

Engenharia de Software I

Prof. Levi Rodrigues Munhoz

Engenharia de Software

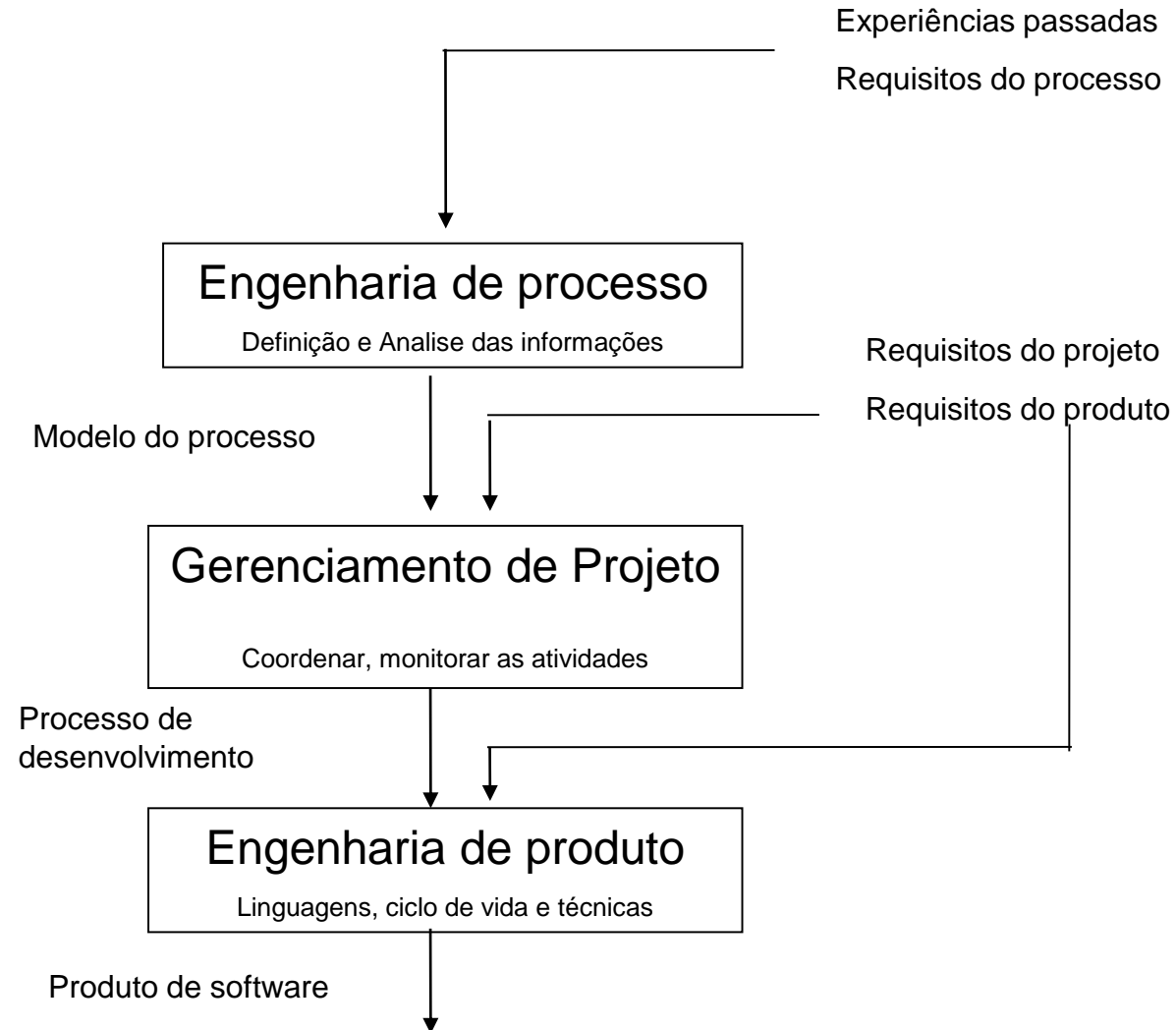
- A crise do software
 - Não eram exigidos requisitos e configurações complexas
- 1968 – Conferência sobre engenharia de software da OTAN. Mais barato planejar do que corrigir
- Princípios de engenharia no desenvolvimento de software (enfoque estruturado e metodológico)

Engenharia de Software

- Estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software, que seja confiável e que funcione eficientemente em máquinas reais".

Friedrich Ludwig Bauer

Engenharia de Software



Engenharia de Software

Camadas de Engenharia de Software



Engenharia de Software

- Qualidade (melhoria contínua do processo)
- Processos (organizam métodos e ferramentas)
- Métodos (indicam como fazer)
- Ferramentas (fornecem apoio informatizado aos métodos)

Engenharia de Software

- Qualidade
 - Qualquer abordagem de engenharia deve estar fundamentada em um comprometimento organizacional com a qualidade. Por exemplo seis sigma. A pedra fundamental que sustenta a engenharia de software é a qualidade.

Engenharia de Software

- Processos
 - Serie de passos previsíveis – roteiro que ajuda a criar um resultado de alta qualidade e dentro de um prazo estabelecido.

Engenharia de Software

- Métodos
 - Fornecem as informações técnicas para desenvolver software. Os métodos envolvem, uma ampla variedade de tarefas que incluem: comunicação, análise de requisitos, modelagem de projetos, construção de programas, testes e suporte.

Engenharia de Software

- Ferramentas
 - Fornecem suporte automatizado ou semiautomatizado para o processo e os métodos. Quando integradas são chamadas de CASE.

Engenharia de Software

- Processo de software
 - Metodologia de processos
 - Uma metodologia (framework) de processo estabelece o alicerce para um processo de engenharia de software completo por meio da identificação de um pequeno número de atividades metodológicas aplicáveis a todos os projetos de software. Independentemente de tamanho ou complexidade.

Engenharia de Software

- Processo de software
 - Atividades genéricas
 - Comunicação – Entender os objetivos e obter requisitos
 - Planejamento – Tarefas técnicas a serem conduzidas, riscos, cronogramas, recursos, produtos resultantes
 - Modelagem – Representação gráfica para facilitar entendimento
 - Construção – Codificação e testes
 - Entrega – Entrega completa ou de um incremento

04/08 Engenharia de Software

- Processo de software
 - Atividades de apoio
 - Controle e acompanhamento – Revisar planos
 - Administração de riscos – Avalia riscos que podem afetar o resultado ou qualidade do projeto ou produto
 - Garantia da qualidade de Software
 - Revisões técnicas – Codificação e testes
 - Medição

Engenharia de Software

- Processo de software
 - Atividades de apoio
 - Gerenciamento de configuração de software
 - Gerenciamento da capacidade de reutilização
 - Preparo e produção de artefatos de software

Engenharia de Software

- Processo de software
 - Adaptação do processo
 - Um processo de engenharia de software não é rígido nem deve ser seguido à risca. Deve ser ágil e adaptável (ao problema, ao projeto, à equipe e à cultura organizacional)

06/08nEngenharia de Software

- A prática da engenharia de software
 - A essência da prática (como aplicar as atividades de uma metodologia)
 - Compreender o problema (Comunicação e análise)
 - Planejar uma solução (modelagem e projeto de software)
 - Executar o plano (geração de código_
 - Examinar o resultado para ter precisão (testes e garantia de qualidade)

Engenharia de Software

- Compreender o problema

Cuidado para não pensar que entendeu o problema com uma conversa de meia hora.

- Quem tem interesse na solução do problema?
- É possível dividir o problema?
- Pode ser representado de forma gráfica?_

Engenharia de Software

- Planeje a solução

Agora que você entende o problema (ou acha), não vê a hora de começar programar. Calma, antes disto faça um projeto

- Você já viu problemas semelhantes anteriormente?
- Algum problema semelhante foi resolvido? Há como reusar partes?
- É possível definir subproblemas? Se sim, há soluções para isto?
- É possível criar um modelo?

Engenharia de Software

- Execute o plano

Pode ser que surjam desvios inesperados, o plano permitirá que continuemos sem nos perder

- A solução é adequada ao plano?
- As partes componentes estão corretas?

Engenharia de Software

- Examine o resultado

Mesmo não garantindo uma solução perfeita, devemos assegurar um número de testes suficientes para revelar o maior número possível de erros.

- É possível testar cada parte componente da solução? Foi implementada uma estratégia de testes razoável?
- O software atende aos requisitos? Houve aceitação/validação dos interessados?

Engenharia de Software

- A prática da engenharia de software
 - Princípios gerais

Os princípios ajudam a estabelecer um modo de pensar para a prática segura da engenharia de software. David Hooker propôs sete princípios que se concentram na prática da engenharia de software como um todo, a saber:

Engenharia de Software

- Princípios:

- 1 – A razão de existir:

Pergunte a si mesmo, “Isso realmente agrega valor ao sistema?”
Se a resposta for não, não faça.

Engenharia de Software

- Princípios:
 - 2 – KISS (Keep It Simple, Stupid), ou seja, não complique
Cuidado, pois o sistema não pode ser simplista, ou seja, não
pode estar incompleto.
Simples não significa gambiarra.
Simplicidade facilita o processo de manutenção.

Engenharia de Software

- Princípios:
 - 3 – Mantenha a visão:

Sem uma visão clara o projeto pode se tornar ambíguo. Pode se transformar em uma colcha de retalhos.

Engenharia de Software

- Princípios:
 - 4 – O que um produz, outros consomem:

Como seu sistema deverá ser construído e ou usado por várias pessoas, construa mantendo uma boa documentação, tanto de sistema quanto de usuário. Facilitando o trabalho de todas as pessoas, você agrega maior valor ao sistema

Engenharia de Software

- Princípios:
 - 5 –Esteja aberto para o futuro:

Sistemas com tempo de vida mais longo e com “qualidade industrial”, devem estar preparados para atender a mudanças e se adaptar ao longo de sua vida. (Isto pode gerar conflito com métodos ágeis)

Engenharia de Software

- Princípios:
 - 6 – Planeje com antecedência, visando reutilização:

Planejar com antecedência para a reutilização reduz o custo e aumenta o valor tanto dos componentes reutilizáveis quanto dos sistemas aos quais eles serão incorporados, mas atenção, reutilização poderá ser cara para aqueles que pretendem desenvolver componentes reutilizáveis.

Engenharia de Software

- Princípios:

- 7 – Pense:

Antes de fazer, pense, pois refazer custa mais caro.

06/08ituEngenharia de Software

Mitos e Realidades no Desenvolvimento de Sistemas

Mitos de gerenciamento

- ❌ Já temos um livro cheio de padrões e procedimentos para desenvolver software. Ele não supriria meu pessoal com tudo que precisam saber.
- 👉 O livro de padrões pode existir, mas ele é utilizado? Os praticantes da área sabem de sua existência? Ele reflete a prática moderna de Engenharia de Software? É completo, é adaptável? Foca a qualidade e o prazo de entrega? Em muitos casos a resposta é não.

Engenharia de Software

Mitos e Realidades no Desenvolvimento de Sistemas

Mitos de gerenciamento

- ❌ Se o cronograma atrasar poderemos acrescentar mais programadores e ficar em dia.
- 👉 O desenvolvimento de software não é um processo mecânico como o de uma fábrica Nas palavras de Brooks acrescentar pessoas em um projeto atrasado, irá atrasá-lo mais ainda. Isto pode parecer absurdo, mas haverá gasto de tempo para treinar e preparar as pessoas recém contratadas para o projeto. Pode-se adicionar pessoas, mas somente de forma planejada e bem coordenada.

Engenharia de Software

Mitos e Realidades no Desenvolvimento de Sistemas

Mitos de gerenciamento

- ❌ Se eu decidir terceirizar o projeto de software, posso simplesmente relaxar e deixar a outra empresa realizá-lo?
- 👉 Se uma empresa não souber gerenciar e controlar projetos de software, ela vai, invariavelmente, enfrentar dificuldades ao terceirizá-las.

Engenharia de Software

Mitos e Realidades no Desenvolvimento de Sistemas

Mitos dos clientes

- ❌ Uma definição geral dos objetivos é suficiente para começar a escrever os programas – podemos preencher os detalhes posteriormente.
- 👉 Embora nem sempre seja possível uma definição ampla e estável dos requisitos uma definição de objetivos ambígua é a receita para um desastre. Requisitos não ambíguos (normalmente derivados iterativamente) são obtidos pela comunicação contínua e eficaz entre clientes e desenvolvedor.

Engenharia de Software

Mitos e Realidades no Desenvolvimento de Sistemas

Mitos dos clientes

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

Engenharia de Software

Mitos e Realidades no Desenvolvimento de Sistemas

- 👉 É verdade que os requisitos de software mudam, mas o impacto da mudança varia dependendo do momento em que foi introduzida. Quando as mudanças dos requisitos são solicitadas cedo (antes de o projeto ou de a codificação terem começado), o impacto sobre os custos é relativamente pequeno. Entretanto, conforme o tempo passa, ele aumenta rapidamente - recursos foram comprometidos, uma estrutura de projeto foi estabelecida e mudar pode causar uma revolução que exija recursos adicionais e modificações fundamentais no projeto.

Engenharia de Software

Mitos e Realidades no Desenvolvimento de Sistemas

Mitos dos profissionais da área

- ❌ Uma vez que o programa foi feito e colocado em uso, nosso trabalho está terminado.
- 👉 Uma vez alguém já disse que "o quanto antes se começar a codificar, mais tempo levará para terminar". Levantamentos indicam que entre 60 e 80% de todo o esforço será despendido após a entrega do software ao cliente pela primeira vez..

Engenharia de Software

Mitos e Realidades no Desenvolvimento de Sistemas

Mitos dos profissionais da área

- ❌ Até que o programa esteja "em execução", não há como avaliar sua qualidade.
- 👉 Um dos mecanismos de garantia da qualidade de software mais eficientes pode ser aplicado a partir da concepção de um projeto - a revisão técnica. Os revisores de software são "filtros de qualidade", considerados mais eficientes do que os testes feitos para encontrar certas classes de defeitos de software

Engenharia de Software

Mitos e Realidades no Desenvolvimento de Sistemas

Mitos dos profissionais da área

- ❌ O único produto passível de entrega é o programa em funcionamento.
- 👉 Um programa funcionando é somente uma parte de uma configuração de software que inclui muitos elementos. Uma variedade de artefatos (por exemplo, modelos, documentos, planos) constitui uma base para uma engenharia bem-sucedida e, mais importante, uma orientação para suporte de software.

Engenharia de Software

Mitos e Realidades no Desenvolvimento de Sistemas

Mitos dos profissionais da área

- ❌ A engenharia de software nos fará criar documentação volumosa e desnecessária e, invariavelmente, vai nos retardar.
- 👉 O objetivo da engenharia de software não é criar documentos. É criar um produto de qualidade. Uma qualidade melhor leva à redução do retrabalho. E retrabalho reduzido resulta em tempos de entrega menores.

Bibliografia

Pressman, Roger S. Engenharia de Software: uma abordagem profissional/Roger S. Pressman, Bruce R. Maxim. 8ª Edição
Porto Alegre: AMGH, 2016