

Conceitos fundamentais da engenharia de software

Este capítulo tem por objetivo abordar os principais conceitos sobre a engenharia de software, a engenharia de software para web, processo de desenvolvimento de software, bem como modelos de processo. Os conceitos abordados têm como base informações obtidas dos principais autores da área.

Engenharia de software

Segundo definições da Software Engineering Institute (1990), engenharia é “a aplicação sistemática do conhecimento científico na criação e construção de soluções para problemas práticos a serviço do gênero humano”, e engenharia de software é “uma forma de engenharia em que se aplicam os princípios da ciência da computação e da matemática para alcançar soluções com melhor custo-benefício para o problema do software”.

O termo “engenharia” está associado à construção do software. De acordo com Pressman (2002), uma primeira definição de engenharia de software foi proposta por Fritz Bauer na primeira grande conferência dedicada ao assunto: “O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais.”

O autor ainda menciona a engenharia de software como sendo um rebento da engenharia de sistemas e de hardware,

abrangendo um conjunto de três elementos fundamentais – métodos, ferramentas e procedimentos –, que possibilitam ao gerente o controle do processo de desenvolvimento do software e oferecem ao profissional uma base para a construção de softwares de alta qualidade.

Os métodos de engenharia de software proporcionam os detalhes de “como fazer” para construir o software e envolvem um amplo conjunto de tarefas, que incluem: planejamento e estimativa de projeto, análise e requisitos de software e de sistemas, projeto da estrutura de dados, arquitetura de programa e algoritmo de processamento, codificação, teste e manutenção.

As ferramentas são desenvolvidas para proporcionar apoio aos métodos. Atualmente, existem ferramentas para sustentar cada um dos métodos. Quando elas são integradas de forma que a informação criada por uma ferramenta possa ser usada por outra, é estabelecido um sistema de suporte ao desenvolvimento de software, chamado “engenharia de software auxiliada por computador” (Computer-Aided Software Engineering - CASE).

Os procedimentos da engenharia de software definem a seqüência em que os métodos serão aplicados, os produtos que precisam ser entregues (documentos, relatórios, formulários etc.), os controles que ajudam a assegurar a qualidade e a coordenar as mudanças e os marcos de referência que possibilitam aos gerentes de software avaliar o progresso.

Em suma, engenharia de software, segundo o Institute of Electrical Eletronical Engineers (1993), é um enfoque sistemático para o desenvolvimento, operação, manutenção e descontinuação do software. Segundo a definição de Boehm (1975), é a aplicação prática do conhecimento científico no projeto, construção de programas e da documentação requerida para desenvolver, operar e manter esses programas.

Os fundamentos científicos para a engenharia de software envolvem o uso de modelos abstratos e precisos que permitem ao engenheiro especificar, projetar, implementar e manter sistemas de software, avaliando e garantindo sua

qualidade. Além disso, a engenharia de software deve oferecer mecanismos para se planejar e gerenciar o processo de desenvolvimento (Tonsig, 2003).

Por muitos anos, tem-se vivido uma verdadeira confusão no desenvolvimento de software. É mais freqüente do que se imagina a presença de empresas desenvolvendo e entregando aos clientes softwares imensos, com alto grau de complexidade, apresentando defeitos, mal-estruturados, consumidores de recursos de máquina e, principalmente, que não atendem completamente às necessidades dos reais usuários (Tonsig, 2003).

O objetivo da engenharia de software é, justamente, solucionar esses problemas, reduzindo custos e prazos, gerenciando o processo de desenvolvimento, permitindo o trabalho em grupo e aplicando conceitos de qualidade. A engenharia de software vem sendo cada vez mais exigida no desenvolvimento de software, pois necessita-se planejar, acompanhar, executar e controlar para que ela seja realmente aplicada.

Sistemas web e engenharia de software na web

O crescimento da internet nos últimos anos foi geométrico, a ponto de estar presente no dia-a-dia das pessoas, tanto que, utilizando a internet, podem-se fazer desde compras até um curso superior. Todavia, as técnicas para o desenvolvimento de sistemas web não tiveram a mesma evolução. Os projetos, apesar de muito criativos, são desenvolvidos com pouca disciplina, principalmente, para que possam ser entregues ao cliente no menor tempo possível.

Os principais tipos de sistemas web são intranet, extranet, e-commerce, e-training, wap, sistemas de informação na web, data warehouse, data center, entre outros existentes no vasto "mundo da internet". A modalidade de sistemas web adotada no modelo apresentado neste livro (Modelo ESTRELA) refere-se a e-commerce (comércio eletrônico), uma área carente de processos adequados para o desenvolvimento e gerenciamento.

Os sistemas web são diferentes dos sistemas tradicionais porque envolvem uma mescla de publicação impressa e desenvolvimento de software, comercialização e computação, comunicações internas e relações externas, arte e tecnologia. Conforme Pressman (2002, p. 753), as seguintes características são encontradas na grande maioria dos sistemas web:

- *concentrados em redes*: um sistema baseado na web está numa rede por natureza e necessita atender às necessidades de uma comunidade diversificada de clientes. Este sistema pode estar disponível na internet (permitindo comunicação à rede mundial de computadores), na intranet (implementando as comunicações dentro da organização) ou, ainda, na extranet (comunicação entre redes);
- *impulsionadas pelo conteúdo*: em muitos casos, a função primordial de um sistema baseado na web é usar a hipermídia para apresentar conteúdo de texto, gráficos, áudio e vídeo ao usuário final;
- *evolução contínua*: ao contrário dos softwares tradicionais, que evoluem ao longo de uma série de versões planejadas e cronologicamente espaçadas, aplicações baseadas na web evoluem continuamente.

Em virtude dessa evolução contínua, faz-se necessário estabelecer mecanismos para controle de configuração, garantia de qualidade e suporte continuado. Segundo Pressman (2002), muitos dos conceitos da engenharia de software tradicional podem ser aplicados na web. Contudo, na ausência de um processo disciplinado, podem-se enfrentar sérios problemas no desenvolvimento, entrega e manutenção. A fim de evitar tais problemas, há necessidade de aplicar os conceitos da engenharia da web, novos métodos e ferramentas.

“A engenharia da Web (Web Engineering) ou WebE diz respeito ao estabelecimento e uso de princípios científicos sólidos, de engenharia e de gestão, e abordagens disciplinadas e sistemáticas para o bem-sucedido desenvolvimento, disposição e manutenção de sistemas e aplicações de alta qualidade baseados na Web” (Murugesan apud Pressman, 2002, p. 752).

A engenharia de software para web surgiu com a finalidade de disciplinar o desenvolvimento de aplicações web com alta qualidade e a um custo efetivo. As semelhanças que a engenharia de software para web apresenta em relação à engenharia de software tradicional são os conceitos e princípios fundamentais, incorporando-se novas abordagens, metodologias, técnicas e normas para atender aos requisitos exclusivos dos sistemas para web. Engenharia de software para web é o processo utilizado para criar sistemas para web, baseado em planejamento, gerenciamento de custos e pessoal, bem como em qualidade do produto e do processo.

Como os sistemas web se tornam cada vez mais integrados nas estratégias de negócio, para grandes e pequenas empresas (por exemplo, comércio eletrônico), a importância de construir sistemas confiáveis, usáveis e adaptáveis aumenta. A engenharia da web aplica uma abordagem genérica combinada com estratégias, táticas e métodos especializados; além de contar com usuários diversificados, que priorizam a apresentação, aparência, navegação e outras qualidades estéticas (Nakasato, 2004).

O desenvolvimento de sistemas para web está frequentemente relacionado à criação de uma infra-estrutura. De acordo com Pressman (2002), pode-se comparar a evolução desses sistemas com jardinagem, visto que se devem "plantar" as informações que irão crescer e florescer nesse jardim. Com o tempo, o jardim (sistema web/site) irá evoluir, mudar e crescer, e um planejamento inicial adequado permitirá que esse crescimento ocorra de forma controlada e consistente.

Uma equipe de WebE é formada por vários membros que desempenham diferentes papéis. As atividades típicas desenvolvidas são: coletar conteúdo, organizar o conteúdo coletado, deduzir necessidades, projetar a arquitetura, projetar a interface, projetar a estrutura navegacional, implementar, testar e dar suporte (correções, adaptações, atualizações e aperfeiçoamento) ao sistema. Dentre os conhecimentos necessários à equipe estão uma sólida compreensão de tecnologias

de componentes, arquiteturas cliente/servidor, HTML/XML¹ e tecnologias de bases de dados, bem como conhecimento de conceitos multimídia, plataformas de hardware/software, segurança de rede e aspectos de apoio a sistemas da web. A fim de evitar confusão, frustração e falhas, é necessário planejar, considerar riscos, estabelecer e rastrear cronogramas e definir controles. Essas atividades estão relacionadas ao gerente de projetos/engenheiro de software.

Processos de desenvolvimento de software

Segundo Fiorini et al. (1999), um dos grandes desafios da área da engenharia de software tem sido desenvolver softwares de qualidade, com elevada produtividade, dentro dos prazos estabelecidos, sem necessitar de mais recursos do que os previstos. Aponta-se que uma das principais causas dos problemas é a falta de um processo de desenvolvimento de software claramente definido e efetivo.

Um processo é um conjunto de passos parcialmente ordenados, constituídos por atividades, métodos, práticas e transformações, utilizado para atingir uma meta, a qual geralmente está associada a um ou mais resultados concretos finais, que são os produtos da execução do processo. Um processo é definido quando tem documentação detalhadora do que é feito (produto), quando (passos), por quem (agentes), o que usa (insumo) e o que produz (resultados) (Paula Filho, 2003, p. 13).

Processo pode ser definido como um conjunto de procedimentos e métodos arranjados de maneira ordenada que visam à obtenção de um resultado (Sommerville, 2003). De acordo com Pfleeger (2004, p. 36), pode-se considerar um conjunto de tarefas ordenadas como sendo um processo: uma série de etapas que envolvem atividades, restrições e recursos para alcançar uma saída desejada. Quando o processo envolve a

elaboração de um produto, muitas vezes faz-se referência a ele como “ciclo de vida”. Assim, o processo de desenvolvimento de software pode ser chamado de ciclo de vida do software, porque descreve a “vida” do produto de software desde a concepção até a implementação, entrega, utilização e manutenção.

Os processos são importantes porque imprimem consistência e estrutura a um conjunto de atividades. Um processo ajuda a manter as características de qualidade do produto final, embora produzido por pessoas diferentes. A estrutura do processo orienta as ações, permitindo examinar, entender, controlar e aprimorar as atividades que o compõem. Os processos também nos permitem capturar experiências e passá-las adiante por meio de processos e procedimentos documentados.

No contexto da engenharia de software, um processo é um framework² para as tarefas necessárias à construção de software com alta qualidade, definindo a abordagem que será seguida enquanto o software estiver em desenvolvimento (Pressman, 2002). Os processos são utilizados na engenharia de software com a finalidade de guiar a criação do software visando garantir a qualidade no resultado final.

Conforme Sommerville (2003), os processos de software existentes são os mais diversos, mas todos apresentam atividades fundamentais comuns, tais como:

- *especificação de software*: são definidas as funcionalidades e as restrições do software;
- *projeto e implementação do software*: o software deve ser produzido atingindo a sua especificação;
- *validação do software*: o software deve ser validado antes da entrega ao cliente, a fim de garantir que atenda as suas especificações;
- *evolução do software*: para atender às necessidades mutáveis do cliente, o software deve evoluir.

Sobre as atividades dos processos, o mesmo autor aborda que diferentes processos de software organizam essas atividades de maneiras diversas e são escritos em diferen-

tes níveis de detalhes. Os prazos das atividades variam, do mesmo modo que os resultados de cada atividade. Diferentes organizações podem utilizar diferentes processos para produzir o mesmo tipo de produto. No entanto, alguns processos são mais adequados do que outros para alguns tipos de aplicações. Se um processo inadequado for utilizado, provavelmente reduzirá a qualidade ou a utilidade do produto de software a ser desenvolvido (2003, p. 7).

Na visão de Pressman (2002), todo processo pode ser categorizado em três fases genéricas independentemente da área de aplicação, do tamanho do projeto ou de sua complexidade. São elas:

- *definição*: nesta fase deve-se concentrar em “o quê” o software deve fazer. O engenheiro de software tenta identificar o que deve ser considerado, que funções e desempenho são esperados, que restrições existem e que critérios de validação são necessários para que o sistema seja bem-sucedido. Os requisitos-chave do sistema e do software devem ser definidos. Os métodos utilizados para conduzir essa tarefa dependem do paradigma de engenharia de software que será adotado para o desenvolvimento do sistema;
- *desenvolvimento*: nesta fase focaliza-se o “como”. O engenheiro de software tenta definir como os requisitos identificados serão implementados, preocupando-se com o modo como os dados serão armazenados, como será a arquitetura do sistema, a interface com o usuário, que funções o software terá, como o projeto será traduzido para uma linguagem de programação e como serão conduzidos os testes. Os métodos aplicados nesta fase variam, mas três tarefas específicas devem abranger: projeto, codificação e testes de software;
- *manutenção*: esta fase focaliza as mudanças associadas com a correção de erros; adaptações necessárias à medida que o ambiente do software evolui; modificações exigidas para melhorar o software, ou seja, provocadas pelas mudanças nos requisitos do clien-

te/usuário; inserção de novas funcionalidades no software, não previstas no projeto inicial. A fase de manutenção aplica novamente os passos das fases de definição e desenvolvimento, mas o faz no contexto do software existente.

Apesar da inexistência de um processo de software “ideal”, há muitas oportunidades de trabalho para o melhoramento de processos em muitas organizações. Os processos podem conter técnicas desatualizadas ou não explorar as vantagens das melhores práticas na engenharia de software. Sommerville (2003) relata que muitas organizações ainda confiam em processos criados para atender a um caso específico e não tiram vantagens de métodos de engenharia de software em seu desenvolvimento.

Modelo de processo de software

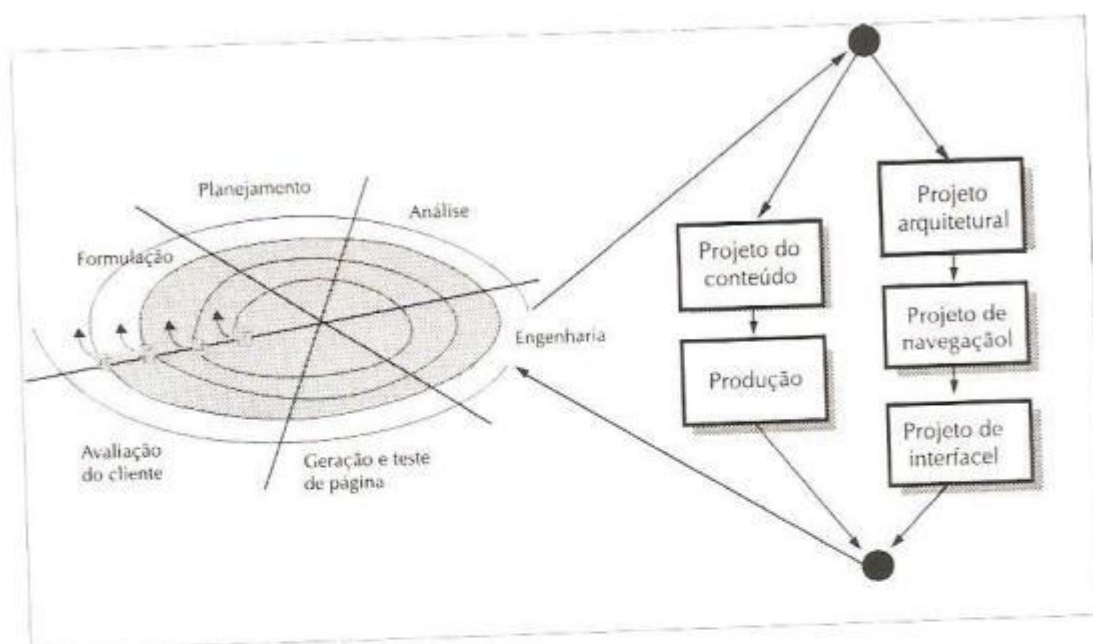
Um modelo de processo de software é uma representação abstrata de um processo de software que está sendo descrito. Essas abstrações podem ser utilizadas para explicar diferentes abordagens do desenvolvimento de software.

Existem diferentes modelos gerais, ou paradigmas de desenvolvimento de software, tais como modelo Cascata, desenvolvimento evolucionário, desenvolvimento orientado ao reúso, desenvolvimento formal de sistemas (como, por exemplo, o Cleanroom), desenvolvimento incremental, entre outros.

Para resolver problemas, a equipe de engenheiros de software deve definir uma estratégia de desenvolvimento que envolva as camadas de processo, métodos, ferramentas e fases genéricas. Pressman (2002) refere-se a essa estratégia como um modelo de processo de software ou paradigma de engenharia de software.

Os sistemas e aplicações baseados na web, diferentemente dos softwares tradicionais, são dinâmicos e com uma população de usuários diversificada. À medida que os sistemas para web evoluem, é necessário aplicar um gerenciamento sólido, no que os princípios da engenharia passam a ter

fundamental importância. A fim de atender a tais requisitos de forma eficiente, Pressman (2002) propõe um modelo de processo genérico para web (Fig. 1).



Fonte: Pressman, 2002.

Figura 1 - Modelo de processo da WebE

O processo inicia-se com a formulação dos problemas a serem resolvidos, atividade que indica as metas do sistema para web e determina o escopo para o primeiro incremento. A fase de planejamento visa estimar os custos do projeto, avaliar os riscos associados ao desenvolvimento e elaborar cronogramas para os incrementos. Na fase de análise são definidos os requisitos técnicos e identificados os itens de conteúdos a serem incorporados, além de serem estabelecidos os requisitos para o projeto gráfico (estética). A atividade de engenharia agrega duas tarefas paralelas (ilustradas do lado direito da Fig. 1): projeto de conteúdo e produção. Essas tarefas são realizadas pelos membros “não técnicos” da equipe de WebE e têm por objetivo projetar, produzir e/ou obter todo o conteúdo (texto, gráfico, áudio e vídeo) que será integrado ao sistema

web. Ao mesmo tempo, um conjunto de tarefas de projeto técnico é conduzida pelos desenvolvedores.

A geração e teste de páginas é uma atividade de construção que usa ferramentas automáticas para a criação de sistemas web. O conteúdo definido na atividade de engenharia é conciliado com projetos de arquitetura, navegação e interface para produzir páginas da web executáveis em HTML, XML e outras linguagens orientadas a processo (por exemplo, Java). Nesta atividade ainda é realizada a integração com componentes de middleware (por exemplo, CORBA, DCON ou JavaBeans). A atividade de teste tenta descobrir erros em applets, scripts e formulários através da navegação do sistema para web e ajuda a garantir que o sistema funcione corretamente em diferentes ambientes (navegadores).

Cada incremento produzido como parte do processo é revisado durante a fase de avaliação do cliente. As modificações pedidas, que estendem o escopo do sistema, são integradas no próximo incremento do processo.

Ginige e Murugesan (2001) destacam alguns passos para a construção de um sistema para web de sucesso:

- entender o funcionamento geral e operacional do ambiente dos sistemas;
- identificar e especificar requisitos técnicos e não técnicos;
- desenvolver uma arquitetura apropriada;
- tratar satisfatoriamente as limitações não técnicas;
- identificar subprojetos ou subprocessos para implementar a arquitetura;
- desenvolver e implementar os subprojetos;
- incorporar mecanismos apropriados e efetivos para gerenciar a evolução e a manutenção.

Pesquisas feitas por McDonald e Welland (2001) apontam que a maioria dos desenvolvedores para web não utiliza um processo de desenvolvimento bem definido e documentado; alguns usam modelos padronizados pela indústria de