

Aula 1 Engenharia de Software

Benay Tomas

May 28, 2024

Princípios de Engenharia de software

Formalidade, completeza, ocultação, localização, integridade e abstração

1 Fundamentos da engenharia de software

O primeiro fundamento de engenharia de software são os tipos de software.

Tipos de software

- Sistemas operacionais
- Aplicações/Programas
- Inteligência artificial
- Aplicações móveis
- Embarcados
- Web

Como engenharia de software veio a existir?

Crise de software

Nos anos 70, não havia padronização no desenvolvimento de softwares. Com isso, muitos problemas surgiam, como por exemplo estouro de cronograma, mal otimização, falta de documentação para manutenção de software, entre outros.

Nessa mesma época, alguns mitos surgiram e muitos deles permanecem até hoje.

Modelo de processo

Se trata de um fluxograma de processo para guiar os desenvolvedores por todas as fases do projetos.

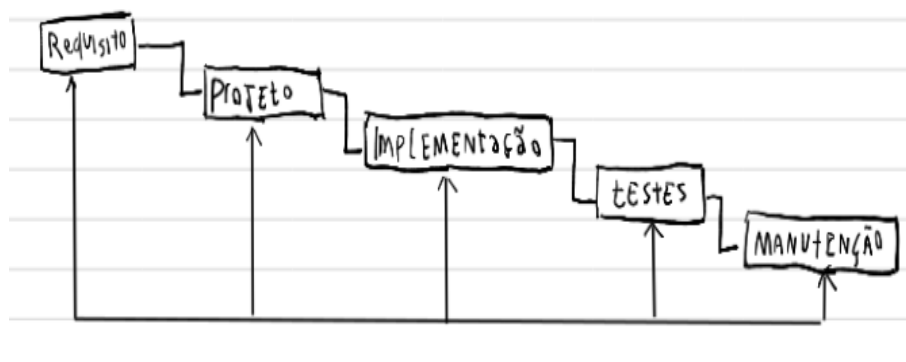
Esses processos são receitas constituídos de métodos e ferramentas.

Processos fundamentais:

Especificação de software – > implantação – > validação – > evolução

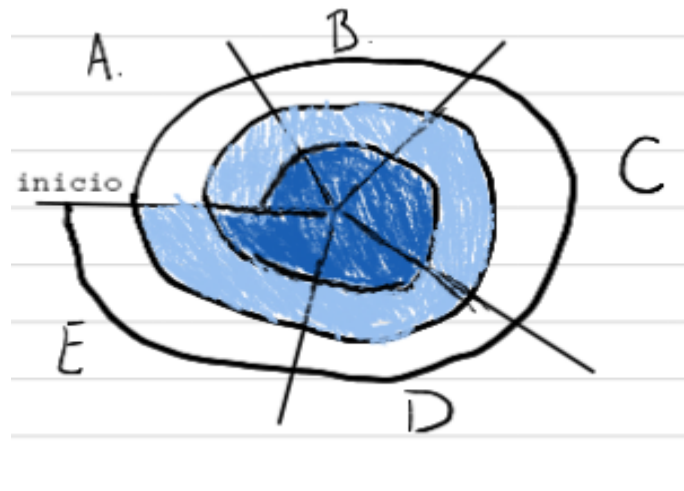
Alguns desses modelos são:

Modelo de cascata



O modelo cascata também é chamado de modelo sequencial, o que significa que ele segue uma sequência de passos e ao final entrega o produto e faz as manutenções necessárias.

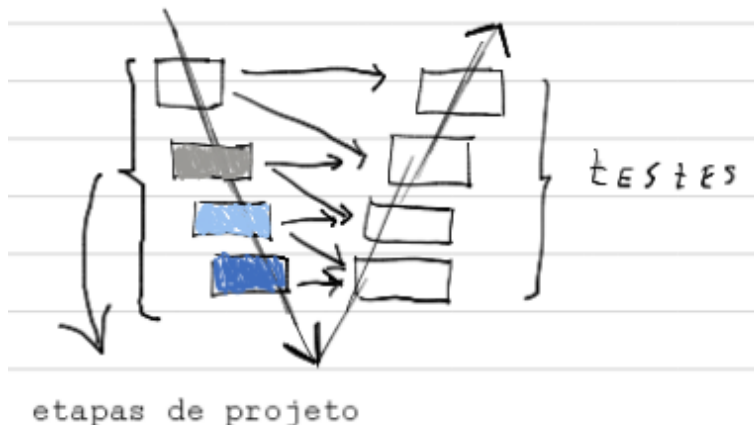
Modelo espiral



A: Comunicação; B: Planejamento; C: Modelagem; D: Construção e teste; E: Entrega

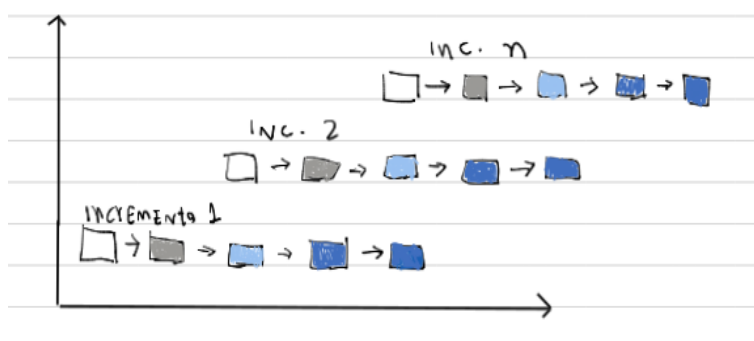
Esse modelo é cíclico, que diferente do sequencial após a entrega do produto, a manutenção executada nada mais é do que repetir as etapas anteriores mudando o que for necessário.

Modelo V



Nesse modelo a cada etapa, são feitos testes.

Modelo incremental



Nesse modelo, a cada uma certa quantidade de etapas são feitos incrementos contínuos a partir daquela etapa.

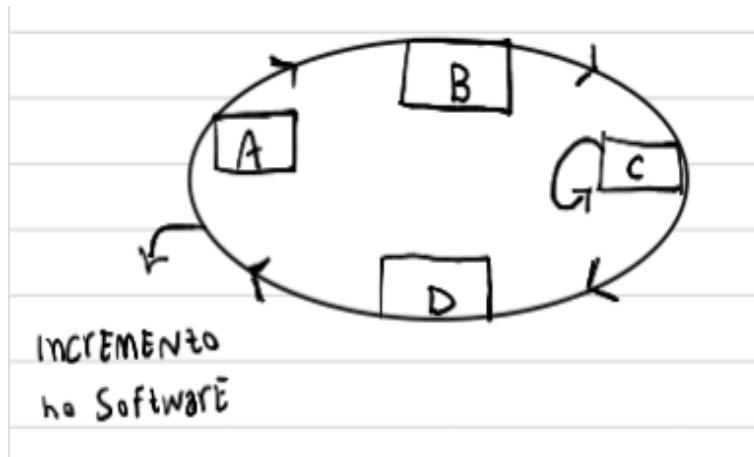
Basicamente funciona como o modelo espiral com a diferença é que são feitos incrementos enquanto fazendo as etapas e não ao final do modelo.

2 Metodologias ágeis

Surgimento em 2000. Menos centrada em documentação. Adaptativas. Orientada a pessoas e não processos. Incremental e interativa.

Extreme Programming (XP)

- Rápido desenvolvimento.
- Requisitos se alteram muito.
- Simplicidade, comunicação e feedback.



A: Planejamento; B: Projeto; C: Codificação; D: Testes

Nessa metodologia a codificação é feita em pares, dois desenvolvedores trabalham no mesmo processo. Além disso, o processo de codificação pode ser repetido múltiplas vezes.

Ao final é feito um incremento de software.

A equipe de trabalho consiste de:

Gerente, coach, analista de testes, redator técnico e desenvolvedor.

SCRUM

Essa é a mais usada hoje em dia. Ela é composta de reuniões regulares (Stand Up Meeting). É feito em ciclos (Sprints).

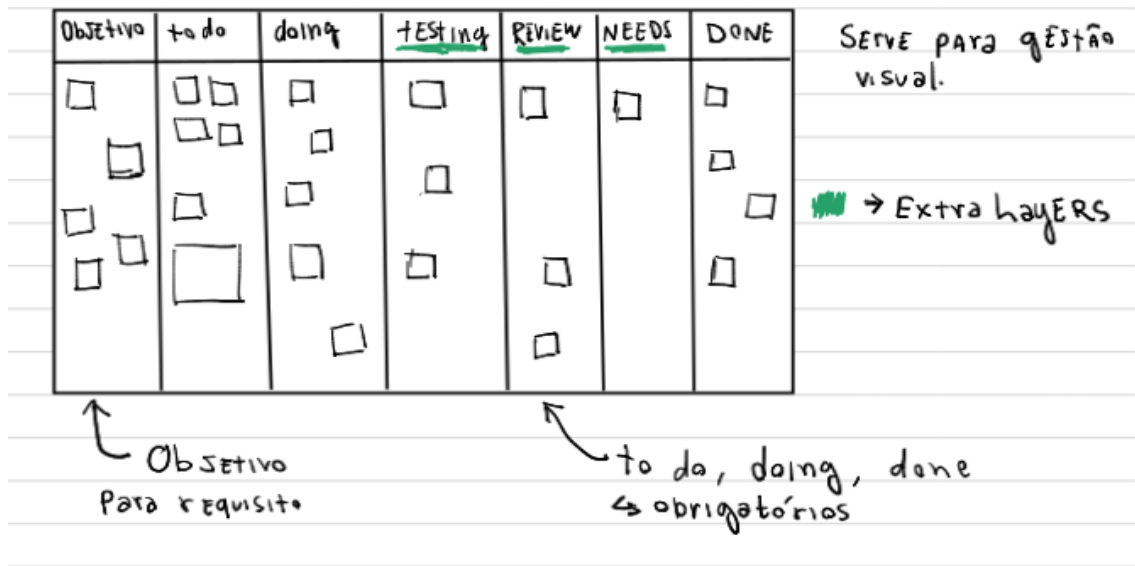
Equipe:

- Scrum Master
- Product Owner
- Scrum Team

Elementos importantes do SCRUM

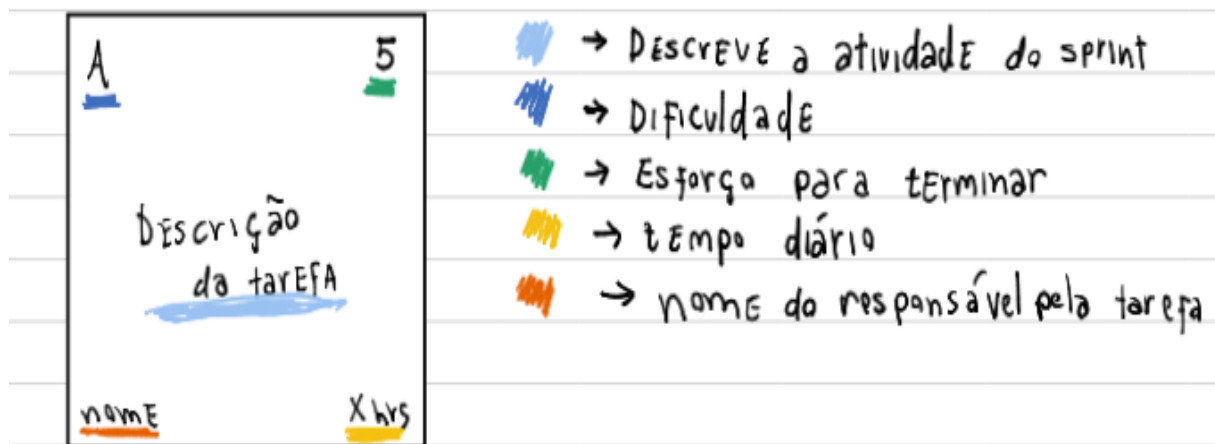
- Product backlog
- Sprint backlog
- Sprints
- Kanban
- Pôquer de planejamento

Product backlogs são os requisitos gerais do projeto. Sprint backlogs são os requisitos apenas das sprints. Sprints são os ciclos do desenvolvimento do projeto. Kanban é um quadro de trabalho para relembrar visualmente os sprints necessários, esse quadro pode ser físico ou virtual (Trello). Pôquer de planejamento é utilizado para apostar em quanto tempo levará para terminar uma sprint.



Post-it para Kanban

Os post-its do kanban são feitos pelas equipe mas existe um padrão para eles também.



Esforço

1. Horas
2. 12 horas
3. Dias
4. Semanas

Dificuldade

- MB - Muito baixa
B - Baixa
A - Alta
MA - Muito alta
S - Sem dificuldade

3 Gestão de configuração

Identificar, organizar e controlar modificações no software.

Definir o que será gerenciado. Estabelecer responsável. Estabelecer políticas de gerenciamento. Especificar ferramentas. Descrever a estrutura do banco de dados.

Etapas

Identificar – > controle de alteração – > controle de versão – > autoria de configuração – > relatos.

Gerenciamento

- Repositórios de códigos
- Baselines (releases)
- Branches (ramificações do código base)

Componentes para controle de versão.

- Código
- Documentação
- Diagramas

Esses componentes servem para manter a rastreabilidade dos itens. Saber onde estão todos os elementos do software. Manter a padronização dos itens.

Gerenciamentos de versões e releases

- > Releases – > São as versões finais alteradas (com novas funcionalidades) entregadas ao cliente.
- > Versões – > Alterações feitas no projeto principal que não são lançadas.

4 Referências

1. Anotações das aulas