ENGENHARIA DE SOFTWARE

Síntese de tópicos importantes

PRESSMAN, Roger S.

Conteúdo

- Componentes e tipos de software
- Problemas com o software e suas causas
- Mitologia que envolve o software
- Configuração de software
- Atividade de pesquisa

Componentes do Software

- Um sistema informatizado é formado por dois tipos de componentes:
  - Executáveis em máquina e
  - Não executáveis em máquina
- Os componentes do software devem mapear as exigências do cliente em código executável.

Tipos Comuns de Software

- **Básico**: compiladores, editores simples, drivers, componentes do SO.
- **Tempo Real**: monitora, analisa e controla eventos em tempo real.
- **Comercial**: controle de estoque, vendas etc. (manipulam algum mecanismo de persistência, como uma de suas principais características)

Tipos Comuns de Software

- **Científico** e de **Engenharia**: intenso processamento de números e cálculos.
- **Embutido** ou **Embarcado**: celulares, micro-ondas, injeção eletrônica.
- **Pessoal**: processador de texto, planilha, jogos, apresentações etc.
- **Inteligência Artificial**: sistemas especialistas, redes neurais e aprendizado.

Principais Problemas

- **Estimativas** de prazo (meses, anos) e custo imprecisas
- **Produtividade** abaixo da praticada pelo mercado
- Software de baixa **qualidade** (erros e não conformidades com requisitos que tiram a confiança do cliente sobre o produto)

Mais Problemas

- Não dedicamos **tempo para coletar dados** sobre o software e seu processo de desenvolvimento. Com **poucos dados históricos** como guia, as estimativas têm sido “a olho”, com resultados previsivelmente ruins.
- Sem nenhum **indicador** sólido de **produtividade**, não poderemos avaliar com precisão a eficácia de novas ferramentas, métodos, padrões ou processos.

Mais Ainda

- A **insatisfação do cliente** com o sistema "concluído" ocorre muito freqüentemente.
- Os projetos de desenvolvimento de software normalmente são levados a efeito apenas com um vago indício das exigências do cliente.
- A **comunicação** entre o cliente e o desenvolvedor de software freqüentemente é muito fraca.

Ainda Não Acabou

- A **qualidade** do software freqüentemente é suspeita. Somente agora estão começando a ser seguidos conceitos quantitativos sólidos de confiabilidade e de **garantia de qualidade** de software.
- Só recentemente começamos a entender a importância dos **testes de software sistemáticos** e tecnicamente **completos**.

Mais Um

- O software existente pode ser muito **difícil de manter.** A tarefa de manutenção de software devora a maioria de todos os dólares destinados a software. A capacidade de **manutenção** de software não foi enfatizada como um **critério importante** para a **aceitação** do software.

Causas

- Gerentes sem vivência em:
  - **Projetos** e seus marcos de evolução
  - Métodos efetivos de **controle**
  - **Tecnologias** que se modificam rapidamente

Mais causas

- Os programadores ou engenheiros de software têm **pouca instrução** formal, holística e estudam pouco as técnicas de desenvolvimento e as áreas de negócio.
- Cada pessoa aborda a tarefa de "escrever programas" com a experiência advinda de **esforços passados**. Algumas pessoas desenvolvem uma abordagem ordeira e eficiente, mesmo por **tentativa e erro**, mas muitas criam maus hábitos, que resultam em **qualidade** e **manutenibilidade** deficientes.

Outras Causas

- Resistência às inevitáveis **mudanças**
- É irônico que enquanto o hardware experimenta enormes mudanças, as **pessoas** da área de software responsáveis pelo aproveitamento desse potencial, muitas vezes se oponham à mudança quando ela é discutida e resistam a ela quando ela é introduzida.

Mitologia do Software

- **Mitos Administrativos.** Advém de gerentes sobre pressão de **orçamento** e **tempo.**
- **Mitos do Cliente.** Advém de falsas expectativas e insatisfação com o desenvolvedor
- **Mitos do Profissional de Desenvolvimento.** Advém de se considerar o software como uma forma de arte. Será que o software é uma arte ou uma engenharia?

Mito do Manual de Práticas

- **Mito:** Já temos um manual repleto de padrões e procedimentos para a construção de software. Isso não oferecerá ao meu pessoal tudo o que eles precisam saber?
- **Realidade:** O manual de padrões pode muito bem existir, mas será que ele é usado? Os profissionais de software têm conhecimento de sua existência? Ele reflete a moderna prática de desenvolvimento de software? É completo? Em muitos casos, a resposta a todas estas perguntas é “não”.

Mito do Computador Moderno

- **Mito:** Meu pessoal tem ferramentas de desenvolvimento de software de última geração; afinal de contas lhes compramos os mais novos computadores.
- **Realidade:** É preciso muito mais do que o último modelo de computador para se fazer um desenvolvimento de software de alta qualidade. As ferramentas de engenharia de software auxiliadas por computador (CASE) são mais importantes do que o hw para se conseguir boa qualidade e produtividade; contudo, a maioria dos desenvolvedores de software não as usa plenamente.

Mito das Hordas de Mongóis

- **Mito:** Se nós estamos atrasados nos prazos, podemos adicionar mais programadores e tirar o atraso.
- **Realidade:** O desenvolvimento de software não é um processo mecânico igual à manufatura. Acrescentar pessoas em um projeto de software atrasado torna-o ainda mais atrasado. Gasta-se tempo formando os recém-chegados, o que reduz o tempo de desenvolvimento produtivo. Pessoas podem ser acrescentadas, mas somente de uma forma planejada e bem coordenada.

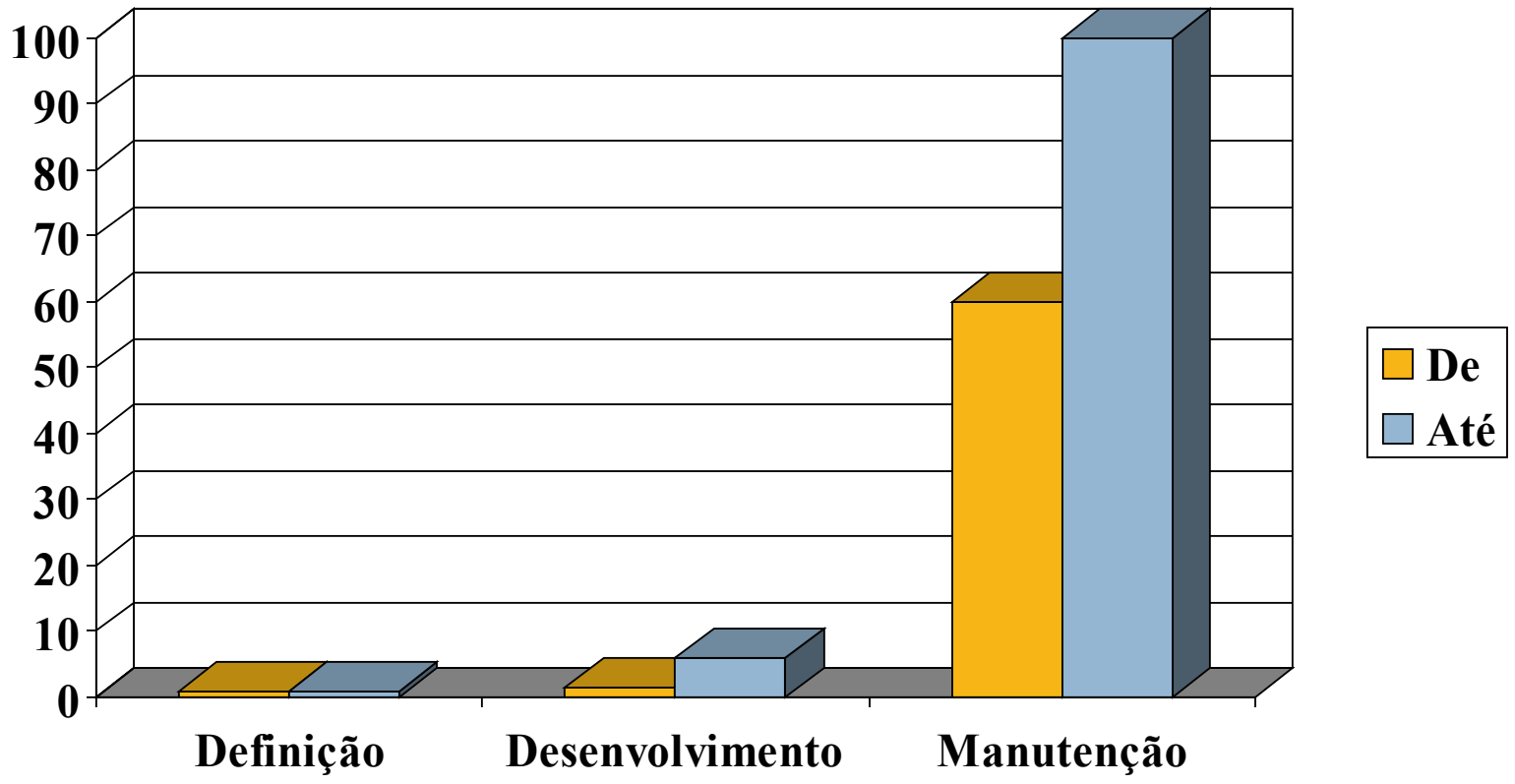
Mitos do Cliente: Especificação

- **Mito**: Uma declaração geral dos objetivos é suficiente para se começar a escrever programas – podemos preencher os detalhes mais tarde.
- **Realidade**: Uma definição inicial ruim é a principal causa de fracasso dos esforços de desenvolvimento de software. Uma descrição formal e detalhada do domínio da informação, função, desempenho, interfaces, restrições de projeto e critérios de validação é fundamental. Essas características podem ser determinadas somente depois de cuidadosa comunicação entre o cliente e o desenvolvedor.

O Pior Mito do Cliente

- **Mito:** Os requisitos de projeto modificam-se continuamente, mas as mudanças podem ser facilmente acomodadas, porque o software é flexível (Cuidado como o "Já que...") .
- **Realidade:** É verdade que os requisitos de software se modificam, mas o impacto da mudança varia de acordo com o tempo em que ela é introduzida.

Custo de Introdução de Mudanças



Mitos do Profissional: Terminar Mais Cedo

- **Mito:** Assim que escrevemos o programa e o colocarmos em funcionamento, nosso trabalho estará completo.
- **Realidade:** Alguém disse certa vez que "quanto mais cedo se começa a 'escrever o código', mais tempo demora para que se consiga terminá-lo". Os dados da indústria indicam que entre 50 e 70% de todo o esforço gasto num programa serão despendidos depois que ele for entregue pela primeira vez ao cliente.

Mito da Qualidade

- **Mito:** Enquanto não tiver o programa "funcionando", eu não terei realmente nenhuma maneira de avaliar sua qualidade.
- **Realidade:** Um dos mecanismos mais efetivos de garantia de qualidade de software pode ser aplicado desde o começo de um projeto – a revisão técnica formal. As revisões de software são um "filtro da qualidade" que têm sido consideradas mais eficientes do que a realização de testes para a descoberta de defeitos.

Mito do Executável

- **Mito:** A única coisa a ser entregue em um projeto bem-sucedido é o programa funcionando.
- **Realidade:** Um programa funcionando é somente uma parte de uma configuração de software que inclui vários outros elementos. A documentação forma os alicerces para um desenvolvimento bem-sucedido e fornece um guia para a tarefa de manutenção do software.

Configuração de Software

- Plano de Projeto
- Especificação de Requisitos
- Desenho Arquitetônico
- Manual do Usuário
- Estruturas de Dados
- Especificação de Teste
- Programa Executável

Atividade de Pesquisa

- Com base na exposição feita em sala sobre tópicos importante da Engenharia de Software, segundo Pressman, e em pesquisas e conclusões de seu grupo, discorra sobre a **mitologia do software**, suas conseqüências e formas de contornar os problemas comuns, atacando suas causas freqüentes.