projeção

Princípios gerais

Princípios

- David Hooker propôs sete princípios que se concentram na pratica da engenharia de software como um todo:
  1. A razão de existir;
  2. KISS;
  3. Mantenha a visão;
  4. O que um produz outros consomem;
  5. Esteja aberto para o futuro;
  6. Planeje com antecedência;
  7. Pense!

A razão de existir

- Um sistema de software existe por uma única razão: **gerar valor a seus usuários**. Todas as decisões deveriam ser tomadas tendo esse princípio em mente.
- Antes de especificar uma necessidade de um sistema, antes de indicar alguma parte da funcionalidade de um sistema, antes de determinar as plataformas de hardware ou os processos de desenvolvimento, pergunte a si mesmo: “Isso realmente agrega valor real ao sistema?”. Se a resposta for “não”, não o faça!

KISS (Keep It Simple, Stupid!)

- O projeto de software não é um processo casual; há muitos fatores a ser considerados em qualquer esforço de projeto — todo projeto deve ser o mais simples possível, mas não tão simples assim. Esse principio contribui para um sistema mais fácil de compreender e manter;
- Isso não significa "rápido e malfeito" — na realidade, simplificar exige muita analise e trabalho durante as iterações, sendo que o resultado será um software de fácil manutenção e menos propenso a erros.

Mantenha a visão

- Uma visão clara é essencial para o sucesso. Sem ela, um projeto se torna ambíguo. Sem uma integridade conceitual, corre-se o risco de transformar o projeto numa colcha de retalhos de projetos incompatíveis, unidos por parafusos inadequados.

O que um produz outros consomem

- Raramente um sistema de software é construído e utilizado de forma isolada. De uma maneira ou de outra, alguém mais irá usar, manter, documentar ou, de alguma forma, dependerá da capacidade de entender seu sistema;
- Especifique tendo em vista os usuários; projete, tendo em mente os implementadores; e codifique considerando aqueles que terão de manter e estender o sistema.

Esteja aberto para o futuro

- Um sistema com tempo de vida mais longo tem mais valor. Nos ambientes computacionais de hoje, em que as especificações mudam de um instante para outro e as plataformas de hardware se tornam rapidamente obsoletas, a vida de um software, em geral, é medida em meses. Jamais faca projetos limitados, sempre pergunte “e se” e prepare-se para todas as possíveis respostas, criando sistemas que resolvam o problema geral, não apenas aquele especifico. Isso muito provavelmente conduziria à reutilização de um sistema inteiro.

Planeje com antecedência

- A reutilização economiza tempo e esforço. Alcançar um alto grau de reuso é indiscutivelmente a meta mais difícil de ser atingida ao se desenvolver um sistema de software. A reutilização de código e projetos tem sido proclamada como o maior beneficio do uso de tecnologias orientadas a objetos, entretanto, o retorno desse investimento não é automático. Planejar com antecedência para o reuso reduz o custo e aumenta o valor tanto dos componentes reutilizáveis quanto dos sistemas aos quais eles serão incorporados.

Pense!

- Pensar bem e de forma clara antes de agir quase sempre produz melhores resultados. Quando se analisa alguma coisa, provavelmente esta sairá correta.

Mitos

- Atualmente, a maioria dos profissionais que trabalham com engenharia de software reconhece os mitos por aquilo que eles representam — atitudes enganosas que provocaram sérios problemas tanto para gerentes quanto para praticantes da área. Entretanto, antigos hábitos e atitudes são difíceis de ser modificados e resquícios de mitos de software permanecem.

Gerenciamento

- Mito ou Verdade?
  - Já temos um livro que está cheio de padrões e procedimentos para desenvolver software. Ele não supre meu pessoal com tudo que eles precisam saber?

Mito do Gerenciamento

- O livro com padrões pode muito bem existir, mas ele é usado?
- Os praticantes da área estão cientes de que ele existe?
- Esse livro reflete a pratica moderna da engenharia de software? É completo?
- É adaptável?

Em muitos casos, a resposta para todas essas perguntas é não!!

Gerenciamento

- Mito ou Verdade?
  - Se o cronograma atrasar, poderemos acrescentar mais programadores e ficaremos em dia.

Mitos de gerenciamento

- O desenvolvimento de software não é um processo mecânico como o de fábrica;
- Acrescentar pessoas num projeto de software atrasado só o tornará mais atrasado ainda;
- Quando novas pessoas entram, as que já estavam terão de gastar tempo situando os recém-chegados, reduzindo, consequentemente, o tempo destinado ao desenvolvimento produtivo.

Cliente

- Mito ou Verdade?
  - Os requisitos de software mudam continuamente, mas as mudanças podem ser facilmente assimiladas, pois o software é flexível.

Mito do cliente

- É verdade que os requisitos de software mudam, mas o impacto da mudança varia dependendo do momento em que ela foi introduzida;
- Quando as mudanças dos requisitos são solicitadas cedo (antes do projeto ou da codificação terem começado), o impacto sobre os custos é relativamente pequeno. Entretanto, conforme o tempo passa, ele aumenta rapidamente.

Profissional

- Mito ou verdade?
  - Uma vez feito um programa e colocado em uso, nosso trabalho está terminado

Mito do profissional

- Uma vez alguém já disse que “Quanto antes se começar a codificar, mais tempo levará para terminá-lo”. Levantamentos indicam que entre 60% a 80% de todo o esforço será despendido após a entrega do software ao cliente pela primeira vez.

Profissional

- Mito ou verdade?:
  - Até que o programa entre em funcionamento, não há maneira de avaliar sua qualidade.

Mito do profissional

- Um dos mecanismos de garantia da qualidade de software mais eficaz pode ser aplicado desde a concepção de um projeto — a revisão técnica;
- Revisores de software são um “filtro de qualidade” que mostram ser mais eficientes do que testes para encontrar certas classes de defeitos de software.

Resumo

- Softwares — programas, dados e informações descritivas — contemplam uma ampla gama de áreas de aplicação e tecnologia. O software legado continua a representar desafios especiais aqueles que precisam fazer sua manutenção.

Resumo

- A engenharia de software engloba processos, métodos e ferramentas que possibilitam a construção de sistemas complexos baseados em computador dentro do prazo e com qualidade.
- A prática da engenharia de software é uma atividade de resolução de problemas que segue um conjunto de princípios básicos.