

Web Engineering: Questões Práticas

Web Engineering: practical issues

Rogério Xavier de Azambuja¹, Christian Puhlmann Brackmann¹, Thaís Andrea Baldissera²

¹Instituto Federal de Educação, Ciência e Tecnologia Farroupilha (IF-Farroupilha)
Rua Erechim, 860 – 98.280-000 – Panambi – RS – Brazil

²Instituto Federal de Educação, Ciência e Tecnologia Farroupilha (IF-Farroupilha)
Caixa Postal 38 – 98.130-000 – São João do Barro Preto – Júlio de Castilhos–RS–Brazil
{xavier, brackmann}@iffpb.edu.br, tbaldissera@jc.iffarroupilha.edu.br}

Abstract: *The innovative technological paradigms have drawn attention to new and emerging needs for improvement in software construction, especially those concerning software development for Web applications (WebApps). This article attempts to characterize the practical aspects of Web Engineering or Oriented Software Engineering for Web Applications, by presenting concepts, fundamentals, and a brief comparison with traditional software engineering, without the intention of exhausting the issue.*

Key words: *Software development. Web applications. Software engineering.*

Resumo: Os inovadores paradigmas tecnológicos têm apontado para novas e emergentes necessidades de aprimoramento na construção de softwares, especialmente no que se refere ao desenvolvimento de softwares para aplicações Web (WebApps). Este artigo procura caracterizar na prática a Web engineering ou engenharia de software voltada para aplicações Web, apresentando conceitos, fundamentos e uma breve comparação com a engenharia de software tradicional, sem a intenção de esgotar o assunto.

Palavras Chaves: de Softwares, Aplicações Web, Engenharia de Software.





Introdução

Nos últimos anos, a Internet tem sido o meio de comunicação com maior crescimento no mundo, sendo um canal utilizado diuturnamente por milhares de pessoas para a comunicação, o entretenimento, a divulgação, compra e venda de produtos e serviços.

Inicialmente, entre os anos 1990 e 1995, a *World Wide Web* (teia de alcance mundial ou simplesmente *Web*) possuía apenas conteúdos estáticos e era utilizada somente para compartilhar informações, sendo aprimorada com o passar do tempo. Também não apresentava a preocupação de serem utilizadas técnicas e métodos padronizados no desenvolvimento de aplicações específicas para o ambiente *Web*.

Com a constante evolução das ferramentas de desenvolvimento e a possibilidade de acessos a bancos de dados, foi permitido fornecer maior capacidade computacional às informações transitadas na grande rede mundial de computadores – conteúdos dinâmicos. Segundo Pressman (2006, p. 378), “sistemas e aplicações *Web* (*WebApps*) produzem uma complexa matriz de conteúdo e funcionalidade para uma ampla população de usuários finais, espalhados pelo globo terrestre”. Nas *WebApps* são evidenciadas características particulares, que as tornam diferentes de outros *softwares*, pois existe maior preocupação com questões visuais, com a usabilidade, além de características específicas de segurança, que justificam a utilização de técnicas de engenharia.

A *Web engineering* ou engenharia de software voltada para aplicações *Web*, não é igual à engenharia de software, mas utiliza muitos dos seus conceitos e princípios fundamentais, com ênfase nas mesmas atividades técnicas e de gestão (CECHELERO, 2004, p.49). Trata-se de uma engenharia focada na construção de aplicações hipermídia com qualidade, baseada em critérios como: necessidades dos usuários, conteúdo de alta qualidade, atualizações constantes (manutenção), tempo de *download* mínimo, facilidade de uso (usabilidade) e cultura corporativa centralizada na rede (PRESSMAN, 2006).

Nesse contexto, o presente artigo tem por objetivo apresentar uma visão geral da engenharia para sistemas e aplicações baseadas na *Web*, apresentando alguns conceitos fundamentais, procurando caracterizar de forma prática a engenharia da *Web* e comparar com a engenharia de software tradicional, servindo como referência e sem a intenção de esgotar o assunto.



Fundamentos Teórico-Conceituais

Engenharia de Software

Para Sommerville (2007), a engenharia de software (ES) é a disciplina que se preocupa com o projeto, a implementação, a instalação e a operação de sistemas que incluem *hardware*, *software* e pessoas, desde os estágios iniciais de especificação do sistema, até a manutenção após entrar em operação.

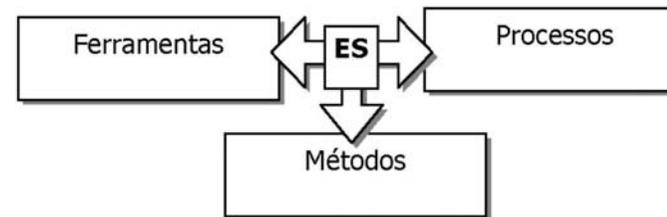


Figura 1: Engenharia de Software = Métodos + Processos + Ferramentas

Conforme ilustrado na Figura 1, a ES é uma área do conhecimento da informática que utiliza um conjunto de métodos, processos e ferramentas para analisar, projetar e gerenciar o desenvolvimento e a manutenção de *softwares*, aplicando tecnologias e práticas de ciência da computação, gerência de projetos e outras disciplinas, objetivando organização, produtividade e qualidade.

Atualmente, essas tecnologias e práticas englobam linguagens de programação, bases de dados, ambientes de desenvolvimento, plataformas de sistemas operacionais, bibliotecas, padrões, processos e aspectos relacionados à qualidade de *software*.

Os fundamentos científicos para a ES envolvem o uso de modelos abstratos e precisos que permitam ao desenvolvedor especificar, projetar, implementar e manter sistemas de *software*, avaliando e garantindo a sua qualidade. Além disso, a ES deve oferecer mecanismos para planejar e gerenciar o processo de desenvolvimento de um sistema de informação.



World Wide Web

A *World Wide Web* (que significa: teia de alcance mundial, WWW ou simplesmente *Web*) é um conjunto de documentos em hipermídia, padronizados pela W3C (www.w3c.org), que são armazenados, interligados e executados através da arquitetura Cliente-Servidor da Internet (Figura 2), utilizando o protocolo TCP/IP (Protocolo de Controle de Transferência/Protocolo da Internet). Esses documentos (conteúdo) processados e transmitidos podem estar em diferentes formatos, tais como, hipertextos, gráficos, áudios e vídeos que serão percorridos (“navegação”) pelo cliente.



Figura 2: Cenário de uma aplicação Web (Cliente-Servidor)

Aplicações Web (WebApps) na Prática

De acordo com Pressman (2006, p.379), sistemas baseados na Web “envolvem uma mistura de publicação impressa e desenvolvimento de software,... de arte e tecnologia”. *WebApps* comumente apresentam as seguintes características:





Concentração em Redes: por residir em uma rede, serve às necessidades de uma comunidade diversificada de clientes (Internet, intranet, extranet ou em nuvem).

Concorrência: um grande número de usuários pode ter acesso ao mesmo tempo.

Carga Imprevisível: o número de usuários pode variar periodicamente (diariamente, semanalmente, mensalmente,...).

Desempenho: o cliente necessita esperar por processamentos realizados no servidor.

Disponibilidade: apesar da oferta contínua de acessos, pode necessitar ficar fora do ar para manutenção.

Voltada a dados: comumente utilizadas para apresentar dados oriundos de bancos de dados e processar as interações correlacionadas.

Sensível ao conteúdo: a qualidade do conteúdo é fator determinante de qualidade do sistema.

Evolução continuada: diferentemente do software de aplicação convencional, que evolui ao longo de uma série de versões planejadas e cronologicamente espaçadas, as WebApps evoluem continuamente.

Imediatismo: há a necessidade de colocação rápida no mercado, em questão de horas, dias ou semanas.

Segurança: dificuldade de limitar os acessos, pois os usuários exigem a implementação de módulos seguros.

Estética: diretamente relacionada ao sucesso da aplicação e ao projeto técnico.

Engenharia da Web (Web Engineering) na Prática

A engenharia da *Web* (*Web Engineering*) é caracterizada como a metodologia utilizada para a criação de aplicações *Web* de alta qualidade. De acordo com Pressman (2006, p. 378), “*não é um clone perfeito da engenharia de software, mas toma emprestado muitos dos conceitos e princípios fundamentais, com ênfase em atividades técnicas e de gestão similares*”.

Desde meados da década de 90, são destacadas na literatura, por vários autores, evidências significativas que projetos para a *Web* tenham características específicas, diferentes de projetos de sistemas de *software* convencionais.

Uma dessas diferenças, defendida por Dart (2000), está relacionada à identificação de requisi-



Secretaria de Educação
Profissional e Tecnológica



Ministério
da Educação





tos (engenharia de requisitos). Projetos para a *Web* podem ser classificados como exemplares de uma classe emergente de aplicativos, para os quais o cliente tem dificuldade em destacar prioridades, de acordo com suas necessidades específicas. Em vez disso, a compreensão dos clientes evolui quando o sistema é utilizado.

É importante realçar que a engenharia da *Web* não objetiva apenas a criação de um conjunto de páginas (*website*) a serem geradas e exibidas pelo *programa* navegador (*browser*). Refere-se ao conjunto de métodos, técnicas e ferramentas para auxiliar os desenvolvedores a produzirem *softwares* com maior qualidade, sejam simples ou complexos. Também são considerados parte do *website* outros produtos gerados durante o processo de desenvolvimento, tais como, os requisitos, a análise do projeto, o código-fonte e os dados de testes.

A qualidade do produto, referida pela engenharia da *Web*, está diretamente relacionada com o conteúdo apresentado, pois o sistema precisa oferecer respostas rápidas e úteis aos usuários. Os usuários visitam o website pelo conteúdo, que é foco da atenção na *Web*. O *design* existe para permitir que as pessoas acessem o conteúdo, sendo destacada a facilidade de leitura (*layout*), a linguagem utilizada e a padronização das páginas com informações textuais e gráficas.

As Camadas

O desenvolvimento de aplicações baseadas na *Web* (*WebApps*) incorpora modelos de processo especializados, métodos de ES adaptados às características do desenvolvimento *Web* e um conjunto importante de tecnologias de autorização. Processos, métodos e tecnologias (ferramentas) fornecem uma abordagem em camadas para a engenharia da *Web*, assim definidas em Pressman (2006, p.381-382).

Processos

Os modelos de processos para a engenharia da *Web*, em suma, adotam a filosofia do desenvolvimento ágil (BECK et al., 2001), que enfatiza uma abordagem de desenvolvimento simples e com ciclos rápidos de desenvolvimento, devido ao tempo necessariamente reduzido de colocação no mercado, tornando-se esta uma das missões mais importantes dos desenvolvedores.



No entanto, as atividades de engenharia, necessárias para o desenvolvimento de *WebApps*, devem ser definidas em um processo que:

- acolhe modificações;
- encoraja a criatividade e independência da equipe de desenvolvimento e forte interação com os interessados nas aplicações *Web*;
- constrói sistemas usando pequenas equipes de desenvolvimento;
- enfatiza o desenvolvimento evolutivo ou incremental usando ciclos curtos de desenvolvimento.

Os modelos de processos mais utilizados em *WebApps* são: modelo cascata, modelo evolutivo baseado em prototipação e modelo espiral. Esses modelos, detalhados a seguir, são extensões e adaptações dos modelos utilizados na engenharia de software tradicional (CECHELERO, 2004, p.51).

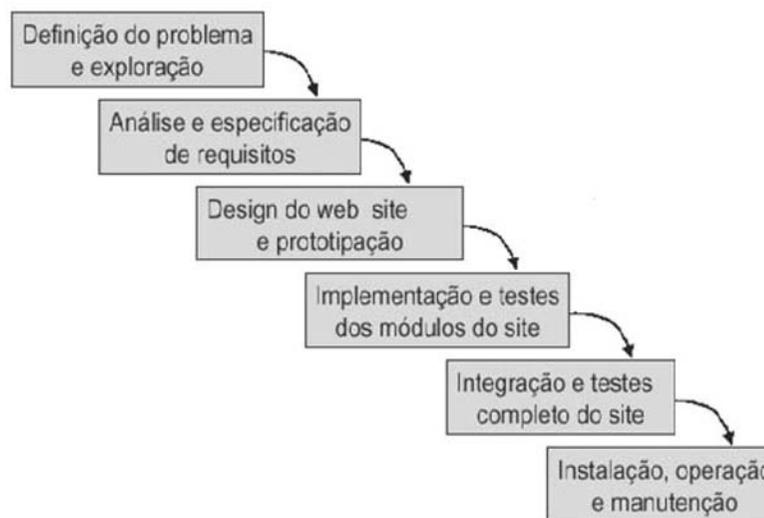


Figura 3: Modelo Cascata



No modelo cascata (Figura 3), as atividades do processo de desenvolvimento são estruturadas em uma cascata de etapas, na qual a saída de uma é a entrada para a próxima etapa. O modelo introduz a separação das atividades de definição, análise e *design* da atividade de implementação, que é o centro das atenções no desenvolvimento *Web*. Possíveis desvantagens poderiam ser a partição inflexível do projeto em estágios distintos, dificuldade para lidar com as mudanças nos requisitos do sistema e a dificuldade de gerenciar riscos.

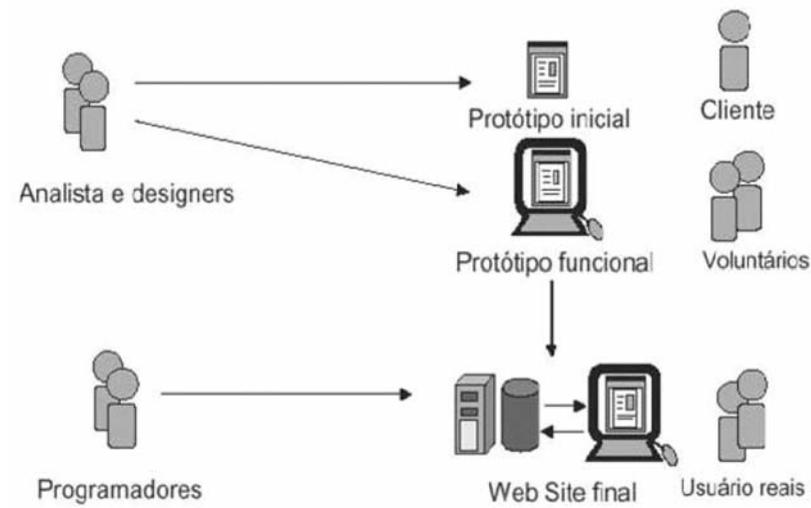


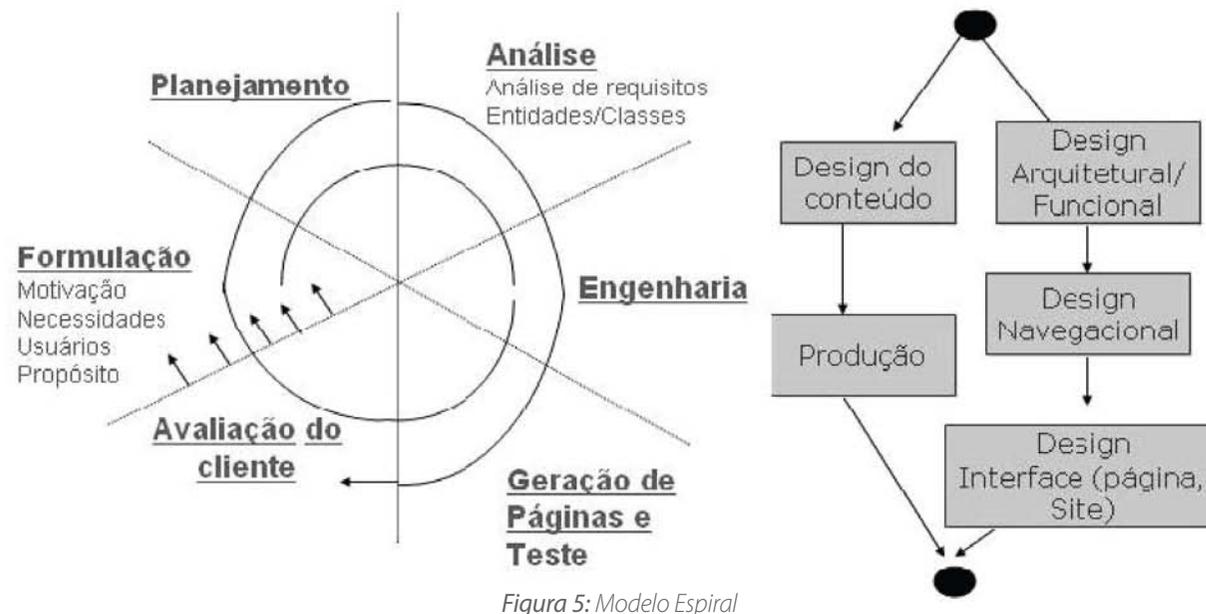
Figura 4: Modelo Cascata

A Figura 4 ilustra o modelo evolutivo baseado em prototipação, que descreve um processo no qual o *software* deve ser desenvolvido de forma a evoluir a partir de protótipos iniciais.

A possível vantagem desse modelo está em permitir a verificação antecipada do produto final por clientes e usuários, facilitando a correção dos problemas detectados e capacidade para ge-



reduzir riscos. Por outro lado, como os requisitos estão sempre sendo revistos, a cada ciclo de desenvolvimento, torna-se mais difícil estimar custos, prazos e planejar as atividades de desenvolvimento.



A Figura 5 ilustra o modelo espiral, inicialmente proposto por Boehm (1986), no qual o processo de desenvolvimento é representado de forma incremental e iterativo, visando atender os curtos prazos exigidos no desenvolvimento de *WebApps*. O fato de ser incremental permite ao processo lidar melhor com um conjunto incerto de requisitos ou sujeito a alterações.

Cada incremento produzido como parte do processo é revisto durante a avaliação do cliente. Nesse ponto, pode haver solicitação de mudanças, sendo integradas ao sistema no próximo ciclo do processo incremental (PRESSMAN, 2006).





Métodos

O panorama de métodos da engenharia da *Web* engloba um conjunto de tarefas técnicas que habilitam um desenvolvedor *Web* entender, caracterizar e então construir uma aplicação de alta qualidade. Os métodos da engenharia da *Web* podem ser:

Métodos de Comunicação: definem a abordagem usada para facilitar a comunicação entre os desenvolvedores e todos os outros interessados em aplicações *Web* (por exemplo: usuários finais, gerentes de negócio, especialistas no domínio do problema, projetistas de conteúdo, líderes de equipe, gerentes de projeto). As técnicas de comunicação são particularmente importantes durante a coleta de requisitos e sempre que um incremento da aplicação *Web* precisar ser avaliado.

Métodos de análise de requisitos: fornecem uma base para o entendimento do conteúdo a ser entregue por uma aplicação *Web*, a funcionalidade a ser fornecida para o usuário final e os modos de interação que cada classe de usuários irá requerer à medida que a navegação pela aplicação *Web* ocorre.

Métodos de Projeto: abrangem uma série de técnicas de projeto que cuidam de conteúdo, arquitetura da aplicação e da informação, projeto da interface e estrutura de navegação da aplicação *Web*.

Métodos de teste: incorporam revisões técnicas formais do conteúdo e modelo de projeto de uma ampla variedade de técnicas de teste, que tratam de tópicos no nível de componente e arquitetural, testes de navegação, testes de usabilidade, testes de segurança e testes de configuração.

Ferramentas e Tecnologias

Uma vasta gama de ferramentas e tecnologias tem evoluído desde a década de 90, à medida que as aplicações *Web* têm se tornado mais sofisticadas e difundidas. Essas tecnologias englobam uma grande variedade de descrição de conteúdo e linguagens de marcas (por exemplo: HTML, VRML, DHTML, XHTML, XML,...), linguagens de programação (por exemplo: Java, PHP, ASP, JSP,...), linguagens de modelagem (por exemplo: UML), recursos de desenvolvimento baseado em componentes (por exemplo, CORBA, COM, ActiveX, .NET,...), navegadores, ferramentas multimídia, ferramentas de autoria de *websites*, ferramentas de conectividade a banco de dados, os próprios banco de dados, ferramentas de segurança, servidores, utilitários de servidor e ferramentas de gestão e análise de *websites*.



Algumas Diferenças entre Engenharia de Software e Engenharia da Web

Ao mesmo tempo em que adota muitos princípios da engenharia de software tradicional, a engenharia da Web incorpora novas abordagens, metodologias, ferramentas, técnicas e normas para atender os requisitos exclusivos das WebApps e segundo Ginige & Murugesan (2001), apresenta também vários desafios adicionais.

O quadro 1 mostra um breve comparativo entre a engenharia de software tradicional e a engenharia da Web, conforme alguns critérios encontrados na literatura da área (LOWE et al., 2001) (LOWE et al., 2005) e também definidos e classificados pelos autores desse texto, porém não fechados. São eles: propósitos ou o objeto da proposta; público-alvo, que são os usuários finais; relação entre as características de apresentação (visuais) e funcionalidades aferidas pelas aplicações; o conhecimento tácito e explícito do desenvolvedor sobre a tecnologia disponível.

Quadro 1: Breve comparativo entre Engenharia de Software (ES) e Engenharia da Web (EW)

	ES	EW
Propósitos	Oferecem serviços e soluções	Oferecem conteúdos que são informações ou serviços
Público-alvo	Público bem definido	Público diversificado
Apresentação x Funcionalidade	Ênfase na funcionalidade e aplicabilidade	Ênfase na apresentação, aparência, navegação e outras qualidades estéticas
Tradição x Experiência	Existe mais experiência no software convencional, possibilitando planejamento e gerência mais realística	Um novo meio em constante exploração
Maturidade da Tecnologia	<ul style="list-style-type: none"> • Tecnologia mais estável • Excelentes ferramentas disponíveis (análise, projeto, implementação e testes) 	<ul style="list-style-type: none"> • Tecnologia em constante evolução • Ferramentas disponíveis para implementação, mas existem poucas para análise, projeto e outras atividades da EW.
Evolução Tecnológica	<ul style="list-style-type: none"> • APIs (Interface de Programação de Aplicações) para interfaces gráficas • Banco de dados multidimensionais 	<ul style="list-style-type: none"> • Navegadores • Servidores • HTML, XML, ASP, PHP... • Frameworks



Considerações Finais

Esse artigo caracterizou a engenharia da *Web*, mostrando que aplicações *Web* são diferentes de outras categorias de *softwares*. São intensamente voltadas para redes de computadores, guiadas por conteúdo e evoluem continuamente.

O imediatismo que preside o desenvolvimento das aplicações *Web*, a necessidade prevalente de segurança na sua operação e a demanda por estética, bem como a distribuição de conteúdo funcional são fatores de diferenciação aos *softwares* tradicionais, necessitando de engenharia própria, com processos, métodos e ferramentas, que fazem uso de uma variedade de critérios de qualidade, tais como, usabilidade, funcionalidade, confiabilidade, eficiência, manutenibilidade, segurança, disponibilidade, escalabilidade e tempo de colocação no mercado.

Referências

BECK, K.; BEEDLE, M.; BENNEKUM, A.; COCKBUM, A.; CUNNINGHAM, W.; FOWLER, M.; GRENNING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R.; MELLOR, S.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. *Manifesto for Agile Software Development*. 2001. Disponível em <http://agilemanifesto.org>. Acesso em: set. de 2010.

BOEHM, B. A Spiral Model of Software Development and Enhancement. *ACM SIGSOFT Software Engineering Notes*, v. 11, n. 4, p. 14-24, 1996.

CECHELERO, Deise; VOLPI, Marlon Marcelo. Engenharia para Aplicações *Web*. *Revista de divulgação técnico-científica do ICPG*, Instituto Catarinense de Pós-Graduação, Blumenau, v. 2, n. 5, p. 49-54, 2004.

DART, S. *Configuration Management: The Missing Link in Web Engineering*. Artech House, 2000.

GINIGE, A.; MURUGESAN, S. The essence of Web engineering: Managing the diversity and complexity of Web application development. *IEEE Multimedia*, p. 22-25, 2001.



Secretaria de Educação
Profissional e Tecnológica



Ministério
da Educação



LOWE, D.; EKLUND, John. *Journal of Web Engineering*, Rinton Editorial, 565 Edmund Terrace, NJ 07652, USA, 2005.

LOWE, D.; HENDERSON-SELLERS, B. Impacts on the development process of differences between Web systems and conventional software systems. In: SSGRR 2001: INTERNATIONAL CONFERENCE ON ADVANCES IN INFRASTRUCTURE FOR ELETRONIC BUSINESS, SCIENCE AND EDUCATION ON THE INTERNET, L'Aquila, Italy, 2001, Scuola Superiore Guglielmo Reiss Romoli, 21.

PRESSMAN, Roger S. *Engenharia de Software*. 6ª ed. São Paulo: McGraw-Hill, 2006.

SOMMERVILLE, Ian. *Engenharia de Software*. 8ª ed. São Paulo: Pearson Education do Brasil Ltda., 2007.



Secretaria de Educação
Profissional e Tecnológica



Ministério
da Educação

