

PONTO-CHAVE

Os roteiros (scripts) do TSP definem os elementos e as atividades realizadas no transcorrer do processo.

que mede quantitativamente o processo e o produto. A autópsia representa o estágio para melhorias dos processos.

Esse processo faz uso de uma grande variedade de roteiros (scripts), formulários e padrões que servem para orientar os membros da equipe em seu trabalho. Os roteiros definem atividades de processos específicas (isto é, lançamento do projeto, projeto, implementação, integração e testes do sistema, autópsia) e outras funções de trabalho mais detalhadas (por exemplo, planejamento do desenvolvimento, desenvolvimento de requisitos, gerenciamento das configurações de software, teste de unidade) que fazem parte do processo de equipe.

O TSP reconhece que as melhores equipes de software são autodirigidas.²³ Seus membros estabelecem os objetivos do projeto, adaptam o processo para atender suas necessidades, controlam o cronograma e, através de medições e análise das métricas coletadas, trabalham continuamente para aperfeiçoar a abordagem em relação à engenharia de software.

Assim como o PSP, o TSP é uma rigorosa abordagem da engenharia de software que fornece benefícios distintos e quantificáveis para a produtividade e para a qualidade. A equipe deve se comprometer totalmente com o processo e deve passar por treinamento consciente para assegurar que a abordagem seja apropriadamente aplicada.

2.7 TECNOLOGIA DE PROCESSOS

Um ou mais dos modelos de processo discutidos nas seções anteriores devem ser adaptados para ser empregados por uma equipe de software. Para tanto, desenvolveram-se *ferramentas de tecnologia de processos*, com o objetivo de auxiliar organizações de software a analisar seus processos atuais, organizar tarefas de trabalho, controlar e monitorar o progresso, bem como administrar a qualidade técnica.²⁴

As ferramentas de tecnologia de processos permitem a uma organização de software construir um modelo automatizado da metodologia de processos, conjuntos de tarefas e atividades de apoio (*umbrella activities*), discutidos na Seção 2.1. O modelo, normalmente representado como uma rede, pode, então, ser analisado para determinar o fluxo de trabalho típico e exa-

FERRAMENTAS DO SOFTWARE



Ferramentas de modelagem de processos

Objetivo: quando uma organização trabalha para aprimorar um processo de negócio (ou de software), ela precisa, primeiramente, compreendê-lo. As ferramentas de modelagem de processos (também chamadas ferramentas de tecnologia de processos ou ferramentas de gerenciamento de processos) são usadas para representar elementos-chave de um processo para que possa ser mais bem compreendido. Essas ferramentas podem também oferecer "links" para descrições de processos, ajudando os envolvidos no processo a compreender as ações e tarefas necessárias para realizá-lo. As ferramentas de modelagem de processos fornecem links para outras ferramentas que oferecem suporte para atividades de processos definidas.

Mecânica: as ferramentas nesta categoria permitem a uma equipe de desenvolvimento definir os elementos de um modelo

único de processo (ações, tarefas, artefato, pontos de garantia da qualidade de software), dar orientação detalhada sobre o conteúdo ou descrição de cada elemento de um processo e, então, gerenciar o processo conforme ele for conduzido. Em alguns casos, as ferramentas de tecnologia de processos incorporam tarefas padronizadas de gerenciamento de projeto como estimativa de custos, cronograma, acompanhamento e controle.

Ferramentas Representativas:

Igrafx Process Tools — ferramentas que capacitam uma equipe a mapear, medir e modelar o processo de software (www.micrografx.com)

Adeptia BPM Server — projetada para gerenciar, automatizar e otimizar processos de negócio (www.adeptia.com)

SpeedDev Suite — conjunto de seis ferramentas com forte ênfase no gerenciamento das atividades de comunicação e modelagem (www.speeddev.com)

²³ No Capítulo 3 discutiremos a importância das equipes "auto-organizadas" como um elemento-chave no desenvolvimento de software ágil.

²⁴ As ferramentas aqui citadas não representam um aval, mas sim uma amostragem de ferramentas nesta categoria. Na maioria dos casos, os nomes das ferramentas são marcas registradas de seus respectivos desenvolvedores.

minar estruturas de processos alternativas que possam levar à redução de custos e tempo de desenvolvimento.

Uma vez criado um processo aceitável, outras ferramentas de tecnologia de processo poderão ser usadas para alocar, monitorar e até mesmo controlar todas as atividades, ações e tarefas de engenharia de software definidas como parte do modelo de processo. Cada membro da equipe poderá usar tais ferramentas para desenvolver uma lista de controle das tarefas a ser realizadas, dos artefatos de software a ser gerados e das atividades de garantia da qualidade a ser realizadas. A ferramenta de tecnologia de processos também pode ser usada para coordenar o uso de outras ferramentas de engenharia de software que são apropriadas para uma determinada tarefa.

2.8 PROCESSO DO PRODUTO

Se o processo for fraco, certamente o produto final sofrerá consequências. Porém, uma confiança excessiva e obsessiva no processo é igualmente perigosa. Em um breve artigo, escrito muitos anos atrás, Margaret Davis [Dav95a] tece comentários atemporais sobre a dualidade produto e processo:

Aproximadamente a cada dez anos (acrescente ou elimine cinco), a comunidade de software redefine “o problema”, mudando seu foco de itens do produto para itens de processo. Assim, adotamos linguagens de programação estruturada (produto), seguidas por métodos de análise estruturada (processo), seguidos pelo encapsulamento de dados (produto), seguido pela ênfase atual no Modelo de Maturação da Capacidade de Desenvolvimento de Software (processo), do Software Engineering Institute [seguido por métodos orientados a objeto, seguido pelo desenvolvimento de software ágil].

Enquanto a tendência natural de um pêndulo é a de vir repousar num ponto intermediário entre dois extremos, o foco da comunidade de software muda constantemente, pois nova força é aplicada quando a última oscilação falha. Essas oscilações causam danos para si mesmos e para o ambiente externo, confundindo o profissional típico de software, mudando radicalmente o que significava desempenhar bem seu trabalho. Essas oscilações também não resolvem “o problema”, pois estão fadadas ao insucesso, enquanto produto e processo forem tratados como formando uma dicotomia (divisão de um conceito em dois elementos, em geral, contrários) em vez de uma dualidade (coexistência de dois princípios).

Na comunidade científica, há precedentes da tendência para adotar noções de dualidade quando, nas observações, as contradições não podem ser explicadas completamente nem por uma nem por outra teoria que competem entre si. A natureza dual da luz, parecendo ser simultaneamente partícula e onda, foi aceita desde os anos 1920, quando Louis de Broglie a propôs. Pelas observações feitas dos artefatos de software e de seu desenvolvimento, fica demonstrada a existência de uma dualidade fundamental entre produto e processo. Jamais poderemos destrinchar ou compreender o artefato completo, seu contexto, uso, significado e valor se o enxergarmos apenas ou como um processo ou um produto...

Todas as atividades humanas podem ser um processo, mas todos sentem-se valorizados quando tais atividades se tornam uma representação ou um exemplo, sendo utilizadas ou apreciadas por mais de uma pessoa, repetidamente, ou então utilizadas num contexto não imaginado. Ou seja, extraímos sentimentos de satisfação na reutilização de nossos produtos, seja por nós mesmos, seja por outros.

Assim, enquanto a assimilação rápida das metas de reuso, no desenvolvimento de software, aumenta potencialmente a satisfação dos profissionais de software, ela também aumenta a urgência da aceitação da dualidade produto e processo. Enxergar um artefato reutilizável apenas como um produto ou apenas como um processo, obscurece o contexto e as maneiras de usá-lo, ou obscurece o fato de que cada uso resulta em produto que, por sua vez, será utilizado como entrada, em alguma outra atividade de desenvolvimento de software. Adotar

uma dessas visões em detrimento da outra reduz dramaticamente as oportunidades de reutilização e, portanto, perde-se a oportunidade de aumentar a satisfação no trabalho.

As pessoas obtêm satisfação tanto do processo criativo quanto do produto final. Um artista sente prazer tanto de suas pinceladas quanto do resultado geral de seu quadro. Um escritor sente prazer tanto da procura da metáfora apropriada quanto do livro finalizado. Como profissional de software criativo, você também deve extrair tanta satisfação do processo como do produto final. A dualidade produto e processo é um elemento importante para manter pessoas criativas engajadas à medida que a engenharia de software continua a evoluir.

2.9 RESUMO

Um modelo de processo genérico para engenharia de software consiste num conjunto de atividades metodológicas e de apoio (*umbrella activities*), ações e tarefas a realizar. Cada modelo de processo, dentre os vários existentes, pode ser descrito por um fluxo de processo diferente — descrição de como as atividades metodológicas, ações e tarefas são organizadas sequencial e cronologicamente. Padrões de processo são utilizados para resolver problemas comuns encontrados como parte do processo de software.

Os modelos de processo prescritivos são aplicados há anos, num esforço para organizar e estruturar o desenvolvimento de software. Cada um desses modelos sugere um fluxo de processos ligeiramente diferente, mas todos realizam o mesmo conjunto de atividades metodológicas genéricas: comunicação, planejamento, modelagem, construção e emprego.

Os modelos de processo sequenciais, tais como o de cascata e o modelo V, são os paradigmas da engenharia de software mais antigos. Eles sugerem um fluxo de processos linear que, frequentemente, é inadequado para considerar as características dos sistemas modernos (por exemplo, contínuas alterações, sistemas em evolução, prazos apertados). Entretanto, eles têm, realmente, aplicabilidade em situações em que os requisitos são bem definidos e estáveis.

Modelos de processo incremental são iterativos por natureza e produzem rapidamente versões operacionais do software. Modelos de processos evolucionários reconhecem a natureza iterativa e incremental da maioria dos projetos de engenharia de software e são projetados para adequar mudanças. Esses modelos, como prototipação e o modelo espiral, produzem rapidamente artefatos de software incrementais (ou versões operacionais do software). Podem ser adotados para ser aplicados por todas as atividades de engenharia de software — desde o desenvolvimento de conceitos até a manutenção do sistema a longo prazo.

Modelo de processo concorrente possibilita que uma equipe de software represente elementos iterativos e concorrentes de qualquer modelo de processo. Modelos especializados incluem o modelo baseado em componentes (que enfatiza a montagem e a reutilização de componentes); o modelo de métodos formais (que encoraja uma abordagem matemática para o desenvolvimento e a verificação de software); e o modelo orientado a aspectos (que considera interesses cruzados que se estendem por toda a arquitetura do sistema). O Processo Unificado é um processo de software “dirigido a casos práticos, centrado na arquitetura, iterativo e incremental”, desenvolvido como uma metodologia para os métodos e ferramentas da UML.

Modelos pessoal e de equipe enfatizam a medição, o planejamento e autodirecionamento como ingredientes-chave para um processo de software bem-sucedido.

PROBLEMAS E PONTOS A PONDERAR

2.1. Na introdução deste capítulo, Baetjer observa: “O processo oferece interação entre usuários e projetistas, entre usuários e ferramentas em evolução e entre projetistas e ferramentas [de tecnologia] em evolução”. Liste cinco perguntas que (a) os projetistas deveriam fazer aos usuários, (b) os usuários deveriam fazer aos projetistas, (c) os usuários deveriam fazer a si mesmos sobre o produto de software a ser desenvolvido, (d) os projetistas deveriam fazer