

Nesta última parte do livro, consideraremos uma série de temas avançados que ampliarão seu entendimento sobre a engenharia de software. Nos próximos capítulos serão tratadas as seguintes questões:

- O que é a melhoria do processo de software e como pode ser usada para melhorar as práticas de engenharia de software?
- Quais as tendências emergentes que podem ter uma influência significativa sobre as práticas de engenharia de software na próxima década?
- Qual é a expectativa para os engenheiros de software?

Uma vez respondidas essas perguntas, você entenderá de assuntos que podem ter um profundo impacto sobre a engenharia de software nos próximos anos.

CAPÍTULO

30

MELHORIA DO PROCESSO DE SOFTWARE

CONCEITOS-CHAVE

avaliação	686
avaliação	689
CMMI	691
CMM das pessoas	695
educação e treinamento	687
fatores críticos de sucesso	690
gerenciamento de riscos	689
instalação / migração	689
justificação	688
modelos de maturidade	685
retorno sobre o investimento	697
seleção	688
melhora do processo de software (SPI)	685
aplicabilidade	686
estruturas	683
processo	686

Muito antes de estar na moda, a frase “melhoria do processo de software”, trabalhei em grandes corporações tentando melhorar a qualidade de suas práticas de engenharia de software. Como resultado de minhas experiências, escrevi o livro *Making software engineering happen*. O prefácio apresenta o seguinte comentário:

Durante os últimos dez anos tive a oportunidade de ajudar muitas grandes empresas a implementar práticas de engenharia de software. A tarefa é difícil e raramente ocorre tão tranquilamente quanto se gostaria — mas quando é bem-sucedida, os resultados são profundos. Os projetos de software têm maior possibilidade de se completarem a tempo. As comunicações entre todas as partes envolvidas no desenvolvimento de software são melhoradas. O nível de confusão e caos que com frequência prevalece para grandes projetos de software é reduzido substancialmente. O número de erros encontrados pelo cliente diminui substancialmente. A credibilidade da organização de software melhora. E a gerência tem um problema a menos com que se preocupar.

Mas nem tudo é um mar de rosas. Muitas empresas tentam implementar práticas de engenharia de software e desistem frustradas. Outras executam tudo pela metade e nunca verão os benefícios que acabamos de descrever. Há ainda outras que o fazem da forma linha-linha-dura, impositiva, o que resulta em rebelião declarada entre o pessoal técnico e os gerentes e subsequente perda de moral.

Embora essas palavras tenham sido escritas há mais de 20 anos, permanecem ainda válidas nos dias atuais.

Enquanto nos aproximamos da segunda década do século 21, muitas grandes organizações têm tentado “fazer acontecer a engenharia de software”. Algumas implementaram práticas individuais que ajudaram a melhorar a qualidade do produto e o prazo de entrega. Outras estabeleceram um processo de software “consolidado” que orienta suas atividades

PANORAMA

O que é? A melhoria da gestão de qualidade abrange uma série de atividades que levarão a um melhor processo de software e, em consequência, melhor qualidade do software fornecido com melhor prazo de entrega.

Quem realiza? As pessoas que aderem à melhoria da gestão de qualidade (*software process improvement* – SPI) são provenientes de três grupos: gerentes técnicos, engenheiros de software e indivíduos com responsabilidade pela garantia da qualidade.

Por que é importante? Algumas organizações de software têm pouco mais do que um processo de software específico. À medida que trabalham para melhorar suas práticas, devem resolver os pontos fracos do processo e tentar melhorar sua abordagem para o trabalho de software.

Quais são as etapas envolvidas? A abordagem para a SPI é iterativa e contínua, mas pode ser encarada em cinco etapas: (1) avaliação do processo de software atual, (2) educação e treinamento dos profissionais e gerentes, (3) seleção e justificação de elementos do processo, métodos de engenharia de software e ferramentas, (4) implementação do plano da SPI e (5) avaliação e ajuste com base nos resultados do plano.

Qual é o artefato? Embora haja muitos produtos SPI intermediários, o resultado final é um processo de software melhorado que leva a uma qualidade mais alta do software.

Como garantir que o trabalho foi realizado corretamente? O software que a sua organização produz será fornecido com menos defeitos, o trabalho em cada estágio do processo de software será reduzido e a entrega no prazo se tornará muito mais possível.

técnicas e de gerenciamento de projeto. Mas há aquelas que continuam a lutar com dificuldades. Suas práticas são às vezes boas e outras não, e o processo é improvisado. Ocasionalmente, o trabalho é espetacular, mas, na maior parte, cada projeto é uma aventura, e ninguém sabe se terminará bem ou mal.

Assim, qual desses dois grupos precisa de melhoria do processo de software? A resposta (que pode surpreendê-lo) é *ambos*. Os que tiveram sucesso ao fazer a engenharia de software acontecer não podem se tornar complacentes. Devem trabalhar continuamente para melhorar sua abordagem de engenharia de software. E os que lutam com dificuldades devem começar a tomar seus rumos em direção às melhorias.

30.1 O QUE É SPI?

PONTO-CHAVE

SPI implica um processo de software definido, uma abordagem organizacional e estratégia para melhoria.

"Grande parte da crise de software é autoimposta, segundo um CIO, 'Prefiro ter o software errado e não atrasado. Sempre podemos corrigir depois.'"

Mark Paulk

O termo *melhoria do processo de software* (*software process improvement* - SPI) envolve muitas coisas. Primeiro, implica que os elementos de um processo de software eficaz podem ser definidos de maneira eficaz; segundo, que uma abordagem organizacional existente para o desenvolvimento de software pode ser avaliada em relação aos elementos; e terceiro, que uma estratégia significativa para melhoria pode ser definida. A estratégia SPI transforma a abordagem existente para o desenvolvimento de software em algo mais focado, com melhor repetibilidade e mais confiável (em termos de qualidade de produto e prazo de entrega).

Como a SPI não é gratuita, deve produzir um retorno do investimento. O trabalho e o tempo necessário para implementar uma estratégia SPI deve se pagar de alguma maneira mensurável. Para tanto, os resultados do processo e prática melhorados devem levar a uma redução nos "problemas" de software que custam tempo e dinheiro. Eles devem reduzir: o número de defeitos que passam para os usuários finais, o volume de retrabalho causado por problemas de qualidade, os custos associados à manutenção e suporte do software (Capítulo 29) e os custos indiretos que ocorrem quando o software é entregue com atraso.

30.1.1 Abordagens para SPI

Embora a organização possa decidir por uma abordagem relativamente informal para SPI, a grande maioria escolhe dentre uma série de estruturas SPI. A *estrutura SPI* define (1) inúmeras características que devem estar presentes quando se deseja obter um processo de software eficaz, (2) um método para determinar se aquelas características estão presentes, (3) um mecanismo para resumir os resultados de qualquer avaliação, e (4) uma estratégia para ajudar a organização na implementação das características de processo consideradas fracas ou ausentes.

A estrutura SPI avalia a "maturidade" do processo de software de uma organização e fornece indicação qualitativa de um nível de maturidade; com frequência, aplica-se o termo "modelo de maturidade" (Seção 30.1.2). Essencialmente, a estrutura SPI abrange um modelo de maturidade que, por sua vez, incorpora uma série de indicadores de qualidade de processo que fornecem indicação geral da qualidade do processo que levará à qualidade do produto.

A Figura 30.1 fornece uma visão geral de uma estrutura SPI típica. São mostrados os elementos-chave da estrutura e suas relações uns com os outros.

Você deve observar que não há uma estrutura SPI universal. De fato, a estrutura SPI escolhida por uma organização reflete o grupo de pessoas que está apoiando o trabalho da SPI. Conradi [Con96] define seis diferentes grupos de suporte a SPI:

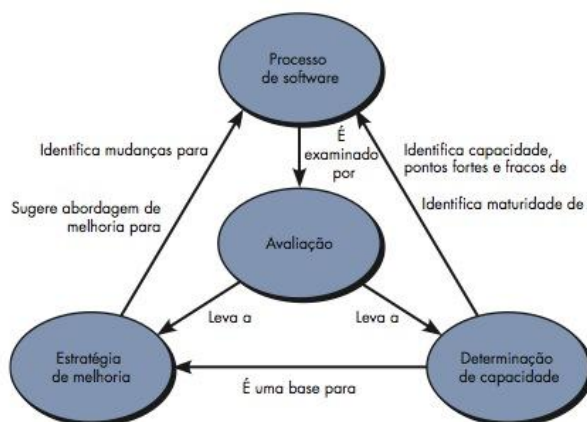
Certificadores de qualidade. Os esforços para melhoria de processo apoiados por esse grupo concentram-se na seguinte relação:

$$\text{Qualidade(Processo)} = \text{Qualidade(Produto)}$$

? "Quais são os grupos que apoiam um trabalho SPI?"

FIGURA 30.1**Elementos de uma estrutura SPI**

Fonte: Adaptado de (Rou02)



A abordagem desse grupo enfatiza métodos de avaliação e examina uma série bem definida de características que lhes permitem determinar se o processo apresenta qualidade. São mais propensos a adotar uma estrutura de processo do tipo CMM, SPICE, TickIT ou Bootstrap.¹

Formalistas. Esse grupo quer entender (e, quando possível, otimizar) o fluxo de trabalho de processo. Para tanto, usam linguagens de modelagem de processo (*process modeling languages* - PMLs) para criar um modelo do processo existente e então projetar extensões ou modificações que tornarão o processo mais eficaz.

Defensores das ferramentas. Esse grupo insiste em uma abordagem assistida por ferramenta para SPI, que modela o fluxo de trabalho e outras características de processo, de maneira a ser analisada para melhoria.

Profissionais. Esse grupo usa uma abordagem pragmática, "enfazando gerenciamento básico de projeto, qualidade e produto, aplicando planejamento e métricas em nível de projeto, mas com pouca modelagem formal de processo ou suporte" [Con96].

Reformadores. O objetivo desse grupo é a mudança organizacional que pode levar a um melhor processo de software. Eles tendem a concentrarem-se mais nos aspectos humanos (Seção 30.5) e enfatizam medidas de capacidade humana e estrutura.

Ideologistas. Esse grupo focaliza a adequação de um modelo particular de processo para um domínio de aplicação específico ou estrutura organizacional. Em vez de modelos de processo de software típicos (por exemplo, modelos iterativos), os ideologistas teriam um interesse maior em um processo que poderia, digamos, suportar reutilização ou reengenharia.

Quando uma estrutura SPI é aplicada, os grupos que a patrocinam (independentemente de seu foco geral) devem estabelecer mecanismos para: (1) suportar transição de tecnologia, (2) determinar o grau segundo o qual uma organização está pronta para absorver mudanças de processo propostas, e (3) medir o grau segundo o qual as mudanças foram adotadas.

“Quais são os diferentes grupos que suportam SPI?”

¹ Cada uma dessas estruturas SPI será discutida mais adiante neste capítulo.

30.1.2 Modelos de maturidade

Um *modelo de maturidade* é aplicado no contexto de uma estrutura SPI. Sua finalidade é proporcionar uma indicação geral da “maturidade do processo” exibida por uma organização de software. Por exemplo, a indicação da qualidade do processo de software, o grau segundo o qual os profissionais entendem e aplicam o processo, e o estado geral da prática de engenharia de software. Para tanto, utiliza-se algum tipo de escala ordinal.

Por exemplo, o *Capability Maturity Model* (Seção 30.4) do Software Engineering Institute sugere cinco níveis de maturidade [Sch96]:

Nível 5, Otimizado – A organização tem sistemas quantitativos de realimentação para identificar os pontos fracos do sistema para corrigi-los proativamente. As equipes de projeto analisam os defeitos para determinar suas causas; os processos de software são avaliados e atualizados para evitar que defeitos conhecidos voltem a ocorrer.

Nível 4, Controlado – Métricas detalhadas de processo de software e qualidade de produto estabelecem os fundamentos de avaliação quantitativa. As variações significativas no desempenho do processo podem ser distinguidas do ruído aleatório e previstas as tendências no processo e na qualidade do produto.

Nível 3, Definido – Processos para gerenciamento e engenharia são documentados, padronizados e integrados em um processo de software padrão para a organização. Todos os projetos usam uma versão aprovada, personalizada do processo de software padrão da organização para desenvolvimento de software.

Nível 2, Reprodutível – São estabelecidos processos básicos de gerenciamento de projeto para rastrear o custo, cronograma e funcionalidade. O planejamento e gerenciamento de novos produtos baseiam-se na experiência com projetos similares.

Nível 1, Inicial – Poucos processos são definidos, e o sucesso depende mais dos esforços heroicos individuais do que seguir um processo e usar um trabalho de equipe em sinergia.

A escala de maturidade CMM não vai além, mas a experiência indica que muitas organizações apresentam níveis de “imaturidade de processo” [Sch96] que enfraquece qualquer tentativa racional de melhorar as práticas de engenharia de software. Schorsch [Sch06] sugere quatro níveis de imaturidade muitas vezes encontrados nas organizações de desenvolvimento de software:

Nível 0, Negligente – Quando não se permite o processo de desenvolvimento bem-sucedido. Todos os problemas são considerados técnicos. As atividades gerenciais e de garantia de qualidade são consideradas complicadas e supérfluas para a tarefa do processo de desenvolvimento de software. Dependem de amuletos.

Nível – 1, Obstrutivo – Impõem-se processos contraproduativos. Os processos são rigidamente definidos e exagera-se na sua obediência. Rituais cerimoniais ocorrem em abundância. O gerenciamento coletivo inviabiliza a atribuição de responsabilidade. *Status quo über alles* (Por cima de todos).

Nível – 2, Arrogante – Desprezo completo pela boa engenharia de software. Uma separação total entre atividades de desenvolvimento de software e atividades de melhoria do processo de software. Ausência completa de um programa de treinamento.

Nível – 3, Sabotador – Negligência total do próprio compromisso, descrédito consciente dos esforços de melhoria do processo de software da organização. Falta de incentivo e mau desempenho.

Os níveis de imaturidade de Schorsch são prejudiciais para qualquer organização de software. Se você encontrar qualquer um deles, as tentativas de SPI estão fadadas ao fracasso.

A questão fundamental é se as escalas de maturidade, como a proposta como parte da CMM, proporcionam qualquer benefício real. Penso que sim. A escala de maturidade proporciona uma visão clara da qualidade do processo que pode ser utilizado pelos profissionais e gerentes, como um benchmark a partir do qual podem ser planejadas as estratégias de melhorias.

30.1.3 A SPI é para todos?

Por muitos anos, a SPI foi vista como uma atividade “corporativa” – eufemismo para algo que apenas grandes empresas executam. Mas nos dias atuais, uma porcentagem significativa de todo o desenvolvimento de software está sendo executada por empresas que empregam menos

PONTO-CHAVE

O modelo de maturidade define níveis de competência no processo de software e implementação.

“Como você reconhece uma organização que resistirá aos esforços de SPI?”



Se um modelo de processo específico ou uma abordagem SPI parecem ser opressivos para sua organização, provavelmente são.

de cem pessoas. Uma organização pequena pode empreender atividades de SPI e executá-las de maneira bem-sucedida?

Há diferenças culturais importantes entre organizações de desenvolvimento de software grandes e pequenas. Não é surpresa o fato de que pequenas organizações são mais informais, aplicam menos práticas padronizadas e tendem a ser auto-organizadas. Elas também tendem a se vangloriar da "criatividade" de membros da organização de software e encaram, inicialmente, uma estrutura SPI como demasiadamente burocrática e de grande peso. No entanto, a melhoria do processo é tão importante para a organização pequena quanto para a grande.

Em pequenas organizações, a implementação de uma estrutura SPI requer recursos que podem ser escassos. Os gerentes precisam alocar pessoas e dinheiro para fazer acontecer a engenharia de software. Portanto, independentemente do tamanho da organização de software, é razoável considerar a motivação comercial para a SPI.

A SPI será aprovada e implementada somente depois que seus proponentes demonstrarem uma alavancagem financeira [Bir98]. A alavancagem financeira é demonstrada examinando-se os benefícios técnicos (por exemplo, menos defeitos que passam para o campo, retrabalho reduzido, menores custos de manutenção, ou uma chegada ao mercado mais rápida) e traduzindo-os em dólares. Essencialmente, você deve mostrar um retorno realístico sobre o investimento (Seção 30.7) para os custos SPI.

30.2 O PROCESSO DE SPI

A parte difícil da SPI não é a definição de características que caracterizam um processo de software de alta qualidade ou a criação de um modelo de maturidade de processo. Essas coisas são relativamente fáceis. No entanto, a parte difícil é estabelecer um consenso para iniciar a SPI e definir uma estratégia contínua para implementá-la por meio de uma organização de software.

O Software Engineering Institute desenvolveu a IDEAL – “modelo de melhoria organizacional que serve de roteiro para iniciar, planejar e implementar ações de melhoria” [SEI08]. A IDEAL representa os muitos modelos de processo para SPI, definindo cinco atividades distintas – iniciar, diagnosticar, estabelecer, agir e aprender – que orientam uma organização com base nas atividades de SPI.

Neste livro, apresento um roteiro um pouco diferente para a SPI, baseado no modelo de processo para SPI proposto originalmente em [Pre88]. Ele aplica uma filosofia de bom senso que requer organização para que a empresa (1) se autoavalie, (2) depois se torne mais inteligente para que possa fazer escolhas inteligentes, (3) selecione o modelo de processo (e os elementos de tecnologia relacionados) que melhor satisfaça às suas necessidades, (4) crie uma instância do modelo em seu ambiente operacional e sua cultura, e (5) avalie o que foi feito. Essas cinco atividades (discutidas nas próximas subseções)² são aplicadas de maneira iterativa (cíclica) para fortalecer a melhoria contínua do processo.

30.2.1 Avaliação e análise de lacunas

Qualquer tentativa de melhorar o processo de software sem primeiro avaliar a eficácia das atividades da estrutura e práticas de engenharia de software associadas seria o mesmo que iniciar uma longa jornada a uma localidade sem nenhuma ideia de onde se deve partir. Você iniciará com grande entusiasmo, vai tentar encontrar seu rumo, vai gastar muita energia e suportar grandes doses de frustração e, provavelmente, decidirá que realmente quer ir a lugar nenhum. Resumindo, antes de começar sua jornada, é aconselhável saber precisamente onde você está.

² Parte do conteúdo dessas seções foi adaptado de [Pre88] com permissão.



Procure entender seus pontos fortes e fracos. Se for inteligente, trabalhará sobre os pontos fortes.

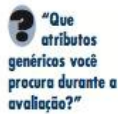
A primeira atividade do roteiro, chamada de *avaliação*, permite que você tome o seu rumo. A finalidade da avaliação é revelar os pontos fortes e fracos na maneira como sua organização aplica processos de software existentes e as práticas de engenharia de software que compõem o processo.

A avaliação examina uma grande variedade de ações e tarefas que levarão a um processo de alta qualidade. Por exemplo, independentemente do modelo de processo escolhido, a organização deve estabelecer mecanismos genéricos como: abordagens definidas para comunicação com o cliente; métodos estabelecidos para representar os requisitos do usuário; definição de uma estrutura de gerenciamento de projeto que inclui definição de escopo, estimativa, cronograma e rastreamento de projeto; métodos de análise de risco; procedimentos de gerenciamento de alterações; garantia de qualidade e atividades de controle, incluindo revisões; e muitas outras. Cada um desses mecanismos é considerado no contexto das atividades de estrutura e proteção (Capítulo 2) estabelecido e avaliado para determinar se uma das seguintes questões pode ser resolvida:

- O objetivo da ação está claramente definido?
- Os produtos requeridos como entrada e produzidos como saída estão identificados e descritos?
- Os trabalhos a ser realizados estão claramente descritos?
- As pessoas que devem executar as ações estão identificadas de acordo com suas funções?
- Os critérios de entrada e saída foram estabelecidos?
- As métricas para a ação foram estabelecidas?
- Há ferramentas disponíveis para suportar a ação?
- Há um programa de treinamento explícito que trata da ação?
- A ação é executada uniformemente para todos os projetos?

Embora as questões apresentadas impliquem uma resposta *sim* ou *não*, o papel da avaliação é examinar a resposta para determinar se a ação em questão está sendo executada conforme as melhores práticas.

À medida que é conduzida a avaliação do processo, você (ou aqueles que foram contratados para executar a avaliação) deverá também concentrar-se nos seguintes atributos:



Consistência. As atividades importantes, ações e tarefas são aplicadas de forma consistente em todos os projetos de software e por todas as equipes de software?

Sofisticação. As ações de gerência e técnicas são executadas com um nível de sofisticação que envolve completo entendimento das melhores práticas?

Aceitação. O processo de software e as práticas de engenharia de software são amplamente aceitos pela gerência e pessoal técnico?

Compromisso. A gerência forneceu os recursos necessários para obter consistência, sofisticação e aceitação?

A diferença entre aplicação local e melhor prática representa um “vazio” que oferece oportunidades para melhoria. O grau segundo o qual a consistência, sofisticação, aceitação e compromisso são atingidos indica o nível de mudança cultural necessário para atingir uma melhoria significativa.

30.2.2 Educação e treinamento

Embora poucos do mundo do software questionem o benefício de um processo de software ágil, organizado ou práticas de engenharia de software sólidas, muitos profissionais e gerentes não sabem o bastante sobre qualquer um dos assuntos.³ Consequentemente, percepções incorretas de processos e práticas levam a decisões inadequadas quando é introduzida uma estrutura

³ Se você está lendo este livro, não será um deles!

SPI. Conclui-se que o elemento-chave de qualquer estratégia SPI é a educação e o treinamento para os profissionais, gerentes técnicos e gerentes seniores que têm contato direto com a organização de software. Deverão ser promovidos três tipos de educação e treinamento:



Tente providenciar treinamento “just-in-time” dirigido para as reais necessidades da equipe de software.

Conceitos e métodos genéricos. Direcionada para gerentes e profissionais, essa categoria explora tanto o processo quanto a prática. A finalidade é fornecer aos profissionais as ferramentas intelectuais de que necessitam para aplicar o processo de software efetivamente e para tomar decisões racionais sobre as melhorias do processo.

Tecnologia e ferramentas específicas. Direcionada primariamente para os profissionais, essa categoria explora tecnologias e ferramentas adotadas para uso local. Por exemplo, se foi escolhida a UML para análise e modelagem de projeto, deve-se providenciar um currículo de treinamento para engenharia de software usando UML.

Comunicação comercial e tópicos relacionados à qualidade. Direcionada para todos os interessados, essa categoria concentra-se nos tópicos leves que ajudam a obter melhor comunicação entre os envolvidos e adotar um foco para melhor qualidade.

Em um contexto moderno, a educação e o treinamento podem ser fornecidos em uma variedade de maneiras diferentes. Tudo, desde podcasts a treinamento baseado na Internet (por exemplo, [QAI08]), até DVDs e cursos em salas de aulas, pode ser oferecido como parte de uma estratégia SPI.

30.2.3 Seleção e justificação



Ao fazer suas escolhas, não deixe de considerar a cultura da sua organização e o nível de aceitação que cada escolha poderá atingir.

Uma vez terminada a atividade de avaliação⁴ e iniciada a educação, uma organização de software deve começar a fazer suas escolhas, que ocorrem durante a *atividade de seleção e justificação* na qual se optam por características de processo e métodos e ferramentas específicos de engenharia de software para preencher o processo de software.

Primeiro você deverá escolher o modelo de processo (Capítulos 2 e 3) que melhor se adapte à sua organização, seus interessados e o software que você produz. Você deverá decidir qual o conjunto de atividades de estrutura a ser aplicado, os principais artefatos que serão produzidos e os pontos de verificação de garantia de qualidade que permitirão à sua equipe acompanhar o progresso. Se a atividade de avaliação de SPI indica uma deficiência específica (por exemplo, não há funções formais de SQA), você deverá concentrar-se nas características do processo que tratam diretamente dessas deficiências.

Em seguida, faça uma divisão do trabalho para cada atividade estrutural (por exemplo, modelagem), definindo o conjunto de tarefas que seriam aplicadas a um projeto típico. Considere também os métodos de engenharia de software que podem ser aplicados para executar essas tarefas. Conforme as escolhas forem feitas, a educação e o treinamento deverão ser coordenados para que o entendimento seja reforçado.

Idealmente, todos trabalham juntos para selecionar vários elementos de processo de tecnologia e poder direcionar-se para a atividade de instalação ou migração (Seção 30.2.4). Na realidade, a seleção pode ser um caminho complicado. Muitas vezes é difícil obter consenso entre os diferentes grupos. Se os critérios para seleção forem estabelecidos por meio de discussões, participantes podem argumentar indefinidamente sobre sua adequação e se determinada escolha realmente atende aos critérios estabelecidos.

Contudo uma má escolha pode prejudicar mais do que ajudar, mas “paralisia por análise” indica pouco progresso (se houver algum) e que os problemas de processo permanecem. Contanto que a característica de processo ou elemento de tecnologia tenha uma boa chance de atender às necessidades de uma organização, muitas vezes é melhor ser objetivo e fazer uma escolha do que esperar por uma solução ótima.

⁴ Na atualidade, a avaliação é uma atividade contínua. Ela é conduzida periodicamente para determinar se a estratégia SPI atingiu seus objetivos imediatos e preparar o cenário para melhorias futuras.

Uma vez feita a escolha, é preciso despendar tempo e dinheiro para fazê-la acontecer na organização, e essas despesas de recursos deverão ser justificadas. Na Seção 30.7 há uma discussão sobre a justificação do custo e o retorno sobre investimento em SPI.

30.2.4 Instalação/migração

Instalação é o primeiro ponto no qual uma organização de software sente os efeitos das mudanças implementadas em consequência do roteiro de SPI. Em alguns casos, é recomendado um processo inteiramente novo para a empresa. Atividades estruturais, ações de engenharia de software e tarefas de trabalho individual devem ser definidas e instaladas como parte de uma nova cultura de engenharia de software. Essas mudanças representam transição organizacional e tecnológica importante e devem ser administradas com muito cuidado.

Em outros casos, mudanças associadas com SPI são relativamente menos importantes, representando modificações pequenas, mas significativas a um modelo de processo existente. Essas mudanças muitas vezes são chamadas de *migração de processo*. Atualmente, muitas organizações de software têm um “processo” em andamento. O problema é que ele não funciona de maneira eficaz. Portanto, a *migração* incremental de um processo (que não funciona tão bem quanto se desejaria) para outro é uma estratégia mais eficaz.

A instalação e migração são na realidade atividades de *redesenho de processo de software* (*software process redesign* - SPR). Scacchi [Sca00] afirma que o “SPR está ligado à identificação, aplicação e refinamento de novas maneiras de melhorar radicalmente e transformar o processo de software”. Quando é iniciada uma abordagem formal ao SPR, consideram-se três diferentes modelos de processo: (1) o processo existente (“da forma como está”), (2) um processo de transição (“daqui para lá”), e (3) o processo-alvo (“o novo”). Se o processo-alvo for significativamente diferente do existente, a única abordagem racional para a instalação é uma estratégia incremental na qual o processo de transição é implementado em etapas. O processo de transição proporciona uma série de pontos intermediários que permitem à cultura da organização de software se adaptar a pequenas alterações durante um período.

30.2.5 Mensuração

Embora ela esteja listada como a última atividade no roteiro de SPI, a *mensuração* ocorre durante toda a SPI. A atividade de mensuração mede o grau segundo o qual as alterações foram criadas e adotadas, o grau segundo o qual essas alterações resultam em software de melhor qualidade ou outros benefícios de processo perceptíveis, e o estado geral do processo e a cultura da organização conforme a SPI progride.

Durante a atividade de mensuração são considerados fatores qualitativos e métricas quantitativas. Do ponto de vista qualitativo, as atitudes da gerência e dos profissionais no passado sobre o processo de software podem ser comparadas com as escolhidas após instalação das mudanças de processo. Métricas quantitativas (Capítulo 25) são coletadas de projetos que usaram o processo de transição ou “o novo” e comparadas com métricas similares coletadas para projetos executados de acordo com o processo “da forma como está”.

30.2.6 Gerenciamento de risco para SPI

A SPI é uma atividade de risco. Na verdade, mais da metade de todos os empreendimentos de SPI terminam em fracasso. As razões do fracasso variam muito e são específicas da organização. Entre os riscos mais comuns estão: falta de suporte gerencial, resistência cultural por parte do pessoal técnico, uma estratégia de SPI mal planejada, uma abordagem excessivamente formal à SPI, escolha de um processo inadequado, falta de interesse por parte dos principais interessados, orçamento inadequado, falta de treinamento do pessoal, instabilidade organizacional e uma infinidade de outros fatores. O papel dos responsáveis pela SPI é analisar os riscos prováveis e desenvolver uma estratégia interna para controlá-los.

PONTO-CHAVE

A SPI muitas vezes falha porque os riscos não foram considerados adequadamente e não houve planejamento de contingência.

Uma organização de software deve administrar o risco em três pontos-chave no processo de SPI [Sta97b]: antes de iniciar o roteiro da SPI, durante a execução das atividades de SPI (avaliação, educação, seleção, instalação) e durante a atividade de avaliação que segue à ocorrência de alguma característica de processo. Em geral, as seguintes categorias podem ser identificadas [Sta97b] para os fatores de risco da SPI: orçamento e custos, conteúdo e resultados práticos, cultura, manutenção de resultados práticos de SPI, missão e metas, gerenciamento organizacional, estabilidade organizacional, interessados no processo, cronograma para o desenvolvimento da SPI, ambiente de desenvolvimento da SPI, processo de desenvolvimento da SPI, gerenciamento de projeto da SPI e pessoal para a SPI.

Em cada categoria, há muitos fatores de risco genéricos. Por exemplo, a cultura organizacional tem uma forte influência sobre o risco. Podem ser identificados os seguintes fatores de risco genéricos⁵ para a categoria cultura [Sta97b]:

- Atitude em relação à mudança, com base em esforços anteriores para mudar
- Experiência com programas de qualidade, nível de sucesso
- Orientação de ações para resolver os problemas *versus* debates políticos
- Uso de fatos para administrar a organização e os negócios
- Paciência com as mudanças, habilidade em empregar o tempo para a socialização
- Orientação para as ferramentas – a expectativa de que as ferramentas possam resolver os problemas
- Nível de “totalidade de planejamento” – habilidade da organização em planejar
- Habilidade dos membros da organização em participar com vários níveis da organização abertamente nas reuniões
- Habilidade dos membros da organização em administrar as reuniões eficientemente
- Nível de experiência em organização com processos definidos

Usando como guia fatores de risco e atributos genéricos, a exposição ao risco é calculada da seguinte maneira:

$$\text{Exposição} = (\text{probabilidade do risco}) \times (\text{perda estimada})$$


Pode ser elaborada uma tabela de riscos (Capítulo 28) para isolar aqueles que requerem mais atenção da gerência.

30.2.7 Fatores críticos de sucesso

Na Seção 30.2.6, destacou-se que a SPI é um empenho arriscado e que a taxa de falha das empresas que tentam melhorar seus processos é assustadoramente alta. Riscos organizacionais, riscos pessoais e riscos de gerenciamento de projeto apresentam desafios para aqueles que empreendem qualquer esforço em SPI. Embora o gerenciamento de risco seja importante, é igualmente importante reconhecer os fatores críticos que levam ao sucesso.

Após examinar 56 organizações de software e seus esforços em SPI, Stelzer e Mellis [Ste99] identificam uma série de fatores críticos de sucesso (*critical success factors* – CSFs) que precisam estar presentes para que a SPI tenha êxito. Nesta seção apresentaremos os cinco principais CSFs.

Comprometimento e suporte da gerência. Assim como a maioria das atividades que causam mudanças organizacionais e culturais, a SPI terá êxito apenas se a gerência estiver envolvida ativamente. Os membros da alta direção deverão reconhecer a importância do software para a empresa e ser patrocinadores ativos do esforço para a SPI. Os gerentes técnicos deverão estar profundamente envolvidos no desenvolvimento da estratégia SPI local. Conforme observam os autores do estudo: “A melhoria do processo de software não é viável

 Que fatores críticos de sucesso são cruciais para uma SPI bem-sucedida?

⁵ Fatores de risco para cada uma das categorias de risco descritas nessa seção podem ser encontrados em [Sta97b].

sem investir tempo, dinheiro e esforços” [Ste99]. O comprometimento e apoio da gerência são essenciais para sustentar esse investimento.

Envolvimento do pessoal. A SPI não pode ser imposta de cima para baixo nem pode ser imposta de fora. Quando se deseja que os esforços em SPI sejam bem-sucedidos, a melhora tem de ser orgânica — patrocinada por gerentes técnicos e tecnólogos seniores e adotada por profissionais locais.

Integração e entendimento do processo. O processo de software não existe em um vácuo organizacional. Deve ser caracterizado de maneira que esteja integrado a outros processos e requisitos comerciais. Para tanto, os responsáveis pelo trabalho de SPI devem ter um conhecimento e entendimento dos outros processos comerciais. Além disso, devem entender o processo de software “da maneira como está” e apreciar quanta mudança de rumo for tolerável na cultura local.

Uma estratégia SPI personalizada. Não há uma estratégia de SPI genérica. Conforme mencionado neste capítulo, o roteiro da SPI deve ser adaptado ao ambiente local — devem ser considerados a cultura da equipe, variedades de produtos, pontos fortes e fracos locais.

Sólido gerenciamento do projeto de SPI. SPI é um projeto como qualquer outro. Envolve coordenação, cronograma, tarefas paralelas, resultados práticos, adaptação (quando os riscos se tornam realidade), políticas, controle de orçamento e muito mais. Sem um gerenciamento ativo e eficaz, um projeto de SPI está fadado ao fracasso.

30.3 A CMMI

WebRef

Informações completas sobre a CMMI podem ser obtidas no site www.sei.cmu.edu/cmmi/.

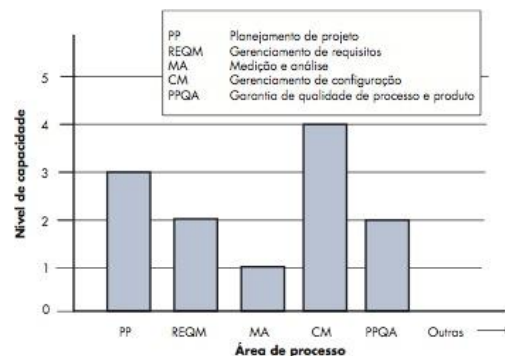
A CMM original foi desenvolvida e atualizada pelo Software Engineering Institute na década de 1990 como uma estrutura SPI completa. Hoje, evoluiu tornando-se a *Capability Maturity Model Integration* (CMMI) [CMM07], um metamodelo de processo abrangente qualificado em uma série de capacidades de sistema e engenharia de software que devem estar presentes à medida que as organizações alcançam diferentes níveis de capacidade e maturidade de processo.

A CMMI representa um metamodelo de processo de duas maneiras diferentes: (1) como um modelo “contínuo” e (2) como um modelo “encenado”. O metamodelo CMMI contínuo descreve um processo em duas dimensões conforme está ilustrado na Figura 30.2. Cada área de processo (por exemplo, planejamento de projeto ou gerenciamento de requisitos) é formalmente avaliada

Figura 30.2

Perfil de capacidade de área de processo CMMI

Fonte: (Phi02)



em relação a metas e práticas específicas e classificada de acordo com os seguintes níveis de capacidade:

Nível 0: Incompleto – a área de processo (por exemplo, gerenciamento de requisitos) não funciona ou não atinge todas as metas e objetivos definidos pela CMMI para a capacidade nível 1 para a área de processo.

Nível 1: Executado – todas as metas específicas da área de processo (definida pela CMMI) foram satisfeitas. Estão sendo executadas as tarefas necessárias para produzir os artefatos definidos.

Nível 2: Controlada – todos os critérios do nível de capacidade 1 foram satisfeitos. Além disso, todo o trabalho associado com a área de processo está de acordo com uma política definida em termos de organização; todas as pessoas que estão fazendo o trabalho têm acesso a recursos adequados para executar o trabalho; os interessados são envolvidos ativamente na área de processo conforme necessário; todas as tarefas e produtos são “monitorados, controlados, e revisados; e são avaliados quanto à conformidade com a descrição de processo” [CMM07].

Nível 3: Definido – todos os critérios do nível de capacidade 2 foram satisfeitos. Além disso, o processo é “adaptado com base no conjunto de processos padronizados da organização de acordo com as regras de adaptação da organização, e dos produtos acabados, medidas e outras informações de melhoria de processo para agregar valores ao processo organizacional” [CMM07].

Nível 4: Controlado quantitativamente – todos os critérios do nível de capacidade 3 foram satisfeitos. Além disso, a área de processo é controlada e melhorada usando medição e avaliação quantitativa. “São estabelecidos objetivos quantitativos para qualidade e desempenho de processo e utilizados como critérios no controle do processo” [CMM07].

Nível 5: Otimizado – todos os critérios do nível de capacidade 4 foram satisfeitos. Além disso, a área de processo é adaptada e otimizada usando meios quantitativos (estatísticos) para atender à mudança de necessidades do cliente e melhorar continuamente a eficiência da área de processo em consideração.

A CMMI define cada área de processo em termos de “metas específicas” e as “práticas específicas” necessárias para atingir essas metas. As *metas específicas* estabelecem as características que devem existir para que as atividades envolvidas por uma área de processo sejam eficazes. *Práticas específicas* refinam uma meta transformando-a em uma série de atividades relacionadas ao processo.

Por exemplo, **planejamento de projeto** é uma das oito áreas de processo definidas pela CMMI para a categoria “gerenciamento de projeto”.⁶ As metas específicas (*specific goals* – SG) e as práticas específicas associadas (*specific practices* – SP), definidas para o **planejamento de projeto** são [CMM07]:

SG 1 Estabelecer Estimativa

- SP 1.1-1 Estimar o Escopo do Projeto
- SP 1.2-1 Estabelecer Estimativas de Atributos de Produto e Tarefa
- SP 1.3-1 Definir o Ciclo de Vida do Projeto
- SP 1.4-1 Determinar Estimativas de Trabalho e Custo

SG 2 Desenvolver um Plano de Projeto

- SP 2.1-1 Estabelecer o Orçamento e o Cronograma
- SP 2.2-1 Identificar os Riscos do Projeto



Toda organização deve lutar para atingir o objetivo da CMMI. No entanto, a implementação de todo o aspecto do modelo pode ser desastrosa na sua situação.

WebRef

Informações completas, bem como download de uma versão da CMMI, podem ser obtidos no site www.sei.cmu.edu/cmmi/.

⁶ Outras áreas de processo para “gerenciamento de projeto” incluem: monitoramento e controle de projeto, gerenciamento de acordos com fornecedores, gerenciamento integrado de projeto para IPPD, gerenciamento de risco, equipe integrada, gerenciamento integrado de fornecedor e gerenciamento quantitativo de projeto.

- SP 2.3-1 Planejar o Gerenciamento de Dados
- SP 2.4-1 Elaborar Plano para Recursos de Projeto
- SP 2.5-1 Elaborar Plano para Conhecimento e Habilidades Necessárias
- SP 2.6-1 Elaborar Plano para Envolvimento dos Interessados
- SP 2.7-1 Estabelecer o Plano de Projeto

SG 3 Obter Comprometimento com o Plano

- SP 3.1-1 Rever Planos Que Afetam o Projeto
- SP 3.2-1 Reconciliar Trabalho e Níveis de Recursos
- SP 3.3-1 Obter Comprometimento com o Plano

Além das metas e práticas específicas, a CMMI também define um conjunto de cinco metas genéricas e práticas relacionadas a cada área de processo. Cada uma delas corresponde a um dos cinco níveis de capacidade. Portanto, para atingir determinado nível de capacidade, a meta genérica para aquele nível e as práticas genéricas que correspondem àquela meta devem ser atingidas. As metas genéricas (*generic goals* – GG) e práticas genéricas (*generic practices* – GP) para a área de processo de **planejamento de projeto** são [CMM07]:

GG 1 Atingir Metas Específicas

- GP 1.1 Executar Práticas Básicas

GG 2 Instituir um Processo Controlado

- GP 2.1 Estabelecer uma Política Organizacional
- GP 2.2 Planejar o Processo
- GP 2.3 Fornecer os Recursos
- GP 2.4 Atribuir Responsabilidades
- GP 2.5 Treinar o Pessoal
- GP 2.6 Gerenciar Configurações
- GP 2.7 Identificar e Envolver Interessados Importantes
- GP 2.8 Monitorar e Controlar o Processo
- GP 2.9 Avaliar Objetivamente a Adesão
- GP 2.10 Rever o Status com a Alta Gerência

GG 3 Instituir um Processo Definido

- GP 3.1 Estabelecer um Processo Definido
- GP 3.2 Coletar Informações de Melhorias

GG 4 Instituir um Processo Controlado Quantitativamente

- GP 4.1 Estabelecer Objetivos Quantitativos do Processo
- GP 4.2 Estabilizar Desempenho de Subprocesso

GG 5 Instituir um Processo de Otimização

- GP 5.1 Garantir a Melhora Contínua do Processo
- GP 5.2 Corrigir as Causas Importantes dos Problemas

O modelo CMMI apresentado define as mesmas áreas de processo, metas e práticas do modelo contínuo. A diferença principal é que o modelo apresentado define cinco níveis de maturidade, em vez de cinco níveis de capacidade. Para atingir um nível de maturidade, as metas específicas e as práticas associadas com uma série de áreas de processo devem ser atingidas. A relação entre níveis de maturidade e áreas de processo é apresentada na Figura 30.3.

INFORMAÇÕES

**A CMMI – Fazemos ou não fazemos?**

A CMMI é um metamodelo de processo. Ela define (em mais de 700 páginas) as características de processo que devem existir se uma organização de software quiser estabelecer um processo de software completo. A questão debatida por mais de uma década é: “A CMMI é opressiva?”. Como muitas coisas na vida (e em software), a resposta não é um simples “sim” ou “não”.

O espírito da CMMI deverá ser sempre adotado. Com o risco da simplificação, argumenta que o desenvolvimento de software deve ser encarado com seriedade – deve ser planejado amplamente, precisa ser controlado com uniformidade, deve ser acompanhado com precisão e conduzido profissionalmente. Deve concentrar-se nas necessidades dos patrocinadores/interessados pelo projeto, nas habilidades dos engenheiros de software e na qualidade do produto final. Essas ideias são indiscutíveis.

Os requisitos detalhados da CMMI deverão ser considerados seriamente se uma organização cria sistemas grandes e

complexos que envolvem dezenas ou centenas de pessoas por muitos meses ou anos. Pode ocorrer que a CMMI seja “na medida certa” em tais situações, se a cultura organizacional for favorável a modelos de processo-padrão e a gerência estiver comprometida em torná-la um sucesso. No entanto, em outras situações, a CMMI pode ser demais para uma organização poder assimilá-la. Isso significa que a CMMI é “ruim” ou “demasiadamente burocrática” ou “à moda antiga?”. Não ... não é. Significa que o que é certo para uma cultura organizacional pode não ser certo para outra.

A CMMI é uma conquista significativa na engenharia de software. Proporciona uma discussão abrangente das atividades e ações que devem existir quando uma organização cria software de computador. Mesmo que a empresa prefira não adotar seus detalhes, toda equipe de software deveria aderir a esse espírito e aprender com a discussão de processo e prática de engenharia de software.

FIGURA 30.3

Áreas de processo necessárias para atingir um nível de maturidade

Fonte: [Ph102]

Nível	Foco	Áreas de Processo
Otimizante	Melhoria contínua do processo	Inovação organizacional e distribuição (deployment) Análise causal e resolução
Controlado quantitativamente	Gerenciamento quantitativo	Desempenho de processo organizacional Gerenciamento quantitativo de projeto
Definido	Padronização de processo	Desenvolvimento de requisitos Solução técnica Integração de produto Verificação Validação Foco no processo organizacional Definição de processo organizacional Treinamento organizacional Gerenciamento de projeto integrado Gerenciamento de fornecimento integrado Gerenciamento de risco Análise de decisão e resolução Ambiente organizacional para integração Equipe integrada
Repetível	Gerenciamento básico de projeto	Gerenciamento de requisitos Planejamento de projeto Monitoração e controle de projeto Gerenciamento de acordo com fornecedor Medição e análise Garantia de qualidade de processo e produto Gerenciamento de configuração
Executado		

30.4 A CMM DAS PESSOAS

PONTO-CHAVE

A CMM das Pessoas sugere práticas que melhoram a competência e cultura da força de trabalho.

Um processo de software, não importa quão bem seja concebido, não terá sucesso sem profissionais talentosos e motivados. O *Modelo de Maturidade da Capacidade das Pessoas* (*People Capability Maturity Model* – CMM) “é um roteiro para implementar práticas de trabalho que aperfeiçoam continuamente a capacidade dos profissionais de uma organização” [Cur02]. Desenvolvida no decorrer da década de 1990 e refinada nos anos seguintes, o objetivo da CMM das Pessoas (*People CMM*) é estimular a melhoria contínua do conhecimento genérico da força de trabalho (chamado de “competências núcleo”), habilidades específicas de engenharia de software e gerenciamento de projeto (chamado de “competências da força de trabalho”) e habilidades relacionadas ao processo.

Assim como a CMM, a CMMI e estruturas SPI relacionadas, a CMM das Pessoas define um conjunto de cinco níveis de maturidade organizacional que proporcionam uma indicação da sofisticação relativa das práticas e processos da força de trabalho. Esses níveis de maturidade [CMM08] estão ligados à existência (dentro de uma organização) de um conjunto de áreas-chave de processo (*key process areas* – KPAs). Uma visão geral dos níveis organizacionais e KPAs relacionadas está na Figura 30.4.

A CMM das Pessoas complementa qualquer estrutura SPI estimulando uma organização a cultivar e melhorar seu bem mais importante – as pessoas. Sendo importante, estabelece uma atmosfera na força de trabalho que permite a organização de software “atrair, desenvolver e preservar talentos notáveis” [CMM08].

FIGURA 30.4

Áreas de processo para CMM das Pessoas

Nível	Foco	Áreas de Processo
Otimizante	<i>Melhoria contínua</i>	Inovação contínua da força de trabalho Alinhamento do desempenho organizacional Melhora contínua da capacidade
Gerenciado	<i>Identifica e desenvolve conhecimento, prática e habilidades</i>	Tutela Gerenciamento da capacidade organizacional Gerenciamento do desempenho quantitativo Propriedades baseadas na competência Grupos de trabalho fortalecidos Integração de competência
Definido	<i>Quantifica e administra conhecimento, prática e habilidades</i>	Cultura participativa Desenvolvimento de grupo de trabalho Práticas baseadas na competência Desenvolvimento de carreira Desenvolvimento de competência Planejamento da força de trabalho Análise de competência
Repetível	<i>Práticas básicas de gerenciamento de pessoas que podem ser repetidas</i>	Compensação Treinamento e desenvolvimento Gerenciamento de desempenho Ambiente de trabalho Comunicação e coordenação de pessoal
Inicial	<i>Práticas inconsistentes</i>	

30.5 OUTRAS ESTRUTURAS SPI

Embora a CMM e CMMI do Software Engineering Institute (SEI) sejam as estruturas SPI mais amplamente aplicadas, têm sido propostas e estão em uso muitas alternativas⁷. Entre elas, destacam-se:

- **SPICE** – iniciativa internacional para suportar a avaliação de processo da ISO e padrões de processo de ciclo de vida [SP199]
- **ISO/IEC 15504** para Avaliação de Processo (Software) [ISO08]
- **Bootstrap** – uma estrutura SPI para organizações de pequeno e médio porte em conformidade com SPICE [Boo06]
- **PSP e TSP** – estruturas SPI individuais e específicas de equipe ([Hum97], [Hum00]) que se concentram no processo em detalhes, uma abordagem mais rigorosa do desenvolvimento de software combinada com a medição
- **TickIT** – um método de auditoria [Tic05] que avalie se uma organização está em conformidade com a Norma ISO 9001:2000

Nos próximos parágrafos há uma descrição resumida das estruturas SPI. Se houver interesse, para cada uma delas há uma grande variedade de material impresso e baseado na Web.

Além da CMM, há outras estruturas SPI que podem ser consideradas?

SPICE. O modelo SPICE (*Software Process Improvement and Capability dEtermination* - *Melhoria de Processo de Software e Determinação de Capacidade*) proporciona uma estrutura de avaliação SPI compatível com a ISO 15504:2003 e ISO 12207. O conjunto de documentos SPICE [SDS08] apresenta uma estrutura SPI completa, incluindo um modelo para gerenciamento de processo, diretrizes para conduzir uma avaliação e classificação do processo em consideração, construção, seleção e uso dos instrumentos e ferramentas de avaliação e treinamento para os avaliadores.

Bootstrap. A estrutura SPI *Bootstrap* "foi desenvolvida para assegurar a conformidade com a norma ISO emergente para avaliação e melhoria de processo de software (SPICE) e para alinhar a metodologia com a ISO 12207" [Boo06]. O objetivo da *Bootstrap* é examinar um processo de software usando um conjunto de melhores práticas de engenharia de software como base para a avaliação. Assim como a CMMI, a *Bootstrap* proporciona um nível de maturidade de processo por meio dos resultados de questionários que reúnem informações sobre o processo de software e projetos de software "da forma como está". As diretrizes SPI baseiam-se no nível de maturidade e nas metas organizacionais.

PSP e TSP. Embora a SPI seja caracterizada em geral como uma atividade organizacional, não há razão para que uma melhora de processo não possa ser conduzida em nível individual ou de equipe. Tanto a PSP quanto a TSP (Capítulo 2) enfatizam a necessidade de continuamente coletar dados sobre o trabalho que está em execução e usar esses dados para desenvolver estratégias para melhorias. Watts Humphrey [Hum97], o desenvolvedor de ambos os métodos, comenta:

A PSP [e TSP] lhe mostrarão como planejar e acompanhar o seu trabalho e como produzir consistentemente software de alta qualidade. Usando PSP [e TSP] você obterá os dados que indicam a eficácia do seu trabalho e que identificam seus pontos fortes e fracos... Para uma carreira bem-sucedida e bem remunerada, você precisa conhecer suas experiências e habilidades, lutar para melhorá-las e concentrar seu talento exclusivamente no trabalho.

TickIT. O método de auditoria Ticket assegura a conformidade com a norma ISO 9001:2000 para Software – uma norma genérica que se aplica a qualquer organização que queira melhorar

"As organizações de software têm mostrado deficiências significativas em tirar proveito das experiências adquiridas com projetos executados."

NASA

⁷ É razoável argumentar que algumas dessas estruturas não são muito "alternativas" já que são abordagens complementares à SPI. Uma tabela abrangente de muitas outras estruturas SPI pode ser encontrada no site www.geocities.com/ibu_measure/spi/spi.htm#p2.

a qualidade geral dos produtos, sistemas ou serviços que ela fornece. Portanto, a norma é diretamente aplicável a organizações e empresas de software.

A estratégia subjacente sugerida pela ISO 9001:2000 é descrita da seguinte maneira [ISO01]:

ISO 9001:2000 destaca a importância para uma organização em identificar, implementar, administrar e melhorar continuamente a eficácia dos processos necessários para o sistema de gerenciamento de qualidade, e para administrar as interações desses processos com o fim de atingir os objetivos da organização... A eficácia e eficiência do processo podem ser avaliadas por meio de processos de revisão interna e externa e podem ser avaliadas em uma escala de maturidade.

WebRef

Um excelente resumo da ISO 9001:2000 pode ser encontrado no site <http://proxim.com/iso-9001.htm>.

A norma ISO 9001:2000 adotou um ciclo "planejar-fazer-verificar-agir" aplicado aos elementos de gerenciamento de qualidade de um projeto de software. Em um contexto de software, o "planejar" estabelece os objetivos do processo, atividades e tarefas necessárias para obter software de alta qualidade e como resultado a satisfação do cliente. "Fazer" implementa o processo de software (incluindo tanto atividades de estrutura quanto de apoio). "Verificar" monitora e mede o processo para garantir que todos os requisitos estabelecidos para gerenciamento da qualidade tenham sido atingidos. "Agir" inicia as atividades de melhoramento do processo de software que contribuem continuamente para melhorar o processo. TickIT pode ser usado em todo o ciclo "planejar-fazer-verificar-agir" para assegurar que o progresso SPI esteja ocorrendo. Os auditores TickIT avaliam a aplicação do ciclo como um precursor da certificação ISO 9001:2000. Para uma discussão detalhada da ISO 9001:2000 e TickIT, consulte [Ant06], [Tri05], ou [Sch03].

30.6 RETORNO SOBRE INVESTIMENTO EM SPI

A SPI é um trabalho pesado e requer investimento substancial em dinheiro e profissionais. Os administradores que aprovam o orçamento e recursos destinados à SPI invariavelmente fazem a seguinte pergunta: "Como posso saber se obtivemos um retorno razoável do dinheiro que gastamos?"

Em nível qualitativo, os proponentes da SPI argumentam que um processo de software melhorado levará a uma qualidade superior. Eles sustentam que um processo melhorado resultará na implementação de filtros de qualidade superior (resultando em menor propagação de defeitos), melhor controle das alterações (resultando em menor caos no projeto) e menos retrabalho técnico (resultando em menor custo e melhor prazo de entrega para o mercado). Mas podem esses benefícios qualitativos ser traduzidos em resultados quantitativos? A equação clássica do retorno sobre o investimento (*return on investment* – ROI) é:

$$ROI = \left[\frac{\Sigma(\text{benefícios}) - \Sigma(\text{custos})}{\Sigma(\text{custos})} \right] \times 100\%$$

em que

benefícios incluem as economias em custos associados a melhor qualidade do produto (menos defeitos), menos retrabalho, redução do trabalho associado com alterações e lucro proveniente de uma entrega mais rápida ao mercado.

custos incluem custos SPI diretos (por exemplo, treinamento, medição) e custos indiretos associados a maior ênfase no controle de qualidade e atividades de gerenciamento de mudanças e uma aplicação mais rigorosa dos métodos de engenharia de software (por exemplo, a criação de um modelo de projeto).

Na prática, esses benefícios quantitativos e custos são muitas vezes difíceis de medir com precisão, e todos dependem de uma interpretação. Mas isso não significa que uma organização de software deverá executar um programa SPI sem cuidadosa análise dos custos e benefícios

que se acumulam. Um tratamento abrangente do retorno de investimento para a SPI pode ser encontrado em um livro especial de David Rico [Ric04].

30.7 TENDÊNCIAS DA SPI

Durante as últimas duas décadas, muitas empresas tentaram melhorar suas práticas de engenharia de software aplicando uma estrutura SPI para promover mudança organizacional e transição de tecnologia. Conforme observamos antes neste capítulo, mais da metade fracassou nessa tentativa. Independentemente do sucesso ou falha, tudo custa muito. David Rico [Ric04] relata que uma aplicação típica de uma estrutura SPI como a SEI CMM pode custar entre \$ 25 mil e \$ 70 mil por pessoa e levar anos para se completar! Não é surpresa o fato de que o futuro da SPI deve dar ênfase a uma abordagem menos custosa e menos demorada.

Para ser eficaz no mundo do desenvolvimento de software do século 21, as futuras estruturas SPI devem se tornar significativamente mais ágeis. Em vez de um foco organizacional (que pode levar anos para se completar com sucesso), os esforços de SPI contemporâneos devem concentrar-se no nível de projeto, trabalhando para melhorar um processo de equipe em semanas, não em meses ou anos. Para obter resultados significativos (mesmo em nível de projeto) em curto prazo, modelos complexos de estrutura podem ser substituídos por modelos mais simples. Em lugar de dezenas de práticas-chave e centenas de práticas suplementares, uma estrutura SPI ágil deve dar ênfase apenas a poucas práticas críticas (por exemplo, análogas às atividades de estrutura discutidas neste livro).

Qualquer tentativa em SPI demanda uma força de trabalho com conhecimento, mas as despesas com educação e treinamento podem ser altas e deverão ser minimizadas (e enxugadas). Em vez de cursos em salas de aula, (caros e demorados), trabalhos de SPI futuros deverão utilizar o treinamento baseado na Web, destinado a práticas críticas. Ao contrário de tentativas avançadas para mudar a cultura organizacional (com todos os perigos políticos que surgem), a mudança cultural deverá ocorrer com um pequeno grupo de cada vez até alcançar um estágio importante.

O trabalho de SPI das duas últimas décadas tem um mérito significativo. As estruturas e modelos desenvolvidos representam valores intelectuais importantes para a comunidade de engenharia de software. Mas esses valores direcionam futuras tentativas em SPI não se tornando dogma recorrente, mas servindo como base para modelos SPI melhores, mais simples e mais ágeis.

30.8 RESUMO

Uma estrutura para melhorar o processo de software define as características que devem estar presentes quando se quer atingir um processo de software eficaz, um método de avaliação que ajude a determinar se aquelas características estão presentes e uma estratégia para ajudar a organização de software na implementação das características de processo consideradas fracas ou ausentes. Independentemente do grupo que apoia a SPI, o objetivo é melhorar a qualidade do processo e, como consequência, melhorar a qualidade do software e os prazos.

O modelo de maturidade de processo proporciona uma indicação geral da "maturidade de processo" exibida por uma organização de software. Proporciona uma sensação qualitativa sobre a eficiência relativa do processo de software que está em uso no momento.

O roteiro da SPI começa com a avaliação – uma série de atividades que revelam os pontos fracos e fortes como sua organização aplica o processo de software e as práticas de engenharia de software que fazem parte do processo. Desse modo, a empresa pode desenvolver um plano geral de SPI.

Um dos elementos-chave de qualquer plano SPI é a educação e o treinamento, uma atividade que se concentra na melhoria do nível de conhecimento dos gerentes e profissionais. Depois que

o pessoal se torna experiente nas tecnologias de software atuais, começa a seleção e a justificação. Essas tarefas levam a escolhas sobre a arquitetura do processo de software, os métodos que fazem parte dela e as ferramentas que a suportam. Instalação e avaliação são atividades de SPI que causam mudanças no processo e examinam sua eficácia e impacto.

Para melhorar o processo de software, a organização deve possuir as seguintes características: comprometimento e suporte da gerência para com a SPI, envolvimento do pessoal durante todo o processo de SPI, integração do processo na cultura organizacional geral, uma estratégia SPI que tenha sido personalizada para as necessidades locais e uma administração sólida do projeto de SPI.

Hoje há uma série de estruturas SPI em uso: CMM e CMMI da SEI são amplamente utilizadas. A People CMM foi personalizada para avaliar a qualidade da cultura organizacional e os profissionais envolvidos. SPICE, Bootstrap, PSP, TSP e TickIT são estruturas adicionais que podem levar a uma SPI eficaz.

SPI é trabalho sério e requer investimento substancial em dinheiro e pessoas. Para garantir a obtenção de um retorno sobre o investimento razoável, a organização deve medir os custos associados com a SPI e os benefícios que podem ser atribuídos.

PROBLEMAS E PONTOS A PONDERAR

- 30.1.** Por que as organizações de software usualmente lutam com dificuldade quando aderem a um esforço para melhorar o processo local de software?
- 30.2.** Descreva o conceito de “maturidade de processo” com suas próprias palavras.
- 30.3.** Faça uma pesquisa (verifique no site da SEI) e determine a distribuição da maturidade de processo para organizações de software nos Estados Unidos e no mundo todo.
- 30.4.** Você trabalha para uma pequena organização de software – apenas 11 pessoas estão envolvidas no desenvolvimento de software. A SPI é adequada a sua empresa? Explique sua resposta.
- 30.5.** Avaliação é semelhante a um exame anual de saúde. Usando essa metáfora, descreva a atividade de avaliação SPI.
- 30.6.** Qual a diferença entre um processo “como está”, um processo “desta para próxima etapa”, e um processo “vir a ser”?
- 30.7.** Como é aplicado o gerenciamento de risco no contexto da SPI?
- 30.8.** Selecione um dos fatores críticos de sucesso na Seção 30.2.7. Faça uma pesquisa e escreva um pequeno artigo sobre como pode ser alcançado.
- 30.9.** Faça uma pesquisa e explique como a CMMI difere de sua antecessora, a CMM.
- 30.10.** Selecione uma das estruturas SPI discutidas na Seção 30.5, e redija um pequeno artigo descrevendo-a em mais detalhes.

LEITURAS E FONTES DE INFORMAÇÃO COMPLEMENTARES

Um dos recursos mais facilmente acessíveis e abrangentes de informações sobre SPI foi desenvolvido pelo Software Engineering Institute e está disponível no site www.sei.cmu.edu. O site do SEI contém centenas de artigos, estudos e descrições detalhadas de estruturas SPI.

Durante os últimos anos, muitos bons livros foram acrescentados a uma ampla literatura desenvolvida durante as duas últimas décadas. Land (*Jumpstart CMM/CMMI Software Process Improvements*, Wiley-IEEE Computer Society, 2007) apresenta os requisitos definidos como parte dos padrões de engenharia de software SEI CMM e CMMI com IEEE, com ênfase na interseção de processo e prática. Mutafelija e Stromberg (*Systematic Process Improvement Using ISO 9001:2000 and CMMI*, Artech House Publishers, 2007) discutem as estruturas ISO 9001:2000 e CMMI SPI e a “sinergia” entre elas. Conradi e seus colegas (*Software Process Improvement: Re-*

sults and Experience from the Field, Springer, 2006) apresentam os resultados de uma série de estudos de caso e experimentos relacionados com SPI. Van Loon (*Process Assessment and Improvement: A Practical Guide for Managers, Quality Professionals and Assessors*, Springer, 2006) discute a SPI no contexto da ISO/IEC 15504. Watts Humphrey (*PSP*, Addison-Wesley, 2005, and *TSP*, Addison-Wesley, 2005) trata de sua estrutura SPI Personal Team Process e Team Software Process em dois livros. Fantina (*Practical Software Process Improvement*, Artech House Publishers, 2004) traz instruções pragmáticas com ênfase em CMMI/CMM.

Uma grande variedade de fontes de informação sobre melhoria de processo de software está disponível na Internet. Uma lista atualizada das referências na Web relevantes a SPI pode ser encontrada no site www.mhhe.com/engcs/compsci/pressman/professional/olc/ser.htm.