

Disciplina:

Desenvolvimento de Software

INTRODUÇÃO À ENGENHARIA DE SOFTWARE

Fonte:

Os slides aqui apresentados estão disponíveis (em inglês) em: <https://iansommerville.com/software-engineering-book/slides/> para o livro: Software Engineering, 10th Edition, Ian Sommerville.

Tradução, adaptação e design para o Português:
Sergio Antônio Andrade de Freitas, 2021.



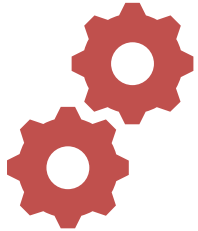
Software engineering

- ✧ As economias de TODAS as nações desenvolvidas dependem do software.
- ✧ Cada vez mais sistemas são controlados por software.
- ✧ A engenharia de software está preocupada com teorias, métodos e ferramentas para o desenvolvimento de software profissional.
- ✧ O gasto com software representa uma fração significativa do PIB em todos os países desenvolvidos.

Custos de software

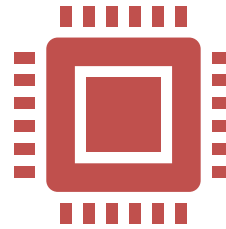
- ✧ Os custos de software muitas vezes dominam os custos do sistema de computador. Os custos do software em um PC são muitas vezes maiores do que o custo de hardware.
- ✧ O software custa mais para manter do que para desenvolver. Para sistemas com uma vida útil longa, os custos de manutenção podem ser várias vezes os custos de desenvolvimento.
- ✧ A engenharia de software está preocupada com o desenvolvimento de software econômico.

Falha no projeto de software



Aumento da complexidade do sistema

À medida que novas técnicas de engenharia de software nos ajudam a construir sistemas maiores e mais complexos, as demandas mudam. Os sistemas devem ser construídos e entregues mais rapidamente; sistemas maiores e ainda mais complexos são necessários; os sistemas têm que ter novas capacidades que antes eram consideradas impossíveis.



Falha no uso de métodos de engenharia de software

É bastante fácil escrever programas de computador sem usar métodos e técnicas de engenharia de software. Muitas empresas entraram no desenvolvimento de software à medida que seus produtos e serviços evoluíram. Eles não usam métodos de engenharia de software em seu trabalho cotidiano. Consequentemente, seu software é muitas vezes mais caro e menos confiável do que deveria ser.

FAQ - Engenharia de Software

| Pergunta | Resposta |
|---|---|
| O que é software? | Programas de computador e documentação associada. Os produtos de software podem ser desenvolvidos para um determinado cliente ou podem ser desenvolvidos para um mercado geral. |
| Quais são os atributos de um bom software? | Um bom software deve entregar a funcionalidade e o desempenho necessários ao usuário e deve ser mantido, confiável e utilizável. |
| O que é engenharia de software? | Engenharia de software é uma disciplina de engenharia que se preocupa com todos os aspectos da produção de software. |
| Quais são as atividades fundamentais de engenharia de software? | Especificação de software, desenvolvimento de software, validação de software e evolução de software. |
| Qual é a diferença entre engenharia de software e ciência da computação? | A ciência da computação se concentra na teoria e nos fundamentos; a engenharia de software está preocupada com as praticidades de desenvolver e fornecer softwares úteis. |
| Qual é a diferença entre engenharia de software e engenharia de sistemas? | A engenharia de sistemas está preocupada com todos os aspectos do desenvolvimento de sistemas baseados em computador, incluindo hardware, software e engenharia de processos. A engenharia de software faz parte desse processo mais geral. |

FAQ - Engenharia de Software

| Pergunta | Resposta |
|---|---|
| Quais são os principais desafios enfrentados pela engenharia de software? | Enfrentando o aumento da diversidade, demandas por redução dos prazos de entrega e desenvolvimento de software confiável. |
| Quais são os custos da engenharia de software? | Cerca de 60% dos custos de software são custos de desenvolvimento, 40% são custos de teste. Para softwares personalizados, os custos de evolução geralmente excedem os custos de desenvolvimento. |
| Quais são as melhores técnicas e métodos de engenharia de software? | Embora todos os projetos de software tenham que ser gerenciados e desenvolvidos profissionalmente, diferentes técnicas são apropriadas para diferentes tipos de sistema. Por exemplo, os jogos devem ser sempre desenvolvidos usando uma série de protótipos, enquanto os sistemas de controle críticos de segurança requerem uma especificação completa e analisável para serem desenvolvidos. Você não pode, portanto, dizer que um método é melhor do que outro. |
| Que diferenças a web fez com a engenharia de software? | A web levou à disponibilidade de serviços de software e à possibilidade de desenvolver sistemas altamente distribuídos baseados em serviços. O desenvolvimento de sistemas baseados na Web levou a importantes avanços em linguagens de programação e reutilização de software. |

Produtos de software

Produtos genéricos

- Sistemas autônomos que são comercializados e vendidos para qualquer cliente que deseje comprá-los.
- Exemplos – Softwares de PC como programas gráficos, ferramentas de gerenciamento de projetos; Software CAD; software para mercados específicos, como sistemas de consultas para dentistas.

Produtos personalizados

- Software que é encomendado por um cliente específico para atender às suas próprias necessidades.
- Exemplos : sistemas de controle embarcados, software de controle de tráfego aéreo, sistemas de monitoramento de tráfego.

Especificação do produto

Produtos genéricos

- A especificação do que o software deve fazer é de propriedade do desenvolvedor de software e as decisões sobre a mudança de software são tomadas pelo desenvolvedor.

Produtos personalizados

- A especificação do que o software deve fazer é de propriedade do cliente para o software e eles tomam decisões sobre mudanças de software que são necessárias.

Atributos essenciais de um bom software

| Característica produto | Descrição |
|----------------------------|---|
| Manutenção | O software deve ser escrito de tal forma para que possa evoluir para atender às necessidades em mudança dos clientes. Este é um atributo crítico porque a mudança de software é um requisito inevitável de um ambiente de negócios em mudança. |
| Confiabilidade e segurança | A confiança num software inclui uma série de características, incluindo confiabilidade e segurança. O software confiável não deve causar danos físicos ou econômicos em caso de falha do sistema. Usuários mal-intencionados não devem ser capazes de acessar ou danificar o sistema. |
| Eficiência | O software não deve desperdiçar os recursos do sistema, como ciclos de memória e processador. A eficiência inclui, portanto, responsividade, tempo de processamento, utilização da memória, etc.. |
| Aceitabilidade | O software deve ser aceitável para o tipo de usuários para os quais ele foi projetado. Isso significa que ele deve ser compreensível, utilizável e compatível com outros sistemas que eles usam. |

Engenharia de software



A Engenharia de software se preocupa com todos os aspectos da produção de software desde os estágios iniciais da especificação do sistema até a manutenção do sistema depois que ele entrou em uso.



Parte engenharia

Usando teorias e métodos apropriados para resolver problemas tendo em conta as restrições organizacionais e financeiras.



Todos os aspectos da produção de software

Não apenas processo técnico de desenvolvimento. Também o gerenciamento de projetos e o desenvolvimento de ferramentas, métodos etc. para apoiar a produção de software.

Importância da engenharia de software

- ✧ Cada vez mais, indivíduos e sociedade dependem de sistemas avançados de software. Precisamos ser capazes de produzir sistemas confiáveis e confiáveis economicamente e rapidamente.
- ✧ Geralmente é mais barato, a longo prazo, usar métodos e técnicas de engenharia de software para sistemas de software em vez de apenas escrever os programas como se fosse um projeto de programação pessoal. Para a maioria dos tipos de sistema, a maioria dos custos são os custos de alterar o software depois que ele entrou em uso.

Atividades de processos de software

- ✧ **Especificação do software**, onde clientes e engenheiros definem o software a ser produzido e as restrições em sua operação.
- ✧ **Desenvolvimento do software**, onde o software é projetado e programado.
- ✧ **Validação do software**, onde o software é verificado para garantir que é o que o cliente exige.
- ✧ **Evolução do software**, onde o software é modificado para refletir as mudanças nas necessidades do cliente e do mercado.

Principais problemas que afetam o software

✧ Heterogeneidade

- Cada vez mais os sistemas são obrigados a operar como sistemas distribuídos em redes que incluem diferentes tipos de computadores e dispositivos móveis.

✧ Mudanças comerciais e sociais

- Os negócios e a sociedade estão mudando incrivelmente rapidamente à medida que as economias emergentes se desenvolvem e novas tecnologias se tornam disponíveis. Eles precisam ser capazes de mudar seu software existente e desenvolver rapidamente novos softwares.

Principais problemas que afetam o software

✧ Segurança e confiança

- Como o software está entrelaçado com todos os aspectos de nossas vidas, é essencial que possamos confiar nesse software.

✧ Escala

- O software tem que ser desenvolvido em uma gama ampla de escalas, desde sistemas incorporados muito pequenos em dispositivos portáteis ou vestíveis até sistemas baseados em nuvem em escala de Internet que atendem a uma comunidade global.

Diversidade de engenharia de software

Existem muitos tipos diferentes de sistema de software e não há um conjunto universal de técnicas de software que é aplicável a todos estes.

Os métodos de engenharia de software e as ferramentas utilizadas dependem do tipo de aplicação que está sendo desenvolvida, dos requisitos do cliente e do histórico da equipe de desenvolvimento.

Tipos de aplicações

✧ Aplicativos autônomos (*stand alone*)

- Estes são sistemas de aplicação que são executados em um computador local, como um PC. Eles incluem todas as funcionalidades necessárias e não precisam ser conectados a uma rede.

✧ Aplicativos interativos baseados em transações

- Aplicativos que são executados em um computador remoto e são acessados por usuários de seus próprios PCs ou terminais. Estes incluem aplicativos web, como aplicativos de e-commerce.

✧ Sistemas de controle embarcados

- Estes são sistemas de controle de software que controlam e gerenciam dispositivos de hardware. Numericamente, há provavelmente mais sistemas embarcados do que qualquer outro tipo de sistema.

Tipos de aplicações



Sistemas de processamento em lote (*batch*)

Estes são sistemas de negócios que são projetados para processar dados em grandes lotes. Eles processam um grande número de entradas individuais para criar saídas correspondentes.



Sistemas de entretenimento

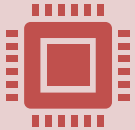
Estes são sistemas que são principalmente para uso pessoal e que são destinados a entreter o usuário.



Sistemas para modelagem e simulação

São sistemas que são desenvolvidos por cientistas e engenheiros para modelar processos físicos ou situações, que incluem muitos objetos, separados e interagindo.

Tipos de aplicações



Sistemas de coleta de dados

Estes são sistemas que coletam dados de seu ambiente usando um conjunto de sensores e enviam esses dados para outros sistemas para processamento.



Sistemas de sistemas

Estes são sistemas que são compostos por uma série de outros sistemas de software.

Fundamentos de engenharia de software

- ✧ Alguns princípios fundamentais se aplicam a todos os tipos de sistema de software, independentemente das técnicas de desenvolvimento utilizadas:
 - Os sistemas devem ser desenvolvidos utilizando um processo de desenvolvimento gerenciado e compreendido. Claro, diferentes processos são usados para diferentes tipos de software.
 - Confiabilidade e desempenho são importantes para todos os tipos de sistema.
 - Entender e gerenciar a especificação e os requisitos do software (o que o software deve fazer) são importantes.
 - Quando apropriado, você deve reutilizar o software que já foi desenvolvido em vez de escrever um novo software.

Engenharia de software para internet

- ✧ A Web é agora uma plataforma para a execução de aplicativos e as organizações estão cada vez mais desenvolvendo sistemas baseados na Web em vez de sistemas locais.
- ✧ Os serviços web permitem que a funcionalidade do aplicativo seja acessada pela web.
- ✧ A computação em nuvem é uma abordagem para a prestação de serviços de computador onde os aplicativos são executados remotamente na 'nuvem'.
 - Usuários não compram software comprar pagamento de acordo com o uso.

Engenharia de software baseada na Web

- ✧ Sistemas baseados na Web são sistemas distribuídos complexos, mas os princípios fundamentais da engenharia de software discutidos anteriormente são tão aplicáveis a eles quanto a quaisquer outros tipos de sistema.
- ✧ As ideias fundamentais da engenharia de software aplicam-se ao software baseado na Web da mesma forma que se aplicam a outros tipos de sistema de software.

Engenharia de software na web

✧ Reuso de software

- O reaproveitamento de software é uma abordagem dominante para a construção de sistemas baseados na Web. Ao construir esses sistemas, você pensa em como você pode montá-los a partir de componentes e sistemas de software pré-existent.

✧ Desenvolvimento incremental e ágil

- Os sistemas baseados na Web devem ser desenvolvidos e entregues incrementalmente. Agora é geralmente reconhecido que é impraticável especificar todos os requisitos para tais sistemas com antecedência.

Engenharia de software na web

✧ Sistemas orientados a serviços

- O software pode ser implementado usando engenharia de software orientada a serviços, onde os componentes de software são serviços web autônomos.

✧ Interfaces avançadas

- Tecnologias de desenvolvimento de interface, como AJAX e HTML5, surgiram que suportam a criação de interfaces ricas dentro de um navegador web.

Ética em engenharia de software



A engenharia de software envolve responsabilidades mais amplas do que simplesmente a aplicação de habilidades técnicas.



Os engenheiros de software devem se comportar de forma honesta e eticamente responsável se quiserem ser respeitados como profissionais.



O comportamento ético é mais do que simplesmente defender a lei, mas envolve seguir um conjunto de princípios que são moralmente corretos.

Questões de responsabilidade profissional



Confidencialidade

Os engenheiros normalmente devem respeitar a confidencialidade de seus empregadores ou clientes, independentemente de um acordo formal de confidencialidade ter sido assinado ou não.



Competência

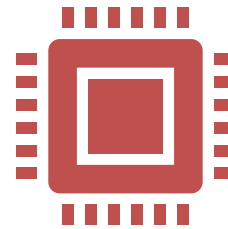
Os engenheiros não devem deturpar seu nível de competência. Eles não devem aceitar conscientemente o trabalho que está fora com sua competência.

Questões de responsabilidade profissional



Direitos de propriedade intelectual

Os engenheiros devem estar cientes das leis locais que regem o uso de propriedade intelectual, como patentes, direitos autorais, etc. Eles devem ter cuidado para garantir que a propriedade intelectual de empregadores e clientes seja protegida.



Uso indevido de computador

Os engenheiros de software não devem usar suas habilidades técnicas para usar indevidamente os computadores de outras pessoas. O uso indevido de computador varia de relativamente trivial (jogo jogando na máquina de um empregador, digamos) a extremamente grave (disseminação de vírus).

Lógica do código de ética

- Os computadores têm um papel central e crescente no comércio, indústria, governo, medicina, educação, entretenimento e sociedade em geral. Os engenheiros de software são aqueles que contribuem por participação direta ou por ensino, para análise, especificação, design, desenvolvimento, certificação, manutenção e testes de sistemas de software.
- Devido às suas funções no desenvolvimento de sistemas de software, os engenheiros de software têm oportunidades significativas de fazer o bem ou causar danos, para permitir que outros façam o bem ou causem danos, ou influenciem outros a fazer o bem ou causar danos. Para garantir, tanto quanto possível, que seus esforços serão usados para o bem, os engenheiros de software devem comprometer-se a tornar a engenharia de software uma profissão benéfica e respeitada.

iLearn: Um ambiente de aprendizagem digital



Um ambiente de aprendizagem digital é uma estrutura na qual um conjunto de ferramentas de uso geral e especialmente projetadas para a aprendizagem pode ser incorporado, além de um conjunto de aplicações voltadas para as necessidades dos alunos que usam o sistema.



As ferramentas incluídas em cada versão do ambiente são escolhidas por professores e alunos para atender às suas necessidades específicas.

Estes podem ser aplicações gerais, como planilhas, aplicativos de gerenciamento de aprendizagem, como um Virtual Learning Environment (VLE) para gerenciar a submissão e avaliação da lição de casa, jogos e simulações.

Sistemas orientados a serviços

- ✧ O sistema é um sistema orientado a serviços com todos os componentes do sistema considerados um serviço substituível.
- ✧ Isso permite que o sistema seja atualizado gradualmente à medida que novos serviços se tornam disponíveis.
- ✧ Também permite configurar rapidamente o sistema para criar versões do ambiente para diferentes grupos, como crianças muito pequenas que não sabem ler, estudantes seniores, etc..

Serviços do iLearn

- ✧ Serviços utilitários que fornecem funcionalidade básica independente de aplicativos e que podem ser usados por outros serviços no sistema.
- ✧ Serviços de aplicativos que fornecem aplicativos específicos, como e-mail, conferência, compartilhamento de fotos etc. e acesso a conteúdos educacionais específicos, como filmes científicos ou recursos históricos.
- ✧ Serviços de configuração que são usados para adaptar o ambiente com um conjunto específico de serviços de aplicativos e definir como os serviços são compartilhados entre alunos, professores e seus pais.

Arquitetura do iLearn

Browser-based user interface

iLearn app

Configuration services

Group
management

Application
management

Identity
management

Application services

Email Messaging Video conferencing Newspaper archive
Word processing Simulation Video storage Resource finder
Spreadsheet Virtual learning environment History archive

Utility services

Authentication
User storage

Logging and monitoring
Application storage

Interfacing
Search

Integração de serviços no iLearn

- ✧ Serviços integrados são serviços que oferecem uma API (interface de programação de aplicativos) e que podem ser acessados por outros serviços através dessa API. A comunicação direta de serviço para serviço é, portanto, possível
- ✧ Serviços independentes são serviços que são simplesmente acessados através de uma interface de navegador e que operam independentemente de outros serviços. As informações só podem ser compartilhadas com outros serviços por meio de ações explícitas do usuário, como copiar e colar; a re-autenticação pode ser necessária para cada serviço independente.

Concluindo

A Engenharia de software se preocupa com todos os aspectos da produção de software.

Atributos essenciais do produto de software são manutenção, confiabilidade e segurança, eficiência e aceitabilidade.

As atividades de alto nível de especificação, desenvolvimento, validação e evolução fazem parte de todos os processos de software.

As noções fundamentais de engenharia de software são universalmente aplicáveis a todos os tipos de desenvolvimento de sistemas.

Concluindo



Existem muitos tipos diferentes de sistema e cada um requer ferramentas e técnicas adequadas de engenharia de software para seu desenvolvimento.



As ideias fundamentais da engenharia de software são aplicáveis a todos os tipos de sistema de software.



Os engenheiros de software têm responsabilidades com a profissão de engenharia e com a sociedade. Eles não devem simplesmente se preocupar com questões técnicas.



Código de ética.