos concorrentes, gerenciar riscos e superar restrições para entregar um produto que satisfaça a necessidade do cliente. O gerenciamento de ambiente é encarregado de dar suporte ao desenvolvimento da organização através da seleção e aquisição de ferramentas e da configuração e melhoria dos processos.

Pode-se afirmar que a arquitetura do RUP é formada em duas dimensões, onde a dimensão horizontal é composta pelas fases do ciclo de vida do projeto, e a dimensão vertical possui as disciplinas que organizam as atividades. Em cada fase, podem ocorrer varias iterações, onde cada iteração aplicará todas as disciplinas previstas para a fase.

O RUP estabelece uma estrutura bem definida para ser utilizada no ciclo de vida do projeto, onde as responsabilidades devem ser bem delimitadas, as atividades devem ser especificadas, e um cronograma estabelecido. Esta estrutura possui elementos principais, que são os papéis, as atividades, os artefatos e os processos.

Um papel define o comportamento e as responsabilidades de um determinado indivíduo ou grupo. Cada papel é composto de um conjunto de atividades, onde uma atividade é uma unidade de trabalho que um indivíduo executa quando está exercendo um determinado papel, podendo esta ser dividida em processos menores. São exemplos de atividades: planejar uma iteração, definir a arquitetura ou executar um teste de performance sobre o sistema. Um artefato é o resultado obtido por uma atividade, que pode ser utilizado como fonte para a

realização de outras atividades. Os artefatos podem ser um diagrama, um documento, o código fonte da aplicação ou uma versão final do software (KRUCHTEN, 2003).

#### 4. EXTREME PROGRAMMING - XP

O modelo XP é um estilo de desenvolvimento de software focado na excelência de aplicação de técnicas de programação, na fácil comunicação, e em uma equipe que permita executar coisas inimagináveis (BECK e ANDRES, 2004). O XP é composto por uma filosofia de desenvolvimento de software baseada na simplicidade das atividades, na comunicação, no *feedback*, na coragem, e no respeito. Estes valores são refletidos nas práticas que compõem o XP. Como características próprias, esta metodologia apresenta:

- Curto ciclo de desenvolvimento, que resultará em um melhor acompanhamento dos resultados, um *feedback* obtido cedo e de forma contínua durante o projeto.
- Planejamento incremental, onde as respostas às mudanças são realizadas conforme o seu surgimento, através de um planejamento realizado constantemente.
- Confiança em: testes automatizados, comunicação oral, colaborações e práticas da equipe de trabalho.

O processo de desenvolvimento adotado pelo XP é semelhante ao de outros modelos, mas prezando pela simplicidade e eficiência nas atividades, onde o resultado é priorizado em detrimento da burocratização e trabalho desnecessário. O processo definido pelo XP é iterativo e incremental, onde as atividades são planejadas e executadas em iterações, de forma a permitir uma melhor reação as mudanças. Durante uma iteração, são executados os processos de análise, projeto, criação dos testes, codificação do módulo e aplicação de testes durante a criação do módulo.

XP tem a sua definição formada por valores, que definem preceitos a serem priorizados em um projeto, por princípios, que pode ser descrito como uma forma mais concreta dos valores, e por práticas, que consiste em uma concretização dos valores em forma de atividades

Os valores definidos pelo modelo *Extreme programming* são:

- Feedback – A interação entre os desenvolvedores do sistema e as partes interessadas é de grande importância, os clientes devem manter-se próximos dos desenvolvedores para proverem informações precisas e avaliarem os resultados obtidos pelo projeto.
- Comunicação – É importante que o fluxo de informações entre os envolvidos no projeto flua livremente, para que os desenvolvedores compreendam o que o cliente deseja e este último entenda os desafios técnicos que precisam ser vencidos.
- Simplicidade – Como acontece no *Just-in-Time*, não se deve produzir mais do que é realmente necessário, simplificando o escopo do projeto de software ao essencial.
- Coragem – As mudanças são variáveis de presença constante em projetos, ainda mais quando estas mudanças são incentivadas, como ocorre com o XP. Diante disto, é necessário adaptar-se as mudanças com segurança e com coragem.
- Respeito - Este valor caracteriza o comprometimento e a valorização dada aos integrantes da equipe e ao projeto como um todo.

Dentre as práticas inerentes ao XP, Beck e Andres (2000) citam:

- Cliente presente – É importante a presença das partes interessadas no projeto junto à equipe de produção para fornecer o *feedback* necessário e avaliar os resultados.
- Jogo do planejamento – O processo de desenvolvimento é realizado de forma iterativa e incremental, onde em cada iteração são realizadas atividades relativas ao projeto como um todo, mas com um escopo limitado, entre estas atividades temos o planejamento, o desenvolvimento, os testes, a refatoração e a avaliação.
- Reunião em pé – Consiste em uma reunião com curto tempo de duração que tem como objetivo atualizar todos os membros da equipe, com informações a respeito do que ocorreu no dia anterior e também priorizar as atividades do dia atual.
- Programação em par – É caracterizada por duas pessoas realizando uma mesma tarefa de desenvolvimento de software, onde uma de fato produz o código do sistema, enquanto uma outra inspeciona constantemente e sugere melhorias para o código.
- Desenvolvimento orientado a testes – Esta pratica defende a estipulação previa de testes para o a funcionalidade a ser desenvolvida, de forma que o desenvolvimento seja guiado para satisfazer os requisitos impostos pelo teste.
- Refatoração – Como forma de manter o código atualizado as mudanças nos diferentes cenários, a refatoração defende a alteração dos componentes do sistema sempre que houver uma oportunidade de melhoria no código produzido
- Código coletivo – Os pares que estarão desenvolvendo estarão em constante revezamento, onde varias pessoas serão responsáveis por um determinado código, e todos os integrantes do projeto tem acesso e autorização para realizar qualquer ação sobre o código que está sendo desenvolvido.
- Design simples – A implementação da aplicação é feita de forma para se tornar o mais simples possível, fazendo apenas o que for de real necessidade no presente.
- Metáfora - São criadas metáforas para simular elementos do mundo físico e seus comportamentos, relacionando termos técnicos com termos do cotidiano do cliente.
- Ritmo sustentável – Consiste em trabalhar de forma mais inteligente e eficiente, onde o importante não é se movimentar muito, mas mover-se na direção certa.
- Integração contínua – Em projetos, que tenham varias pessoas atuando em vários pontos distintos, é necessário garantir que algo produzido por um par consiga se integrar com o produzido por outro par.
- Versões curtas – Versões do resultado esperando com o projeto são produzidos em curto intervalo de tempo, para que as partes interessadas possam avaliar os resultados obtidos e fornecer um *feedback* para a equipe que está desenvolvendo a aplicação.

## 5. ANÁLISE DOS MODELOS

### 5.1 A aplicação do PMBOK à produção de software

O PMBOK não é um manual de uso para ser seguido rigorosamente, mas sim uma fonte de pesquisa e um guia que expõe boas práticas e aspectos importantes de serem considerados na gestão de projetos. Através do conhecimento exposto pelo PMBOK é possível compreender a importância de determinados aspectos dentro da gestão de projetos, desenvolvendo uma base de raciocínio capaz de definir como atuar dentro da gerência de projetos, como organizar as atividades e definir as necessidades e os resultados esperados de

cada atividade realizada. O conhecimento documentado através deste guia pode ser aplicado aos mais variados contextos e dimensões de projetos, no âmbito da produção de software, a utilização do PMBOK tem se mostrado efetiva, e vem sendo adotado por várias organizações, principalmente por aquelas que já possuem ou estão começando a adotar um processo gerenciado de produção, estas organizações buscam unir o conhecimento fornecido pelo guia de gestão de projetos ao conhecimento fornecido por outros modelos específicos à produção de software, dentre estes, modelos específicos à qualidade como o CMMI (*Capability Maturity Model Integration*) e ISO (*International Standards Organization*), ou modelos referente ao processo produtivo de software, como o RUP e o XP (GOEBEL, 2003).

As áreas do conhecimento definidas pelo PMBOK podem ser mapeadas para o contexto da produção de software, sendo algumas destas áreas mais facilmente mapeadas. Existem áreas que apresentam conceitos que estão diretamente ligados ao processo produtivo de software, o que permite um mapeamento de forma simples, estas áreas são: o gerenciamento da integração, o gerenciamento de escopo, o gerenciamento do tempo, o gerenciamento do custo, o gerenciamento da qualidade e o gerenciamento dos riscos. Outras áreas apresentam aspectos que estão relacionados com outros setores de uma organização além de relacionar-se com o processo produtivo, estas áreas são: o gerenciamento dos recursos humanos, o gerenciamento de comunicação e o gerenciamento de aquisições.

Em projetos de software o gerenciamento de integração pode avaliar o retorno sobre o investimento no projeto, através da análise dos resultados esperados, das estimativas de custos e cronograma, e do impacto que a produção do software poderia causar.

Projetos de software têm seu escopo delimitado pelo desenvolvimento de um sistema ou pela alteração de um sistema existente. O gerenciamento do escopo fornece subsídio para que seja definido o problema a ser solucionado por um software e quais os requisitos que este sistema deve implementar.

O gerenciamento do tempo define os processos necessários para que um projeto seja realizado dentro de um prazo previsto. Para isto, é necessário obter informações a respeito dos requisitos do software, para então definir as atividades a serem realizadas, a prioridade e seqüência de cada funcionalidade e estimar o tempo necessário para a sua implementação.

O gerenciamento do custo visa a realização do projeto dentro de um orçamento previsto, para estimar os custos do desenvolvimento de um software é necessário ter informações sobre o esforço a ser realizado, do tempo previsto para a execução das atividades, da equipe envolvida no projeto e dados relevantes de projetos anteriores para servirem como referência para novas estimativas.

O gerenciamento da qualidade determina as responsabilidades e objetivos para obter uma qualidade esperada. Esta qualidade pode estar relacionada ao produto de software ou ao processo utilizado no desenvolvimento. Existem métricas e modelos que mensuram a qualidade e o desempenho do produto, como o CMMI e ISO 9126.

Segundo *The Standish Group International* (1994), o desenvolvimento de software é mais suscetível a determinados riscos, que são apontados como maiores causas de cancelamento ou atrasos em projetos de software. Os cinco maiores riscos de projetos de software são: requisitos imprecisos, interação insuficiente com o cliente, recursos escassos, estimativas imprecisas e as mudanças no escopo do projeto. O gerenciamento dos riscos definido pelo PMBOK é realizado através um plano de gerenciamento de riscos, que é composto pela identificação dos riscos, análise dos riscos, monitoramento e definição das respostas que podem ser dadas aos riscos.

As demais áreas do conhecimento descritas pelo PMBOK apresentam um conjunto de conhecimento que não está diretamente ligado ao processo produtivo de software, pois trata de aspectos que produzem resultados importantes para o desenvolvimento de software, mas podem ter o seu gerenciamento desvinculado da equipe de produção de software e estar presente na organização como um todo.

O gerenciamento da comunicação tem como finalidade garantir que o fluxo de informações do projeto ocorra de maneira proveitosa, administrando a comunicação entre os membros da equipe e entre as partes interessadas no projeto. O gerenciamento de recursos humanos trata da organização e gerenciamento da equipe do projeto, o que é muito importante em projetos de software, pois as pessoas são os maiores responsáveis pelas atividades a serem realizadas no projeto. O gerenciamento das aquisições consiste em processos para adquirir produtos, serviços ou resultados além da equipe do projeto, que em projetos de software podem ser representados pela aquisição de softwares, hardware e serviços terceirizados ou contratos de prestação de serviço.

Os grupos de processo definem o momento em que determinados processos podem ser aplicados em um projeto de software, onde os grupos estão distribuídos ao longo do ciclo de vida do projeto, de forma seqüencial ou iterativa.

O grupo de processos de iniciação estabelece processos que produzem a autorização formal para o início de um projeto, avaliando as necessidades do software e a sua viabilidade de desenvolvimento, de acordo com o escopo do projeto, as tecnologias necessárias, o tempo e os resultados que podem ser obtidos. O grupo de processos de planejamento determina detalhadamente os requisitos que o software deve ter e quais as estimativas de qualidade a serem controladas, além de definir como estas funcionalidades estarão organizadas dentro do sistema, os testes a serem realizados e as pessoas envolvidas no projeto. O grupo de processos de execução é composto de processos que visam realizar as atividades planejadas, transformando as especificações do projeto em código para compor um software executável. O grupo de processo de monitoramento e controle realiza um gerenciamento sobre as atividades em andamento em relação ao plano de gerenciamento do projeto, verificando a conformidade das atividades em relação ao planejado. O grupo de processos de encerramento verifica se o projeto atendeu a todas as necessidades tidas para o seu encerramento, verificando se os resultados obtidos e sua qualidade estão de acordo com o esperado.

## **5.2 Aplicação do RUP ao contexto das MPEs**

A forma como o RUP é descrito permite a sua utilização em projetos de diferentes portes, pois é fornecido embasamento suficiente para a sua aplicação em grandes ou em pequenos projetos, através da sua configuração e implementação para adaptar-se ao contexto de cada caso, como é descrito por Kroll e Kruchten (2003).

Configurar o Rational Unified Process significa adaptar o processo às necessidades e restrições da organização, e implementá-lo significa mudar a cultura da organização para a adoção deste processo. Segundo Kruchten (2003), a implementação deve ser um processo planejado e gerenciado, realizado através da análise do cenário, do planejamento, da implantação e da avaliação e controle dos resultados obtidos.

A avaliação do contexto atual da organização fornece informações importantes para a implantação do processo, avaliando: o processo que está sendo utilizado, os objetivos a serem obtidos com a adoção do novo processo, os pontos que necessitam de melhoria, e a equipe sob o foco das habilidades e competências tidas pelos membros. Os riscos é outro fator que deve ser avaliado, identificando-os e definindo as ações necessárias para prevenir ou atacá-los. Com base nestas informações levantadas é possível montar um plano descrevendo a

implementação do processo, onde são definidas as ferramentas e a forma de controle e avaliação a ser usada.

O RUP fornece várias definições de papéis, atividades, artefatos, disciplinas e fluxos de trabalho, que devem ser analisados quanto a sua necessidade e viabilidade de implantação em uma MPE. Em pequenos projetos, as fases, as disciplinas, os artefatos e os papéis descritos pelo RUP podem não estar precisamente delimitados, mas se apresentando distribuído entre a equipe do projeto, onde um membro da equipe pode assumir dois ou mais papéis.

Independente dos recursos disponíveis, do tamanho da equipe e do projeto, é importante compreender a problematização do projeto de software, planejar as ações necessárias, realizá-las, controlá-las e implantar os resultados obtidos com o projeto no cliente, isto caracteriza os princípios abordados nas fases propostas pelo RUP. Em uma MPE é importante adotar um processo preventivo que permita identificar os riscos prematuramente e melhorar incrementalmente a qualidade do produto e do processo, o que é proporcionado pela adoção do RUP como processo produtivo de software.

As disciplinas definidas pelo RUP definem atividades relevantes para a gestão de projetos de desenvolvimento de software. É importante que o conceito definido em todas as disciplinas se façam presentes no projeto, independente do seu detalhamento. Em pequenas equipes, o uso das disciplinas deve ser avaliado, de forma a determinar o detalhamento e o nível de uso adequado para o seu caso, onde devem ser realizadas apenas as atividades que agregam valor e que tragam benefícios ao processo e produto.

Um projeto de software realizado por uma MPE normalmente possui poucas pessoas envolvidas com o processo produtivo, o que torna necessário que uma pessoa realize atividades de responsabilidade de vários papéis propostos pelo RUP. Mesmo se tratando de uma equipe pequena, é importante que as responsabilidades estejam presentes e que as atribuições sejam bem definidas, para permitir um melhor controle sobre as atividades realizadas.

Artefatos permitem uma melhor compreensão e visualização das informações, facilitando a compreensão do projeto, mas criar e manter estes artefatos demanda esforço e tempo, que são variáveis limitadas em projetos realizados por MPEs. Desta forma, os artefatos a serem utilizados em um projeto devem ser escolhidos através de uma análise e avaliação quanto a sua importância e quanto a relação entre o custo e o benefício.

### **5.3 Aplicação do XP ao contexto das MPEs//**

Beck e Andres (2000) definem o XP como sendo uma metodologia leve para ser utilizada por pequenas e médias equipes desenvolvedoras de software, mas, ao longo do tempo, seu uso apresentou valiosos benefícios, chamando a atenção de grandes equipes produtoras de software e até mesmo equipes que atuam em outro segmento de projeto, como se pode comprovar através da adoção dos princípios ágeis defendido pelo XP na *Agile Project Management* (APM). Beck e Andres (2004) afirmam que o XP é um processo capaz de ser aplicado a projetos de qualquer tamanho, pois os valores e práticas definidos por esta metodologia se mostram aplicáveis a projetos de qualquer escala.

A aplicação do XP a projetos e equipes de tamanho variado é possível devido a forma abrangente como os valores são definidos, e partindo destes valores, são definidos os princípios e as práticas adequadas para cada caso. Um dos valores do XP é a simplicidade, que se mostra presente em todo o processo através da utilização de práticas simples de serem implementadas e que necessitam de um reduzido valor de investimento.

A implantação do XP deve ser um processo racional, gerenciado e incremental, que se adapte às mudanças que possam ocorrer no projeto ou na equipe. O ambiente da organização deve ser analisado antes da implantação para identificar como se dá o processo produtivo adotado atualmente, e avaliar quais são as limitações e os riscos que este processo apresenta. Para combater os problemas identificados e prover uma melhoria no processo utilizado, são estabelecidas metas a serem cumpridas, que devem estabelecer como solucionar os problemas e melhorar a produção de software através da adoção de valores e das práticas.

Os valores e as práticas do XP a serem adotadas pelo processo devem ser definidas de acordo com os problemas identificados e com as metas que foram definidas, de forma a ser identificado o melhor meio de atacar os problemas mais significativos primeiro. A adoção dos valores é uma transformação cultural da organização, onde a equipe deve estar consciente da importância de cada valor e comprometida com a sua adoção. Estes valores devem ser traduzidos em práticas, que representam a aplicação de um valor de forma mais concreta. É importante que a implantação das práticas seja feita de forma gradativa, pois a ordem de adoção das práticas é importante para a resolução dos principais problemas primeiro e para a melhoria do processo.

Os papéis definidos pelo XP apontam responsabilidades que devem estar presentes em projetos de produção de software. Em equipes com um reduzido número de membros, vários papéis podem estar associados a uma única pessoa, que acumula responsabilidades dentro do projeto, como já é previsto pela metodologia. É importante que as responsabilidades definidas pelo XP estejam presentes no processo, mesmo que estas não sejam atribuições feitas a uma única pessoa.

## 6. CONCLUSÕES

O conhecimento abordado pelo PMBOK determina aspectos importantes de serem gerenciados em projetos, independente do porte do projeto, da sua área de atuação ou do tamanho da equipe envolvida. A forma como este conhecimento será aplicado e gerenciado deve ser adequado à organização e ao projeto em questão, para isto é necessário analisar como os grupos de processos, as disciplinas e os processos devem se fazer presentes no processo produtivo adotado pela organização.

Em uma pequena equipe produtora de software, o PMBOK pode ser utilizado como um guia para indicar quais aspectos devem ser gerenciados e quais atividades podem ser realizadas para gerenciar estes aspectos. As áreas do conhecimento descritas pelo PMBOK definem aspectos importantes de serem gerenciados, e os processos definem atividades que podem ser realizadas para gerenciar estes aspectos.

É importante que o gerenciamento de todas as áreas do conhecimento seja efetivamente realizado em projetos, mas com um detalhamento e amplitude definido de acordo com o contexto da equipe e do projeto. Algumas áreas do conhecimento se mostram mais efetivas em projetos realizados por uma equipe pequena, e podem ser priorizadas, estas são: a gerência de integração, a gerência do escopo, a gerência do tempo, a gerência dos custos, a gerência de riscos e a gerência da qualidade. As outras áreas não são menos importantes, mas abordam aspectos que se tornam pouco evidentes em equipes pequenas, como ocorre com o gerenciamento de recursos humanos e o gerenciamento da comunicação, que apresentam poucos conflitos devido ao relacionamento e comunicação presencial adotado nestas equipes. A gerência de aquisições pode se mostrar importante de acordo com o porte do projeto e dos contratos que forem estabelecidos e que necessite de um gerenciamento mais elaborado.

O RUP apresenta um conjunto de conhecimento, boas práticas e ferramentas importantes para o sucesso de projetos de software. Este modelo apresenta uma ampla abordagem de gestão de projetos focada para a produção de software, embora não abarque alguns aspectos importantes na gestão de projetos, como é o caso do gerenciamento de custos. As fases, disciplinas, artefatos e papéis definidos pelo RUP podem estar presentes em projetos de software de diferentes contextos, independente do tamanho do projeto ou da equipe (KROLL e KRUCHTEN, 2003).

As fases do RUP determinam, sob uma abordagem iterativa, o ciclo de vida que um projeto pode utilizar, pois as fases definem um caminho gerenciado que pode ser seguido para a produção racional de software, abordando aspectos relevantes desde da idealização do software até a sua produção e implantação no cliente. O processo iterativo permite um melhor gerenciamento do projeto através de um processo incremental realizado periodicamente, onde as atividades são divididas em iterações menores, facilitando o planejamento e gerenciamento das atividades, que serão realizadas periodicamente e em uma granularidade menor, o que permite um maior detalhamento, controle do projeto, além de tornar possível a identificação e reação a riscos prematuramente.

As disciplinas definidas pelo RUP estabelecem um guia de atividades que podem ser realizadas para garantir a produção de um software de qualidade dentro do prazo e custo estimado, indicando quais aspectos devem ser gerenciados durante a produção do software, cabendo a equipe determinar como e quais atividades serão realizadas para atender ao propósito de cada disciplina.

Os artefatos definidos pelo RUP se mostram efetivos quanto a documentação e representação do sistema e das atividades realizadas, mas é importante notar que a criação e manutenção de artefatos é uma atividade que necessita de um considerável esforço, o que pode se tornar inadequado para equipes pequenas. Portanto, os artefatos a serem utilizados devem ser avaliados quanto a sua relação entre o custo e o benefício. As responsabilidades definidas pelos papéis descritos no RUP devem ser distribuídas entre a equipe, onde uma pessoa poderá ter atribuições de diferentes papéis, de forma que todas responsabilidades sejam abarcadas.

O XP é um processo de desenvolvimento de software com foco na melhor aplicação de técnicas de programação, de comunicação, de gestão de equipe de trabalho, para proporcionar um desenvolvimento de software preciso e focado nos resultados entregues ao cliente. Através da definição de valores, princípios, práticas e papéis o XP define um processo simples e adaptável para a produção de software.

Os valores e práticas descritos pelo XP definem um modo de produção de software simples, que se adapta às mudanças através de um gerenciamento participativo e incremental. O gerenciamento do projeto é realizado pela própria equipe de produção, que através do planejamento e das reuniões periódicas administram o processo de produção de software.

O XP é uma das metodologias ágil de produção de software existentes, este tipo de produção ágil também se faz presente em outros segmentos de projetos, como define o *Agile Project management*. Estas metodologias podem ser aliadas às metodologias tradicionais de gestão de projetos, como o PMBOK, de forma que uma metodologia mitigue as deficiências apresentadas por outra e a complemente.

Do ponto de vista do contexto estudado, estes três modelos, o PMBOK, o RUP e o XP, podem ser adaptados conjuntamente para proporcionar um melhor gerenciamento de projetos de software em uma MPEs produtoras de software, através da adoção dos pontos mais relevantes de cada modelo em relação ao contexto.

Os valores do XP podem ser utilizados para definir a cultura da equipe, que aplicará as áreas do conhecimento definidas pelo PMBOK e as disciplinas definidas pelo RUP dentro do ciclo de vida iterativo e incremental definido pelo RUP. Como forma de atender às definições das áreas do conhecimento estabelecidas pelo PMBOK e das disciplinas definidas pelo RUP serão utilizadas as práticas definidas pelo XP. A forma como se dará a documentação e a definição de papéis do processo deve estar de acordo com a necessidade do projeto em questão, levando sempre em consideração a sua real necessidade e a relação entre o custo e o benefício, podendo utilizar como base os papéis definidos pelo RUP e pelo XP.

## 7. REFERÊNCIAS

BECK, Kent; ANDRES, Cynthia. *Extreme Programming Explained: Embrace Change*, Addison Wesley Professional, 2000.

BECK, Kent; ANDRES, Cynthia. *Extreme Programming Explained: Embrace Change*, 2. ed. Addison Wesley Professional, 2004.

DIAS, Marisa Villas Bôas, SOLER, Afonso Mazini. *Agile Project Management*. Mundo Project Management, Paraná, v. 1, n. 4, p. 14-19. 2005.

GOEBEL, Clement James. *Extreme programming practices used to facilitate effective project management*. Menlo Institute LLC, 2003.

KROLL, Per; KRUCHTEN, Philippe. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*, Addison Wesley, 2003.

KRUCHTEN, Philippe. *Introdução ao RUP Rational Unified Process*, Editora Moderna, 2003.

LEPPER, Ricardo, ESPINOZA, Hariano H, SILVEIRA, Bárbara. *Gerenciamento de projetos utilizando PDCA e modelos workflow*. Mundo Project Management, Paraná, v. 1, n. 1, p. 24-28. 2005.

MAGALHÃES, Ana L. de Cenni de C. *O gerenciamento de projetos a luz das metodologias ágeis*. PMI. Minas Gerais, 2005.

PMI – Project Management Institute, *Um guia do conjunto de conhecimento em gerenciamento de projetos*, 3. ed, 2004.

PRESSMAN, Roger S. *Engenharia de software*. Makron Books, 1995.

The Standish Group International. *The chaos report*. 1994. Disponível em <[http://www.standishgroup.com/sample\\_research/chaos\\_1994\\_1.php](http://www.standishgroup.com/sample_research/chaos_1994_1.php)>. Acesso em 06 de junho de 2006.