UNIVERSIDADE ABERTA DO BRASIL
UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação
Curso de Bacharelado em Sistemas de Informação

Engenharia de Software

Unidade 1: Conceitos

Prof. Arturo Hernández Domínguez

Conteúdo:

1. Introdução
2. Engenheiro de Software e a Resolução de Problemas
3. Conceitos no Contexto de Engenharia de Software
4. Responsabilidade Profissional e Ética

Exercícios

Agosto – 2010

1. Introducão

Este texto representa uma introdução ao estudo da disciplina de Engenharia de Software (PRESSMAN, 2010; SOMMERVILLE, 2007; PFLEEGER, 1998; MAGELA, 2006). Neste contexto, são apresentados conceitos básicos (SOMMERVILLE, 2007) e considerações éticas (SOMMERVILLE, 2007).

Também é apresentado o código de ética da ACM[1], a IEEE[2] que objetiva fornecer ao aluno de Engenharia de Software um conjunto de informações relacionadas sobre o comportamento do profissional de Engenharia de Software no desenvolvimento da sua pratica profissional.

Na seção 3, serão apresentadas questões e conceitos no contexto de Engenharia de Software, na seção 3, no contexto do engenheiro de software e a resolução de problemas é apresentado o processo de solução de problemas, na seção 4, considerações do ponto de vista responsabilidade profissional e ética são apresentadas, e na seção 5, são apresentadas as considerações finais.

2. Engenheiro de Software e a Resolução de Problemas

O engenheiro de software usa conhecimento de computação para ajudar na resolução de problemas (PFLEEGER, 1998), para isto é essencial o entendimento do problema.

Solucionando problemas

Solucionando problemas é representada através de um processo de análise e síntese (PFLEEGER, 1998).

Análise representa "quebrar" o problema maior em partes ou subproblemas (Figura 1) de tal forma que possamos lidar e entender mais facilmente o problema maior a partir do entendimento de cada subproblema e as interações entre esses subproblemas.

Uma vez realizada a análise, se deve construir a solução do problema a partir da integração da solução especifica de cada subproblema.

---

1 ACM: Association for Computing Machinery

2 IEEE: Institute of Electrical and Electronic Engineers

Síntese representa a elaboração de uma estrutura maior a partir da integração das partes (Figura 2).

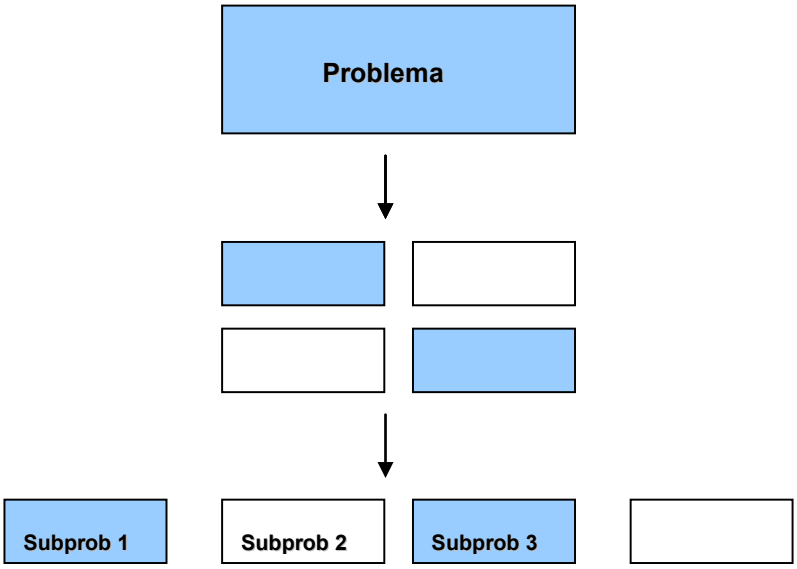

Figura 1: Análise no contexto da resolução de problemas.

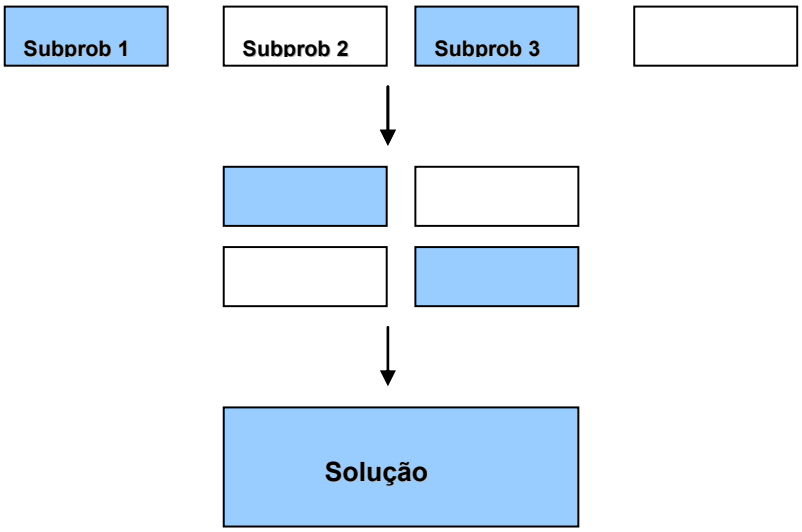

Figura 2: Síntese no contexto da resolução de problemas.

Exemplos no contexto de resolução de problemas:

a construção de uma casa;

a construção de um carro;

a construção de um relógio digital;

a construção de um computador.

No contexto do processo de construção de uma casa (PFLEEGER, 1998):

- Definição e análise dos requisitos da casa. Como queremos a casa ?
- Elaborar e documentar o projeto da casa.

  Planta de uma casa, projeto de fundação, projeto estrutural, projeto hidro-sanitário (hidráulico e esgoto).

- Produção detalhada das especificações da casa. Projeto hidro-sanitário (hidráulico e esgoto). Projeto elétrico e telefônico.
- Identificação e projeto dos espaços e componentes (elementos) da casa.
- Execução da obra: Construção dos espaços e componentes (elementos) da casa. Também considerar Água, Esgoto e Energia elétrica.
- Teste de cada espaço e componente (elemento) da casa. Também testar as instalações de Água, Esgoto e Energia elétrica.
- Integração de todos os espaços, componentes (elementos), água, esgoto e energia elétrica da casa.
- Teste dos serviços da casa.
- Entrega da casa e validação do uso funcional da casa por parte dos moradores.
- Manutenção da casa pelos moradores.

3. Conceitos no Contexto de Engenharia de Software

Nesta seção, objetivando introduzir a disciplina, questões e conceitos no contexto de Engenharia de software são apresentadas.

Porque da engenharia de software?

O contexto no ano de 1968 em relação ao desenvolvimento de sistemas era:

Atraso em projetos importantes, custo superava as previsões, difícil de manter e desempenho insatisfatório. O desenvolvimento de software estava em crise. Objetivando discutir a crise de software, o conceito de Engenharia de Software foi proposto em 1968 (SOMMERVILLE, 2007).

O que é software?

Segundo Sommerville (2007), software é o programa, mas também a documentação e configuração associadas e necessárias para que o programa opere corretamente. Um sistema de software (SOMMERVILLE, 2007) consiste de um conjunto de programas separados; arquivos de configuração; documentação do sistema, que descreve a estrutura do sistema; a documentação do usuário, que explica como usar o sistema.

Características de software

O software é um elemento de um sistema lógico e não de um sistema físico (PRESSMAN, 2010). O software possui características diferentes daquelas do hardware (PRESSMAN, 2010):

1. O software é desenvolvido ou passa por um processo de engenharia; não é fabricado no sentido clássico.

2. O software não se desgasta. Com o passar do tempo o hardware começa a se desgastar. Também o software não é suscetível aos males ambientais que causam desgaste no hardware.

Atributos essenciais de um bom software.

Segundo Sommerville (2007), os atributos essenciais de um sistema de software bem projetado são:

Caraterística do produto	***Descrição***
Facilidade de manutenção	*O software deve ser escrito de modo que possa evoluir para atender às necessidades de mudança dos clientes. A mudança de software é uma consequência inevitável de um ambiente de negócios em constante mutação.*
Confiança	*Um software confiável não deve causar danos físicos ou econômicos no caso de falha no sistema.*
Eficiência	*O software não deve desperdiçar os recursos do sistema, como memoria e ciclos de processador. Portanto, a eficiência inclui tempo de resposta, tempo de processamento, utilização de memoria, etc.*
Usabilidade	*O software deve ser usável, sem esforço excessivo, pelo tipo de usuário para o qual ele foi projetado. Deve apresentar uma interface com o usuário adequada.*

Engenharia de software

Segundo Sommerville (2007), a engenharia de software é uma disciplina de engenharia relacionada com todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até sua manutenção, depois que o sistema entrou em operação.

É importante destacar que a engenharia de software alem de considerar os processos técnicos de desenvolvimento de software, também está relacionada com o gerenciamento de projeto de software e o desenvolvimento de ferramentas, e métodos que apóiem a produção de software (SOMMERVILLE, 2007).

Processo de software

Um processo de software é um conjunto de atividades e resultados associados que produz um produto de software. As atividades fundamentais e comuns a todos os processos de software são (SOMMERVILLE, 2007):

1. Especificação de software: clientes e engenheiros definem o software a ser produzido e as restrições para a sua operação.

2. Desenvolvimento de software: o software é projetado e programado.

3. Validação de software: na qual o software é verificado para garantir que é o que o cliente deseja.

4. Evolução do software: o software é modificado para se adaptar às mudanças dos requisitos do cliente, e do mercado.

Modelo de processo de software

Um modelo de processo de software é uma descrição simplificada de um processo de software (SOMMERVILLE, 2007).

Métodos de engenharia de software

Um método de engenharia de software segundo Sommerville (2007), é uma abordagem estruturada para desenvolvimento de software, objetivando facilitar a produção de software de alta qualidade dentro de custos adequados.

Todos os métodos estão baseados em modelos de desenvolvimento de um sistema, que pode ser representado graficamente, os modelos representam a especificação e projeto de um sistema.

Os componentes de um método são (SOMMERVILLE, 2007):

- Modelos e notação usada para a definição desses modelos.

  Exemplos de modelos: Modelo de objetos (Diagrama de Objetos), Modelo de máquina de estados (Diagrama de Estados), Diagrama de Classes, Diagrama de Componentes, Diagrama de Atividades, Diagrama de Implantação.

  Notação: UML (Unified Modeling Language) – Linguagem Unificada de Modelagem de sistemas orientados a objetos (BOOCH et al., 2005).

- Regras ou restrições que são sempre aplicáveis aos modelos do sistema.

  Exemplo de regra: Cada entidade em um modelo deve ter um único nome

- Guia de processo. Descrições das atividades que devem ser seguidas para desenvolver os modelos de sistema e a organização dessas atividades.

- Recomendações. Heurísticas que caracterizam uma boa pratica de projeto nesse método.

CASE (Computer-Aided Software Engineering)

CASE (Computer-Aided Software Engineering) corresponde a Engenharia de Software Auxiliada por Computador, isto é um conjunto de programas usados para dar apoio às atividades de processo de software, tais como análise de requisitos, modelagem do sistema, programação e teste.

Exemplos de Tecnologia CASE:

- Ferramentas de edição (editor de texto, editor de diagramas);

- Ferramentas de apoio a métodos (editores dos modelos de análise e projeto, dicionário de dados e gerador de código);
- Ferramentas de programação (um ambiente integrado para o desenvolvimento de software – IDE);
- Ferramentas de teste (geradores de dados de teste).

4. Responsabilidade Profissional e Ética

O trabalho, no contexto de engenharia de software, implica responsabilidades mais amplas do que a aplicação de habilidades técnicas. Se deve comportar de forma responsável ética e moralmente (SOMMERVILLE, 2007). Se deve defender comportamentos padrões normais de honestidade e integridade. Em algumas áreas um comportamento aceitável está associado a uma noção de responsabilidade profissional, exemplos (SOMMERVILLE, 2007):

Área	***Descrição***
Confidencialidade	*Se deve respeitar a confidencialidade dos funcionários ou clientes. Independentemente de ter ou não assinado um acordo formal.*
Competência	*Não se deve conscientemente aceitar um trabalho que esteja fora da sua competência.*
Direitos sobre propriedade intelectual	*Se deve ter cuidado para assegurar que a propriedade intelectual de funcionários e clientes seja protegida.*
Mau uso de computadores	*Não se deve usar as habilidades técnicas para fazer mau uso dos computadores de outras pessoas.*

4.1 Código de Ética e Pratica Profissional da Engenharia de Software

Um código de conduta profissional ou código de ética foi publicado pelas organizações internacionais (SOMMERVILLE, 2007): ACM, a IEEE e British Computer Society.

O fundamento do código de ética da ACM/IEEE é:

> Os computadores desempenham um papel central e crescente no comercio, na industria , no governo, na medicina, na educação, no entretenimento e na sociedade em geral. Os engenheiros de software são aqueles que contribuem, por participação direta ou pelo ensino, com a análise, especificação, projeto, desenvolvimento, certificação, manutenção e teste de sistemas de software. Em razão de seu papel no desenvolvimento de sistemas de software, ele têm

> oportunidades significativas de praticar o bem ou de causar o mal, tornarem outros capazes de praticar o bem ou causar o mal ou de influenciar outros indivíduos a praticar o bem ou causar o mal. Para garantir, tanto quanto possível, que seus esforços sejam utilizados para o bem, os engenheiros de software devem se comprometer a fazer da engenharia de software uma profissão benéfica e respeitada (SOMMERVILLE, 2007).

Os oito princípios do código de ética da ACM/IEEE são (SOMMERVILLE, 2007):

Principio do Código de Ética	***Descrição***
1. Público	*Os engenheiros de software devem agir consistentemente com o interesse público.*
2. Cliente e Empregador	*Os engenheiros de software devem agir dentro dos melhores interesses de seu cliente e empregador, de forma consistente com o interesse público.*
3. Produto	*Os engenheiros de software devem assegurar que seus produtos e as modificações a eles relacionadas atendam aos mais altos padrões profissionais possíveis.*
4. Julgamento	*Os engenheiros de software devem manter a integridade e a independência em seu julgamento profissional.*
5. Gerenciamento	*Os gerentes e líderes de engenharia de software devem aceitar e promover uma abordagem ética no gerenciamento de desenvolvimento e manutenção de software.*
6. Profissão	*Os engenheiros de software devem promover a integridade e a reputação da profissão de forma consistente com o interesse público*
7. Colegas	*Os engenheiros devem ser honestos e colaborativos com seus colegas.*
8. Indivíduo	*Os engenheiros de software devem participar, ao longo da vida, aprendendo, respeitando e promovendo uma abordagem ética na prática da profissão.*

Conforme Massiero e Bigonha (2008) “Falhas éticas graves, quando ocorrem, podem causar severos prejuízos às organizações e às pessoas envolvidas”.

5. Considerações Finais

Conforme citado inicialmente este texto representa uma introdução ao estudo da Engenharia de Software. Objetiva fornecer um conjunto de conceitos básicos e também contextualiza a atividade do engenheiro de software do ponto de vista ético e responsabilidade profissional.

No contexto de Engenharia de Software foram apresentados os seguintes conceitos: software, características de software, atributos essenciais de um bom software, engenharia de software, processo de software, método, ferramenta CASE.

No contexto do engenheiro de software e a resolução de problemas foi apresentado o processo de solução de problemas.

Também foram apresentadas algumas considerações de natureza ética e o código de ética (oito princípios) da ACM/IEEE foi apresentado.

Referências Bibliográficas

Booch G., Rumbaugh J., Jacobson I. UML: Guia do usuário, 2ª edição. Editora Campus. 2005.

Magela, R. Engenharia de Software Aplicada: Princípios (volume 1). Alta Books. 2006.

Massiero P. C., Bigonha R. S. Ética e Computação, Volume 1 - Número 1 - Dezembro, SBC HORIZONTES. 2008.

Pfleeger S. L. Software Engineering: theory and practice. Prentice-Hall. 1998.

Pressman R. Engenharia de Software, 6a edição, AMGH Editora. 2010.

Software Engineering Institute, Disponivel em: <http://www.sei.cmu.edu>. Acesso em 1 ago. 2010.

Sommerville, I. Engenharia de Software. 8a Edição. Addison Wesley. 2007.

Unified Modeling Language, Disponivel em: <http://www.uml.org>. Acesso em 1 ago.2010.