

Processo de Produção de Software - PS22S

2019 - Sem 2

Prof. Rafael Oliveira (UTFPR - campus DV)





Por que aprender sobre processos de desenvolvimento de software?

- Engenharia de Software é uma das atividades mais complexas realizadas pelo homem
- A Engenharia Civil também é uma das atividades mais complexas desenvolvidas pelo homem
- A diferença entre essas duas ciências é que na Eng. Civil é possível ver que algo está sendo construído de modo errado ou inadequado
- Na Engenharia de Software, temos que lidar com o problema de que o software é abstrato/intangível
- Não se pode prever que um software está sendo construído/implementado de modo errado
- Quando considerarmos um software de largar escala, isso pode se tornar ainda pior



Por que aprender sobre processos de desenvolvimento de software?

- Standish Group of Chaos:
 - Em 2015, U\$ 81 bi fora, gastos em projetos cancelados
 - Em 2015, 19% dos projetos de software falharam
 - UM a cada CINCO projetos de software não se consolidam.



Por que aprender sobre processos de desenvolvimento de software?

- Quais os benefícios de estudarmos processos de desenvolvimento de software?
 - Seu time é beneficiado
 - Sua organização (empresa) é beneficiada
 - Seus clientes são beneficiados
 - A sociedade como um todo tem um produto de qualidade à disposição



O que aprenderemos no curso de processos de desenvolvimento de software?

- Aprenderemos como trabalham os times de desenvolvimento
- Aprenderemos o que faz um time/processo serem considerados bons
- Aprenderemos a modelar/adaptar processos
- Aprenderemos alguns metodologias contemporâneas e históricas de desenvolvimento de software



O que aprenderemos no curso de processos de desenvolvimento de software?

- Ciclo de vida do desenvolvimento de software
 - Modelos de ciclo de vida
- Aspectos sobre comunicação efetiva entre clientes e equipe
- Práticas para implementar soluções em projetos
- Prós e Contras de determinados processos
- Modelos tradicionais de desenvolvimento (cascata, V, sashimi, RUP)
- Mindset ágil
- DevOps
- Modelos de qualidade
- Estudos de caso



O que aprenderemos no curso de processos de desenvolvimento de software?

OBJETIVO:

... Conhecer e praticar todas as práticas para a construção de software de qualidade ...



O que aprenderemos no curso de processos de desenvolvimento de software?

“ ... Ao final do curso seremos capazes de comparar processos e identificar prós e contras para cada tipo de projeto/equipe ... ”



Motivação e Introdução

- O software rapidamente se tornou parte integrante da vida humana
- Exatamente como esperamos que um carro funcione, as pessoas (usuários) esperam que os sistemas de software funcionem
- Problemas e mal funcionamentos causam transtornos
- Saber fazer software com qualidade é a primeira motivação para aprendermos processos de desenvolvimento de software de modo efetivo.



Motivação e Introdução

Para pensar:

O que se parece com o
processo de desenvolvimento
de software?





Motivação e Introdução

Para pensar:

O que se parece com o
processo de desenvolvimento
de software?

Resp: Construir uma casa?





Motivação e Introdução

Para pensar:

O que se parece com o
processo de desenvolvimento
de software?

Resp: Construir uma casa?

2 quartos, 1 cozinha, 1 banheiro, etc..





Motivação e Introdução

Para pensar:

O que se parece com o
processo de desenvolvimento
de software?

Resp: Construir uma casa?

Voce: 2 quartos, 1 cozinha, 1 banheiro,
etc..

Construtores: ok. As opção são





Motivação e Introdução

Construção de uma casa:

- Falo o que quero;
- Construtora me dá opções
- Eu faço ajustes e temos um acordo
- Construtora prepara um planta
- Mais ajustes são feitos e temos um novo acordo
- Construtora prepara mais detalhes:
 - Alicerce, planta elétrica, planta hidráulica, etc.





Motivação e Introdução

Construção de uma casa:

- A construção é, então, iniciada
- A construtora irá fazer a casa aos poucos
- Você acompanhará a construção
- Mais ajustes poderão ser feitos
- Após construída, a casa é definitivamente sua
- Ao final, você se muda para a casa
- A construtora faz manutenções de tempos em tempos





Motivação e Introdução

Para pensar:

Esse processo é
similar à
implementação de
um software?





Motivação e Introdução

- Vamos checar quão similar são os dois processos ...
- Construindo um software:
 - Analista coleta requisitos (similar)
 - Opções são apresentadas e protótipos são sugeridos
 - O analista tem, então, exatamente o que o cliente precisa
 - O analista apresenta o protótipo para um arquiteto
 - São definidos diferentes componentes para criar o que é requisitado





Motivação e Introdução

- Vamos checar quão similar são os dois processos ...
- Construindo um software:
 - Uma vez que o design é feito, a codificação é iniciada
 - Testes unitários são feitos
 - Integrações são feitas
 - Testes de integração são feitos
 - Existindo uma versão executável do software, testes de aceitação são feitos
 - O usuário dá um feedback:
 - Era isso que eu esperava (ou Não foi isso que pedi)





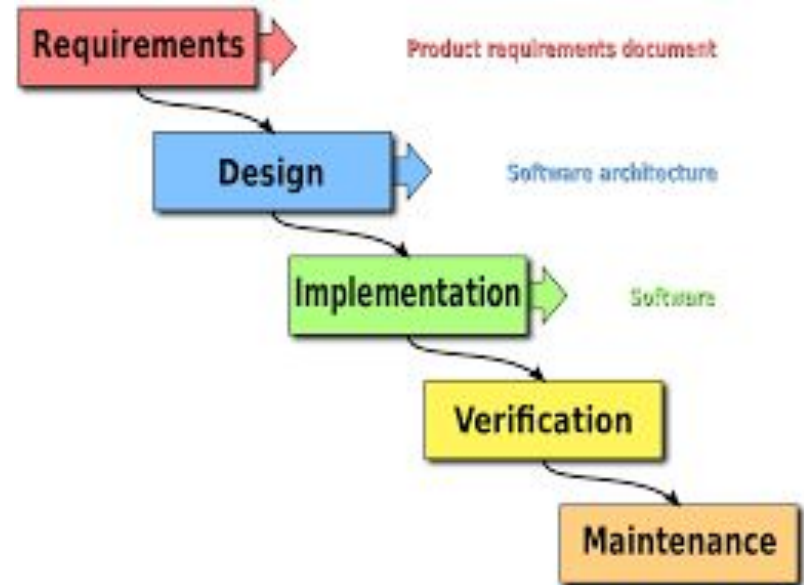
Motivação e Introdução

- Vamos checar quão similar são os dois processos ...
- Construindo um software:
 - Após tudo isso, o software vai para produção (é instalado no ambiente no qual o cliente irá utilizá-lo)
 - Usuário irão começar a utilizar o software e requisitar mudanças, melhorias, ajustes, etc.
 - Defeitos podem ser revelados
 - Estado de operação e manutenção
- O processo usado foi similar ao:
 - CASCATA



Motivação e Introdução

- CASCATA:
 - Quando a indústria começou a utilizar esse método, diversos problemas foram revelados:
 - Dificuldade de prever requisitos futuros
 - Requisitos podem ter sido mal interpretados
 - Problemas de integração podem ter sido escondidos

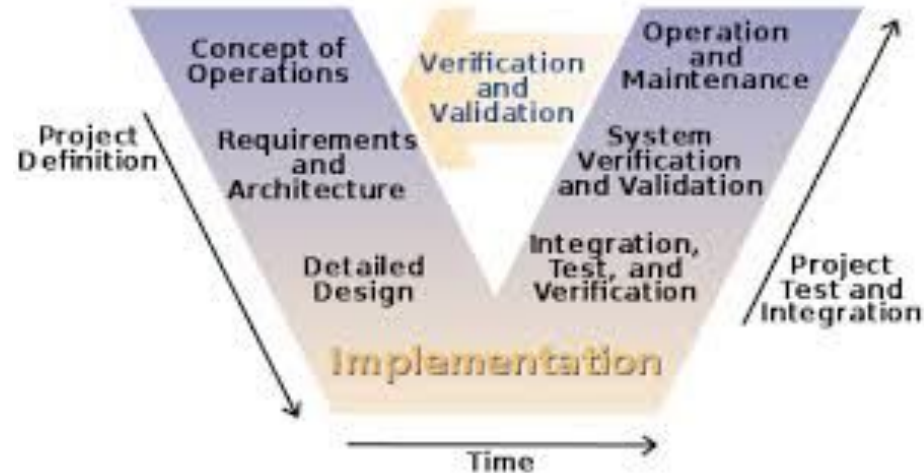




Motivação e Introdução

Devido aos problemas, variações do cascata foram criadas para sanar inconsistências pontuais:

- Modelo V





Motivação e Introdução

Devido aos problemas, variações do cascata foram criadas para sanar inconsistências pontuais:

- Sashimi

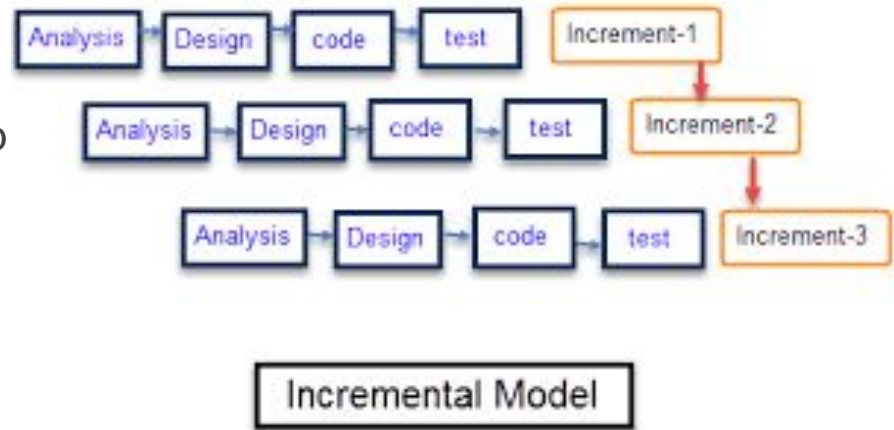




Motivação e Introdução

Devido aos problemas, variações do cascata foram criadas para sanar inconsistências pontuais:

- Incremental

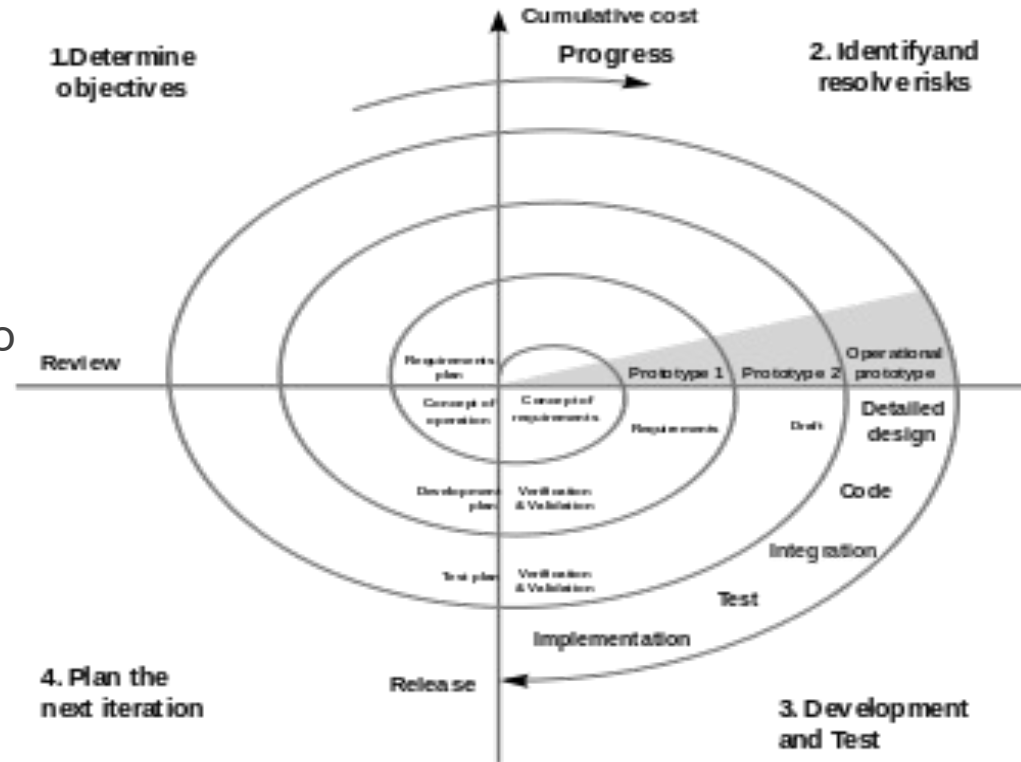




Motivação e Introdução

Devido aos problemas, variações do cascata foram criadas para sanar inconsistências pontuais:

- Espiral





Motivação e Introdução

Enquanto todos os processos eram avaliados e ajustados na indústria, um manifesto surgiu por parte dos principais players da indústria de software:

- Os Ágeis
 - Ágil **não** é um modelo
 - Ágil é um **mindset** (programação mental)





Motivação e Introdução

- Líderes de mercado pensaram junto para responder a seguinte pergunta:
 - O que está nos fazendo obter sucesso em projetos de software?
- A resposta para a pergunta acima foi: Agil (mideset ágil)
- Foi criado, então, o manifesto ágil e princípios ágeis
- Esse manifesto foi ajustado em modelos como, por exemplo,

- Scrum

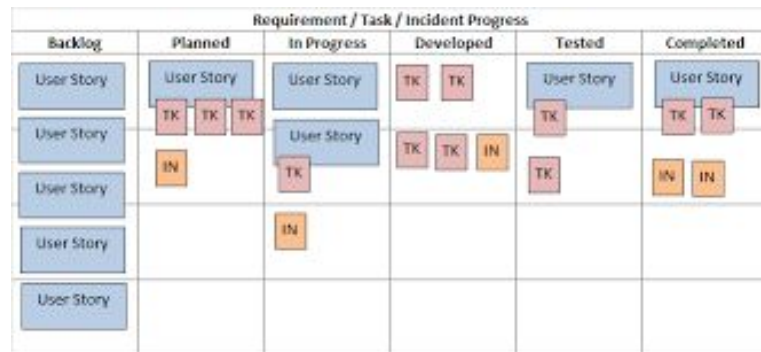




Motivação e Introdução

- Líderes de mercado pensaram junto para responder a seguinte pergunta:
 - O que está nos fazendo obter sucesso em projetos de software?
- A resposta para a pergunta acima foi: Agil (mideset ágil)
- Foi criado, então, o manifesto ágil e princípios ágeis
- Esse manifesto foi ajustado em modelos como, por exemplo,

- **Kanban**



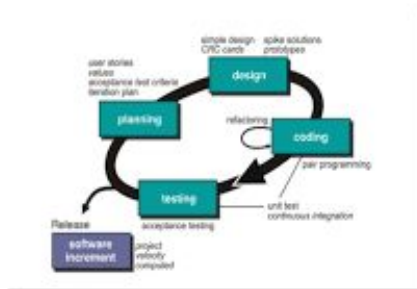


Motivação e Introdução

- Líderes de mercado pensaram junto para responder a seguinte pergunta:
 - O que está nos fazendo obter sucesso em projetos de software?
- A resposta para a pergunta acima foi: Agil (mideset ágil)
- Foi criado, então, o manifesto ágil e princípios ágeis
- Esse manifesto foi ajustado em modelos como, por exemplo,

- XP

Extreme Programming (XP)





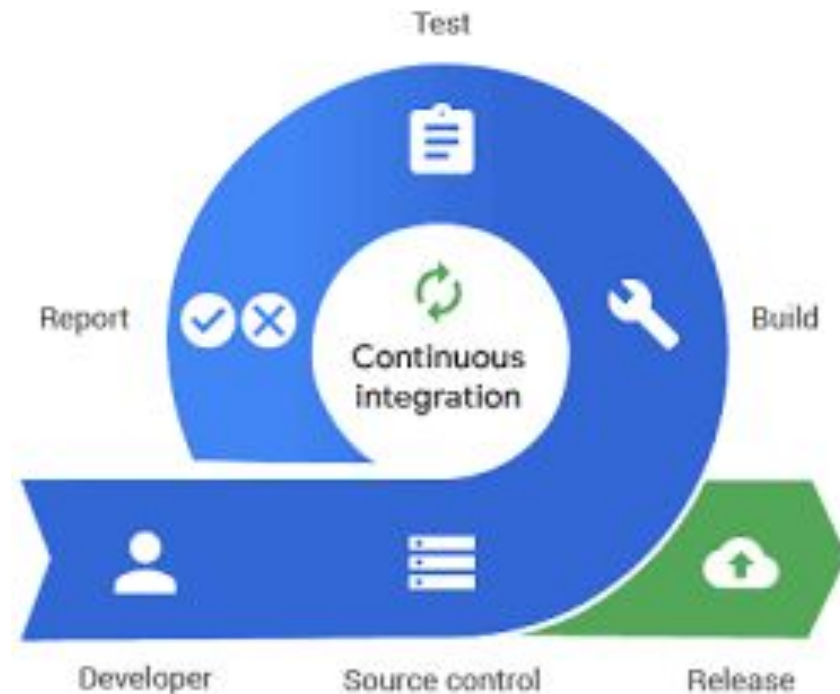
Motivação e Introdução

- Ideia comum dos modelos: construir em ciclos curtos:
 - Projetar um pouco, implementar, teste e aprender sobre o produto.
 - Então, isso é feito diversas vezes seguidas



Motivação e Introdução

- Esse manifesto permitiu:
 - Ajustes rápidos ao mercado
 - Encarar mudanças como uma norma
 - Ajuste de integração contínua
 - Commit
 - Automatizados
 - Build
 - Deploy (implantação)
 -



Motivação e Introdução

- A maturidade dos ágeis tem levado a novas tendências:
 - Confidencialidade entre **desenvolvedores** e **operadores**
 - **Desenvolvedores:** implementam novas features/funcionalidades
 - **Operadores:** estabilizam e deixam o software operante
 - Cultura **DevOps**





Motivação e Introdução

- Ágeis se tornaram popular
- Caso interessantes começaram a surgir
- Ágil tornou-se popular inicialmente em projetos pequenos
- A indústria move em ciclos pequenos
- Devido a isso, conceitos como Lean Startup e Design Thinking têm aparecido
- Tais conceitos tem algo em comum?:
 - Sim: “Identificar a forma mais barata e rápida de aprender e lidar com mudanças”





Case study: dias atuais

[Zappos.com](https://www.zappos.com)

- Criadores queriam saber se conseguiriam vender sapatos/tênis online
- Criaram um website para isso e pensara: Vamos ver se alguém vai comprar
- Caso alguém comprasse um sapato, eles iam à loja, compravam o sapato e enviavam para o comprador.

Zappos!



Case study: dias atuais

[Zappos.com](https://www.zappos.com)

- Fundada em 1999 por Nick Swinmurn
- Comprada pela Amazon em 2009 por U\$1.2 bi

Exemplo:

- Como reagir rápido ao mercado/mudanças
- Como aprender com o mercado/consumidor
- Processos rápidos e eficientes para mudanças
- Foco no resultado





Case study: dias atuais

- Moral de história:
 - Ao modelar processos, devemos:
 - Considerar questões individuais,
 - Considerar motivações
 - Ajustar mindsets
 - Lidar com aspectos humanos, etc.



A indústria de software segue em constante evolução, se tornando a cada dia melhor e mais eficiente.



Dúvidas





Exercício

Pesquise e descreva em um parágrafo sobre a história de alguma startup que focou em resultados e foi bem sucedida como a zappos. (atividade no moodle). Escreva um parágrafo associando mudanças, foco em resultados e processos de desenvolvimento de software.