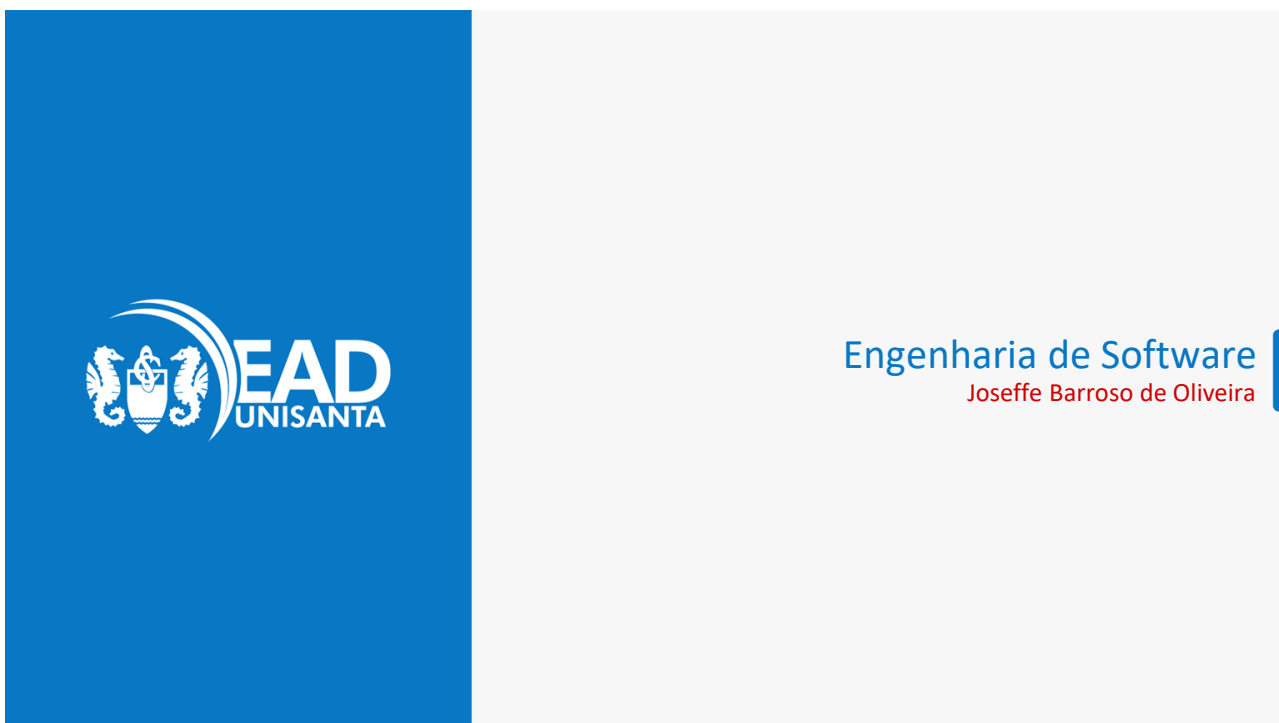




1



2



Introdução

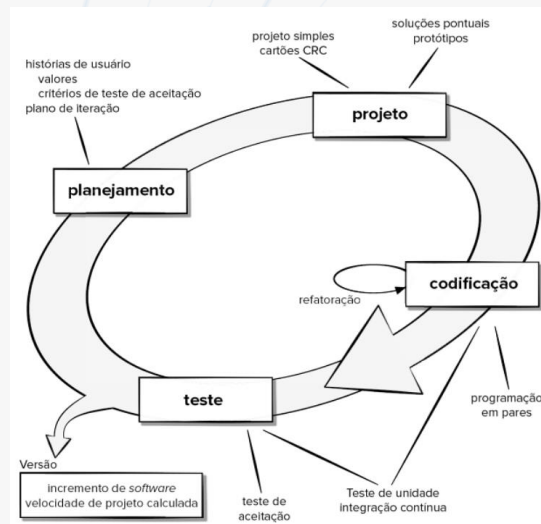
A história da **engenharia de software** é recheada de **metodologias e descrições de processos, métodos e notações de modelagem, ferramentas e tecnologias obsoletas**. Todas atingiram certa notoriedade e foram ofuscadas por algo novo e (supostamente) melhor. Com a introdução de uma ampla variedade de frameworks de processos ágeis – todos disputando aceitação pela comunidade de desenvolvimento de software –, o movimento ágil seguiu o mesmo caminho histórico.

Conforme citado na última seção, **o framework ágil mais utilizado de todos é o Scrum**. Porém, muitos outros têm sido propostos e encontram-se em uso no setor. Nesta seção, apresentamos um breve panorama de três métodos ágeis populares: Extreme Programming (XP), Kanban e DevOps.



O framework XP

A Extreme Programming (Programação Extrema) envolve um conjunto de regras e práticas constantes no contexto de **quatro atividades metodológicas: planejamento, projeto, codificação e testes**. A figura ao lado ilustra o processo XP e destaca alguns conceitos e tarefas-chave associados a cada uma das atividades metodológicas.



O framework XP

- **Planejamento:** A atividade de planejamento (também chamada de o jogo do planejamento) se inicia com ouvir. A atividade de ouvir conduz à **criação de um conjunto de “histórias”** (também denominadas histórias de usuário) que descreve o resultado, as características e a funcionalidade solicitados para o software a ser construído.
- **Projeto.** O projeto XP segue rigorosamente o princípio KISS (keep it simple, stupid!, ou seja, não complique!). O projeto de funcionalidade extra é desestimulado. A XP estimula o **uso de cartões CRC** (classe-responsabilidade-colaborador) como um mecanismo eficaz para pensar o software em um contexto orientado a objetos. Os **cartões CRC identificam e organizam as classes orientadas a objetos** relevantes para o incremento de software corrente.

O framework XP

- **Codificação.** Depois de desenvolvidas as histórias e de o trabalho preliminar de elaboração do projeto ter sido feito, a equipe não passa para a codificação, mas desenvolve uma série de testes de unidades que exercitarão cada uma das histórias a ser incluída na versão corrente (incremento de software). Um conceito-chave na atividade de codificação (e um dos mais discutidos aspectos da XP) é a **programação em pares**. A XP recomenda que **duas pessoas trabalhem juntas** em uma mesma estação de trabalho.
- **Testes.** Os testes de unidades criados devem ser implementados usando-se uma metodologia que **os capacite a ser automatizados** (assim, poderão ser executados fácil e repetidamente). Isso estimula uma estratégia de testes de regressão toda vez que o código for modificado.



7

Kanban

O método Kanban é **uma metodologia enxuta que descreve métodos para melhorar qualquer processo ou fluxo de trabalho**. O Kanban enfoca a **gestão de alterações e a entrega de serviços**. A gestão de alterações define o processo por meio do qual uma **alteração solicitada** é integrada a um sistema baseado em software. A entrega de serviços é incentivada com o **foco no entendimento das necessidades** e expectativas do cliente.

O Kanban nasceu na Toyota, onde era uma série de práticas de engenharia industrial, e foi adaptado ao desenvolvimento de software por David Anderson. O Kanban em si depende de seis práticas fundamentais.



8

Kanban

1. **Visualizar** o fluxo de trabalho utilizando um quadro Kanban;
2. **Limitar** a quantidade de estoque em processo (WIP, do inglês work in progress) em um dado momento;
3. **Gerenciar o fluxo** de trabalho reduz os desperdícios por meio do **entendimento** do fluxo de valor atual;
4. Explicitar as **políticas de processo** (motivos pelos quais um determinado **item foi selecionado**);
5. Enfocar a **melhoria contínua** com a criação de ciclos de **feedback**;
6. **Alterar o processo** colaborativamente e **engajar todos** os membros de equipe e outros envolvidos.



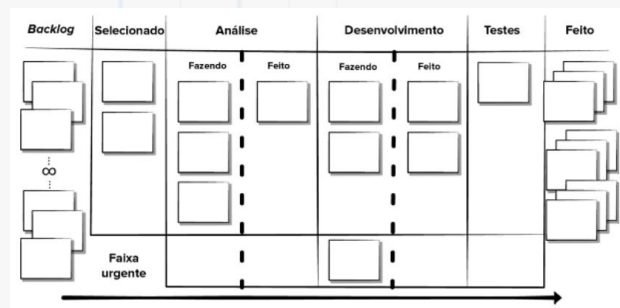
9

Kanban

As reuniões de equipe para o Kanban **são semelhantes** àquelas realizadas na metodologia Scrum.

O objetivo é **avancar primeiro os itens com alto valor de negócio**. A equipe observa o fluxo e tenta identificar obstáculos à finalização com uma análise da carga de trabalho e dos riscos.

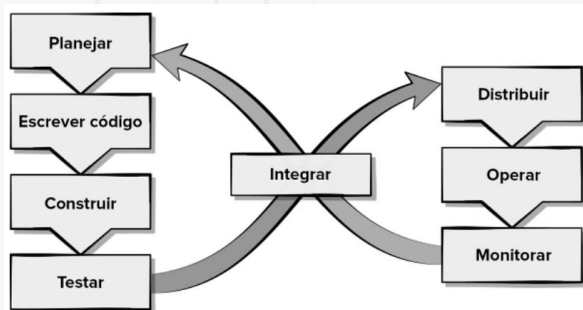
O Kanban pode ser **combinado facilmente com outras práticas** de desenvolvimento ágil para adicionar um pouco mais de disciplina ao processo.



10

DevOps

O DevOps foi criado para combinar Desenvolvimento (Development) e Operações (Operations). O DevOps tenta aplicar os princípios do desenvolvimento ágil a toda a cadeia logística de software. A figura ao lado apresenta um panorama do fluxo de trabalho do DevOps. A abordagem envolve diversas etapas que formam ciclos contínuos até que o produto desejado exista de fato.



DevOps

- **Desenvolvimento contínuo.** Os produtos de software são divididos e desenvolvidos em múltiplos sprints, com os incrementos entregues para os membros da equipe de testes e da equipe de desenvolvimento para serem testados.
- **Teste contínuo.** Ferramentas de teste automatizadas são utilizadas para ajudar os membros da equipe a testar múltiplos incrementos de código ao mesmo tempo para garantir que não há defeitos antes da integração.
- **Integração contínua.** Os elementos de código com a nova funcionalidade são adicionados ao código existente e ao ambiente de execução (run-time) e, então, examinados para garantir que não há erros após a

DevOps

- **Entrega contínua.** Nesta etapa, o código integrado é entregue (instalado) ao ambiente de produção, que pode incluir múltiplos locais em nível global, que, por sua vez, precisam ser preparados para receber a nova funcionalidade.
- **Monitoramento contínuo.** Os membros da equipe de operações que pertencem à equipe de desenvolvimento ajudam a melhorar a qualidade do software, monitorando o seu desempenho no ambiente de produção e buscando proativamente possíveis problemas antes que os usuários os identifiquem.