

# 1 Introdução

Um dos desafios enfrentados pela Engenharia de Software é o de criar instrumentos para que um produto de software possa ser desenvolvido com qualidade e de forma eficiente, consumindo o mínimo de recursos em seu desenvolvimento.

Atualmente existe um consenso de que um dos fatores mais importantes para o desenvolvimento de um produto de qualidade é o seu processo de desenvolvimento. Um processo bem definido faz com que sejam muito altas as chances de uma organização desenvolver um produto de qualidade, respeitando prazo, orçamento e os critérios esperados de qualidade do produto (Chrissis et al., 2003). Isto decorre da premissa de que a definição de atividades padronizadas a partir da ordenação e definição clara da dependência entre elas diminui o espaço para erros humanos.

A inexistência ou a não utilização de processos bem definidos, mesmo que informalmente, na forma de políticas e expectativas da organização para o processo produtivo, faz com que o desenvolvimento de cada produto seja executado de uma forma diferente, provocando resultados imprevisíveis e dificultando a reutilização de boas práticas e das lições aprendidas.

Existe também um consenso sobre o fato de que um processo bem definido torna-se estável, previsível e gerenciável, sendo executado de forma muito mais eficiente (Chrissis et al., 2003). Esta afirmação parte do princípio de que um processo definido permite que estimativas mais exatas sejam feitas utilizando dados das execuções anteriores do processo, na forma de projetos, além de utilizar práticas padronizadas e de eficiência comprovada pelo histórico de execuções. Um outro fato que corrobora para esta afirmação é que, quando uma organização utiliza um conjunto de processos padrão, cuja customização para cada projeto ocorre de forma controlada, torna-se possível definir métricas comuns e comparar medidas de desempenho e de qualidade obtidas em diversas execuções destes processos em projetos da organização. Dessa forma, pode-se controlar a execução

dos projetos, identificando pontos de falha e corrigindo-os antes que estes comprometam o resultado final.

No entanto, a melhoria em práticas implica em um investimento contínuo em pessoas, técnicas e ferramentas como veremos na próxima seção.

### 1.1. Melhoria Contínua

Uma das formas mais indicadas para a definição e implantação de processos de forma eficiente é a utilização de um ciclo de melhoria contínua. O modelo IDEAL (Gremba, Myers, 1997), desenvolvido pelo Software Engineering Institute (SEI), ilustra a utilização deste conceito:

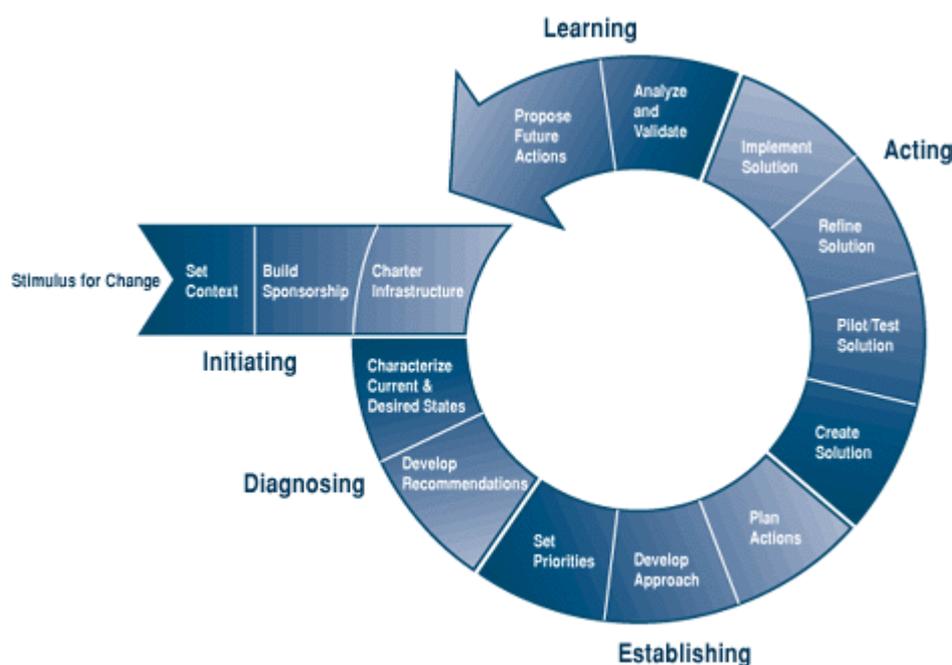


Figura 1: ciclo do modelo IDEAL

A primeira etapa (Iniciação) é executada apenas na primeira iteração do ciclo e corresponde a uma fase de planejamento e obtenção do comprometimento da organização com a implantação e evolução de processos que irão guiar o desenvolvimento de software. Nesta etapa os processos informais são identificados e mapeados, e o ambiente organizacional é preparado para que a definição e implantação dos processos possam ocorrer de forma gradual e eficiente.

A segunda etapa (Diagnóstico) consiste em avaliar o ambiente produtivo e identificar as oportunidades de melhoria. Em geral, nesta etapa é realizado um estudo de “*Gap Analysis*”, no qual um cenário esperado é definido e o cenário atual é avaliado segundo algum critério de qualidade. Dessa forma, obtém-se a

diferença entre o que se espera dos processos da organização e onde eles realmente estão. A partir daí, elaboram-se planos de ação para que esta distância seja diminuída ou eliminada.

Finalizando o ciclo, é realizada uma etapa de priorização e seleção dos planos de ação que serão implantados (Estabelecimento), seguida de uma etapa de implantação destes planos (Ação) e de análise *post mortem* (Aprendizado), onde são identificadas e documentadas as lições aprendidas. O ciclo é então reiniciado, aproveitando os dados e lições coletadas nas execuções anteriores para melhorar cada vez mais os processos e adaptá-los às mudanças no ambiente interno e externo.

A implantação deste ciclo faz com que os processos da organização tornem-se vivos, sendo constantemente avaliados e melhorados. A utilização de iterações de melhoria facilita esta evolução, uma vez que diminui o impacto da mudança.

## **1.2. Motivação**

Embora muitas organizações tenham o conhecimento da importância da utilização de processos definidos e otimizados e, em alguns casos, até tenham um ambiente organizacional preparado para a implantação de um ciclo de melhoria contínua, muitas acabam implantando processos de forma ineficiente (Poulin, 2005).

A necessidade de existir um processo bem definido esbarra no problema de como defini-lo e institucionalizá-lo na organização. Na área de desenvolvimento de software existem tantas disciplinas e metodologias que a definição de um processo de desenvolvimento pode se tornar uma atividade exaustiva, demandando uma grande quantidade de pesquisa, experimentação e estudo, além de um grande conhecimento das várias etapas do desenvolvimento. A quantidade de variáveis que devem ser consideradas ao longo da implantação deste processo definido, como motivação e integração da equipe, características da organização e dos projetos que ela realiza e cultura organizacional é um fator que aumenta ainda mais a complexidade desta atividades.

Visando solucionar este problema foram elaborados diversos modelos de processo, modelos de maturidade e normas de qualidade. Modelos de processo

definem uma estrutura padrão para a execução de projetos de desenvolvimento de software, funcionando como um processo “pré-moldado” que pode ser utilizado de forma integral ou customizado para as características específicas de uma organização ou projeto. Dentre os diferentes modelos de processo existentes podemos citar: Unified Software Development Process (Jacobson, Booch, Rumbaugh, 1999), Rational Unified Process (RUP) (Kruchten, 2000), Crystal (Cockburn, 2000), Extreme Programming (XP) (Beck, 1999).

Modelos de maturidade e normas de qualidade funcionam como guias para a definição de processos, indicando o que deve ser feito durante o desenvolvimento de produtos de software, mas sem indicar como isto pode ser feito. Estes modelos e normas representam o conhecimento de vários indivíduos e organizações, adquiridos ao longo dos anos, refletindo as lições aprendidas tanto no sucesso quanto no fracasso, e indicam uma série de características ou boas práticas de desenvolvimento que devem estar presentes nos processos para que estes resultem em um produto de qualidade, gerado de forma eficiente. Dentre os modelos de maturidade e normas de qualidade podemos citar: Capability Maturity Model Integration (CMMI) (SEI, 2006b), ISO/IEC 12207 (ISO/IEC, 1995), Melhoria do Processo de Software Brasileiro – MPS.BR (SOFTEX, 2006b).

No entanto, a utilização destes modelos e normas nem sempre é uma tarefa simples. Na maioria das vezes é oferecido um grande número de processos e atividades (modelos de processo) e boas práticas isoladas (modelos de maturidade e normas de qualidade), todos descritos de forma abstrata. Entretanto, não é indicado, de forma objetiva, como estes devem ser customizados, implementados e utilizados, ou como eles se encaixam em um processo produtivo já existente, ou seja, eles indicam o que deve estar presente no processo, mas não indicam como um processo que atende a norma pode ser definido.

Tendo um processo definido, o próximo passo é verificar se o mesmo está de acordo com as normas de qualidade e modelos de maturidade selecionados pela organização. Para tal são utilizados métodos de avaliação, que indicam se o processo satisfaz as diretrizes definidas por este modelo. Entretanto, devido à quantidade de informação que deve ser levantada e analisada, os métodos de avaliação, em geral, consomem muito tempo e recursos da organização, tornando-se uma atividade custosa e muitas vezes mal vista pelos integrantes da organização. Em muitos casos as avaliações são realizadas de forma que não são

fornecidos todos os dados necessários para que a organização possa identificar os pontos de melhoria prioritários e tomar as decisões necessárias para implementá-los. Não necessariamente o modelo deve ser aplicado da mesma forma em todas as organizações, prioridades são diferentes de acordo com a natureza da empresa e do contexto onde ela se insere.

O objetivo deste trabalho é fornecer uma ferramenta customizável (método e ferramenta de apoio) de auxílio à avaliação e melhoria pragmática de processos organizacionais. Para isto é utilizada de forma combinada uma abordagem de análise de risco e conformidade. Esta análise é relacionada aos objetivos da organização e fornece suporte à elaboração de um plano de ação.

Dessa forma, espera-se facilitar a implantação de um ciclo de melhoria contínua e também contribuir para a definição e implantação de processos mais eficientes e aderentes aos modelos de maturidade e normas de qualidade.

### **1.3. Guia do Leitor**

No segundo capítulo, serão apresentados os conceitos básicos de análise de risco que regem grande parte das contribuições deste trabalho, identificando como estes foram adaptados para o domínio de processos de desenvolvimento.

O terceiro capítulo apresenta a primeira contribuição deste trabalho, apresentando o método *PrimeUp Assessment Method* (PAM) e definindo as atividades executadas nas duas fases propostas por ele: Avaliação em Abrangência e Avaliação em Profundidade.

No quarto capítulo é apresentada a segunda contribuição deste trabalho, uma ferramenta de apoio a realização de avaliações de risco em conformidade em processos de desenvolvimento e que suporta o método PAM, é apresentada. Inicialmente são discutidos os conceitos básicos da ferramenta Check-up Tool (Módulo, 2007), que foi utilizada como base para a elaboração desta ferramenta. Em seguida as customizações e adaptações são realizadas por este trabalho são identificadas e discutidas.

O quinto capítulo apresenta os estudos de caso realizados para comprovar a eficácia da utilização do método PAM e da ferramenta Check-up Tool customizada na avaliação de processos de desenvolvimento e na implantação de um programa de melhoria contínua nas organizações.

No sexto capítulo, trabalhos relacionados tanto com o método de avaliação proposto quanto com a ferramenta desenvolvida são apresentados e discutidos, comparando a solução proposta neste trabalho com as soluções já existentes.

Finalmente, no sétimo capítulo é apresentada a conclusão deste trabalho e possíveis trabalhos futuros são indicados como próximos passos.