

1

Introdução

1.1. Motivação

Ao longo das últimas décadas, observou-se um aumento enorme na complexidade dos sistemas de software desenvolvidos, no número de profissionais que trabalham nesta área, na qualificação técnica destes profissionais. Surgiram também metodologias, técnicas, linguagens e ferramentas que possibilitam alta produtividade e maior atenção à qualidade dos artefatos. Apesar disso, desenvolver software com qualidade, em tempo hábil e a um custo viável ainda representa um desafio.

Observa-se mais recentemente uma mudança na abordagem para resolver este desafio de forma sistemática, por parte da comunidade de Engenharia de Software: mudar a ênfase de metodologias, ferramentas e linguagens para processos e gerência de projetos.

O grande desafio da Engenharia de Software é desenvolver software de qualidade assegurada, com elevada produtividade e dentro dos recursos inicialmente alocados. Continuamente aparecem novas ferramentas, métodos e metodologias que, freqüentemente, se autodenominam soluções miraculosas capazes de resolver todos os nossos problemas do desenvolvimento de software. No entanto, a contribuição na maior parte das vezes tem sido duvidosa.[GLASS99] [GLASS94] [GLASS03]

O que realmente se percebe é que ainda continuamos a gastar muito tempo para manter o produto de software que construímos, devido a um processo fraco de gerenciamento, dificuldade de entendermos o que realmente precisamos construir para atender as necessidades de nossos clientes, falta de eficiência na capacidade de alterar os sistemas de e de controlar as alterações que nos são solicitadas a todo o tempo, durante o ciclo de vida de um produto de software [KEMERER97].

A busca da qualidade do produto através do gerenciamento do processo de desenvolvimento de software pode ser considerado o último grande esforço do final do século passado em direção a melhoria da capacitação de desenvolvimento de software das organizações. Baseia-se em princípios de qualidade propostos por Walter Shewart [SHEWART31], desenvolvidos por Edwards Deming e Joseph Juran [JURAN88] e adaptados pelo Software Engineering Institute - SEI [W_SEI01] em um modelo de maturidade que estabelece uma base para gerenciamento e engenharia de projeto, visando a melhoria contínua dos processos de software, conhecido como SW-CMM e CMMI [W_SEI02]. Tais modelos são estruturas que descrevem os principais elementos de um processo de software efetivo, baseados nas melhores práticas reconhecidas.

Outro fator motivador é o desafio enfrentado pelo Brasil hoje, de elevar o país ao status de exportador de software. O mercado interno representa hoje o sétimo maior mercado consumidor do setor, que importa grande parte das soluções e teria potencial humano e tecnológico para ser um grande exportador a longo prazo. A melhoria da qualidade do software produzido seria um requisito imprescindível para alcançar este objetivo. Neste cenário, muitos dos produtores de software são micro e pequenas empresas, que contam com equipes de desenvolvimento extremamente pequenas, com em torno de dez desenvolvedores. Em 2000, existiam 5,7 mil empresas de software no Brasil, das quais, 82% eram micro-empresas e somente 2% grandes empresas. As micro, pequenas e médias empresas são responsáveis por aproximadamente 52% de todo software produzido, e por 54% dos empregos formais na área [SOFTEX02].

Por exemplo, com o intuito de criar um modelo de maturidade de capacitação destes pequenos empreendimentos a Sociedade Softex criou o processo MpsBR [W_MPS01]. O projeto tem por objetivo implantar o Modelo de Referência (MR mps) para melhoria no processo de software, principalmente nas pequenas e médias empresas. O modelo é baseado nos padrões ISO 15504/12207 [W_ISO01] e no CMMI [W_CMMI01].

No entanto, observa-se neste ambiente de pequenas organizações desenvolvedoras de software que o uso de processos formais e adequados não é uma prática disseminada ainda, o que compromete a qualidade e cria riscos para o projeto. Dentre as principais dificuldades encontradas por empresas de pequeno porte estão [LARYD00]:

- a limitação de recursos humanos, que torna comum a sobrecarga da equipe em eventualidades comuns, como a contratação de projetos novos ou a concretização de riscos técnicos em projetos já existentes, que geram dificuldades imprevistas que demandam tempo de programação;
- a falta de experiência e conhecimento na equipe, principalmente em alguns processos específicos mas imprescindíveis, como gerência de configuração;
- a limitação de recursos financeiros, que criam uma barreira na adoção de ferramentas e no treinamento da equipe e;
- a pouca disponibilidade de material publicado sobre implantação de processos e gestão da qualidade em ambientes similares.

A incapacidade destes modelos de negócio de implantar a gestão eficiente da qualidade em micro-empresas talvez deva-se ao fato de que a maioria dos modelos de processos mais populares existentes, como CMM [W_SEI01], BOOTSTRAP [KUVAJA94], SPICE [MELO97], e tantos outros, sejam inadequados às suas realidades. Estes processos e modelos foram criados tendo em vista projetos de grande e médio porte e representam uma sobrecarga de atividades de natureza gerencial pesada demais para estas empresas. Além disso, estes processos costumam ser pouco prescritivos, o que torna difícil sua adoção por equipes com um corpo de conhecimento total pouco abrangente, principalmente na área de gestão de processos.

O objetivo deste trabalho é desenvolver e avaliar um processo que seja aderente e de fácil implantação para projetos desenvolvidos por pequenas equipes.. A proposta apresentada é de caráter prescritivo, e visa fornecer material técnico que possibilite a gestão da qualidade em qualquer projeto com as características que descrevemos anteriormente: equipes pequenas, poucos recursos financeiros e onde a qualidade é valorizada. Devem ser consideradas questões e restrições como a adaptabilidade do processo a projetos de tamanhos e características diferentes e um número reduzido de papéis.

1.2. Objetivos do Trabalho

Ao longo desta dissertação proporemos um processo que visa pequenos projetos. Entendemos que um processo deve possibilitar a melhoria da qualidade de serviço, de engenharia e de projeto; a diminuição de custos através do aumento da previsibilidade e capacidade de mitigar riscos, bem como a melhoria da eficiência e produtividade da organização. Em última análise, qualidade refere-se à adequação do processo e do resultado final às expectativas do cliente. Este fator é essencial para garantir que uma pequena empresa seja competitiva no mercado onde está inserida.

Um dos pontos centrais do processo proposto é ser minimalista e ao mesmo tempo fácil de ser implantado e controlado. Por minimalista entende-se o processo que não produz mais artefatos ou consome mais recursos do que o necessário. Define-se o “tempo de projeto” como o tempo total gasto na execução de tarefas e na produção de artefatos que podem ser necessários, mas não são objetivo final do projeto (o objetivo final seria a codificação do próprio sistema, de acordo com as expectativas do cliente). Deseja-se sempre minimizar o tempo de projeto, e supõe-se que em um projeto pequeno este tempo seja percentualmente bem menor do que em projetos de grande porte.

Apesar de apresentarem algumas dificuldades, projetos menores, desenvolvidos por equipes pequenas têm algumas vantagens em relação a projetos de maior escala. Em projetos maiores, é geralmente necessário um esforço (ou um tempo de processo) proporcionalmente maior para gerência eficiente de um número também maior de recursos. Empresas pequenas têm maior facilidade de comunicação, exigem processos mais enxutos para um mesmo nível de qualidade final. Algumas abordagens disponíveis para este tipo de ambiente são de difícil aplicação em empresas maiores. Uma abordagem que recentemente tem tido muita aceitação neste cenário é o uso de métodos ágeis, tais como eXtreme Programming (XP) [BECK00], SCRUM [POWER02], FDD [W_FDD01], Crystal [COCKBURN01] e outros. Estes processos são especialmente aderentes a realidade de desenvolvimento com pequenas equipes.

Porém o minimalismo exagerado do XP pode levar à confusão, ambigüidade e inconsistência. XP parece depender de uma equipe brilhante e disciplinada,

completamente comprometida com os interesses do cliente. Faltam processos bem definidos que possam ser seguidos mesmo por uma equipe mediana, e que possam ser controlados e avaliados.

O processo proposto a seguir é híbrido, e tem características de métodos ágeis, porém herda do Rational Unified Process (RUP) [KRUTCHEN00] um caráter mais prescritivo, no intuito de facilitar a implantação e controlabilidade do processo. Ainda assim, é um processo suficientemente flexível, de forma a poder ser adaptado a objetivos e requisitos de projetos diferentes.

1.3. Organização do Documento

No segundo capítulo serão apresentados alguns conceitos básicos importantes para o entendimento e elaboração do trabalho. Serão abordadas algumas definições e conceituações relacionadas a processos de desenvolvimento de software, além de uma visão geral de alguns processos consagrados e amplamente utilizados pela indústria hoje. Será apresentado também um framework para representação de processos de software que será usado para a representação do processo proposto por este trabalho.

O terceiro capítulo descreve detalhadamente a estrutura do processo proposto, mas antes apresenta algumas das características desejadas para este processo.

No quarto capítulo apresentaremos um estudo de caso real da evolução e implantação do processo proposto. Começaremos o capítulo apresentando o cenário onde o processo foi concebido, sua evolução e questões práticas observadas sobre a implementação gradual do processo, considerando um ambiente típico de aplicação. A seguir, apresentaremos o estudo de caso baseado em um projeto pequeno onde o processo foi aplicado em seu formato final, conforme apresentado no terceiro capítulo. O objetivo deste estudo de caso é validar e dar base realística para uma análise crítica consistente do processo proposto, além de apresentar um registro da experiência de concepção e implantação do processo.

O quinto capítulo, por fim, apresenta uma análise crítica conclusiva abordando tanto o processo proposto quanto a experiência de aplicação do

processo no ambiente do estudo de caso, apresentando oportunidades de melhoria e propostas para possíveis trabalhos futuros que possam ser desenvolvidos.