Arquitetura e projeto de software ágil

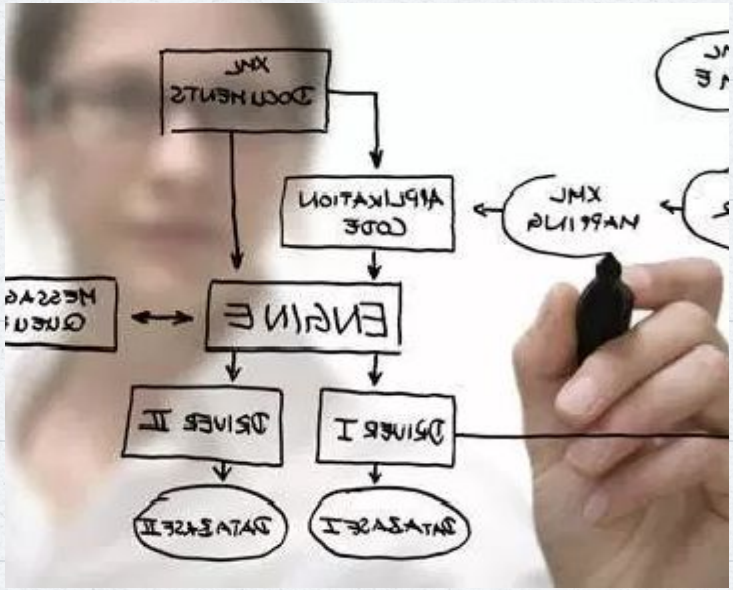
EQUIPE

- Felipe Gomes da Silva
- Isac Maximo Alves de Moura
- Felipe César de Sousa Silva
- Pedro Lucas Luna Araujo Vieira
- Milena Bezerra da Fonseca
- Adriano Mendes

Explicando o que é arquitetura e projeto de software ágil

Arquitetura de software

É o planejamento de um sistema. Isso inclui a organização de todos os componentes, suas interações, tecnologias a serem utilizadas, ambientes de operação, e outras diretrizes usadas para construir o software.



Arquitetura de software ágil

Arquitetura Ágil é um conjunto de valores, práticas e colaborações que apoiam ativamente o projeto e a arquitetura evolutiva de um sistema. Essa abordagem adota a mentalidade DevOps, permitindo que a arquitetura de um sistema evolua continuamente ao longo do tempo, ao mesmo tempo, em que oferece suporte às necessidades dos usuários atuais.

Valores e princípios do manifesto ágil

Como se aplicam a arquitetura e projeto de software ágil

O que é Manifesto Ágil?

O Manifesto Ágil é um dos princípios fundamentais para o desenvolvimento de software. Criado em 2001, surgiu após um grupo de profissionais que utilizam diversas ferramentas da área de tecnologia, porém com objetivos e propósitos de trabalho similares: a inovação.

Valores do Manifesto Ágil

1- “Indivíduos e interações mais que processos e ferramentas”

2- “Software em funcionamento mais que documentação abrangente”

3- “Colaboração com o cliente mais que negociação de contratos”

4- “Responder a mudanças mais que seguir um plano”

Princípios do Manifesto Ágil

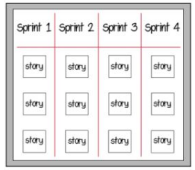
1 Satisfy the **customer**

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2 Welcome **change**

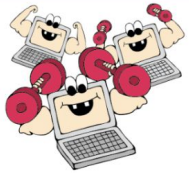
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3 Deliver **frequently**



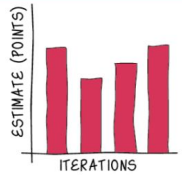
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

7 Working **software**



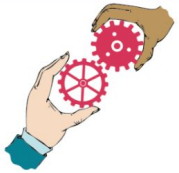
Working software is the primary measure of progress.

8 Sustainable **development**



Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9 Continuous **attention**



Continuous attention to technical excellence and good design enhances agility.

4 Work **together**

Business people and developers must work together daily throughout the project.

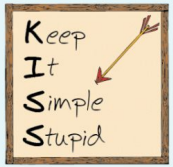
5 Trust and **support**

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6 Face-to-face **conversation**

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

10 Maintain **simplicity**



The art of maximizing the amount of work not done - is essential.

Copyright © 2018 Knowledge Train Limited

11 Self-organizing **teams**

The best architectures, requirements, and designs emerge from self-organizing teams.

12 Reflect and **adjust**

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Explicando o framework scrum

Cerimônias, papéis e artefatos

O QUE É SCRUM ?

Scrum é um framework com o qual as pessoas podem lidar com problemas adaptativos complexos, ao mesmo tempo que entregam produtos com o mais alto valor possível de forma produtiva e criativa.

Framework Scrum

Kanban no desenvolvimento ágil de software

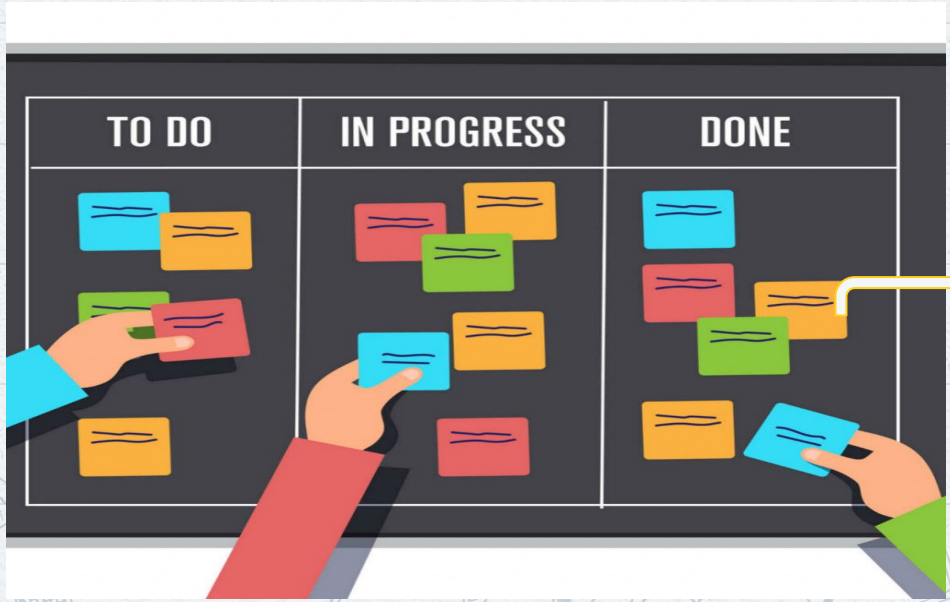
O que é? Como funciona?

"Kanban: palavra japonesa e seu significado literal é cartão ou sinalização"

"Busca melhorar o desempenho e reduzir o desperdicio, eliminando atividades que não geram valor para equipe"

Finalidade do kanban

- Controlar o progresso das tarefas de forma visual



Post it

4 PRINCIPIOS

- Comece com o que você faz agora
- Concordar e buscar mudanças evolucionárias
- Respeitar os papéis e cargos atuais
- Incentivar atos de liderança em todos os niveis

QUADRO

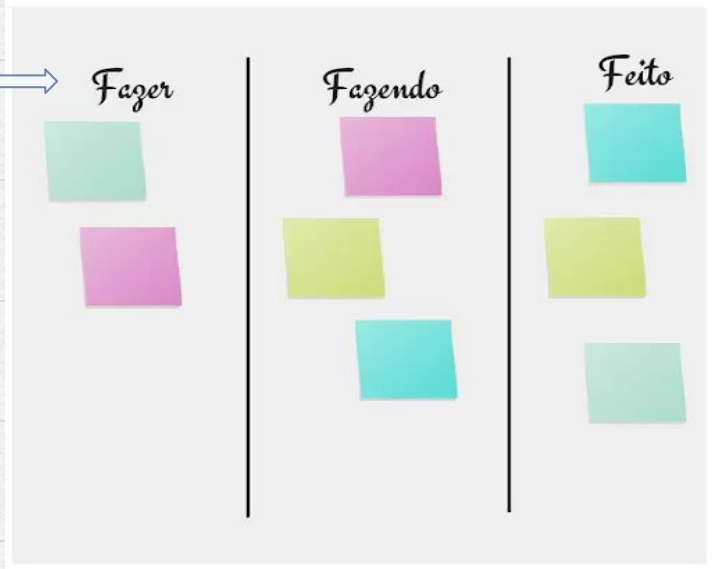
- O trabalho das equipes giram em torno de um quadro Kanban
- Fisico ou virtual

Rastreabilidade, colaboração mais fácil e acessibilidade em vários locais

Função do quadro

- Visualização
- Padronização
- Resolução

Não é regra



Cartões Kanban

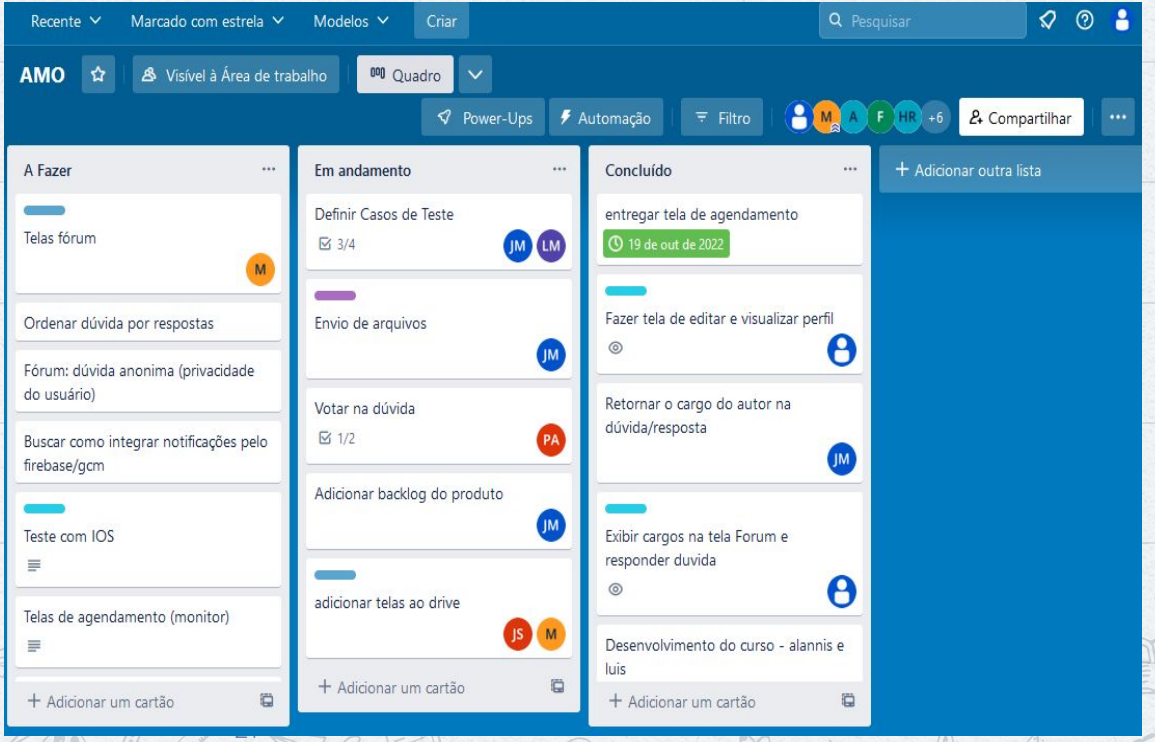
- Para as equipes, cada item de trabalho é representado como um cartão separado no quadro
- Responsável, tempo, descrição

BENEFICIOS

- Redução de desperdícios
- Redução de custos
- Requer menor organização
- Processo simplificado
- Maior visibilidade
- Melhora a motivação e desempenho

FERRAMENTAS

- KanbanFlow
- KanbaNize
- Trello



"O Kanban é uma metodologia que proporciona uma mudança cultural profunda na forma como as equipes trabalham juntas para alcançar objetivos comuns."

- David J. Anderson, criador do Kanban Moderno.

Arquitetura Ágil

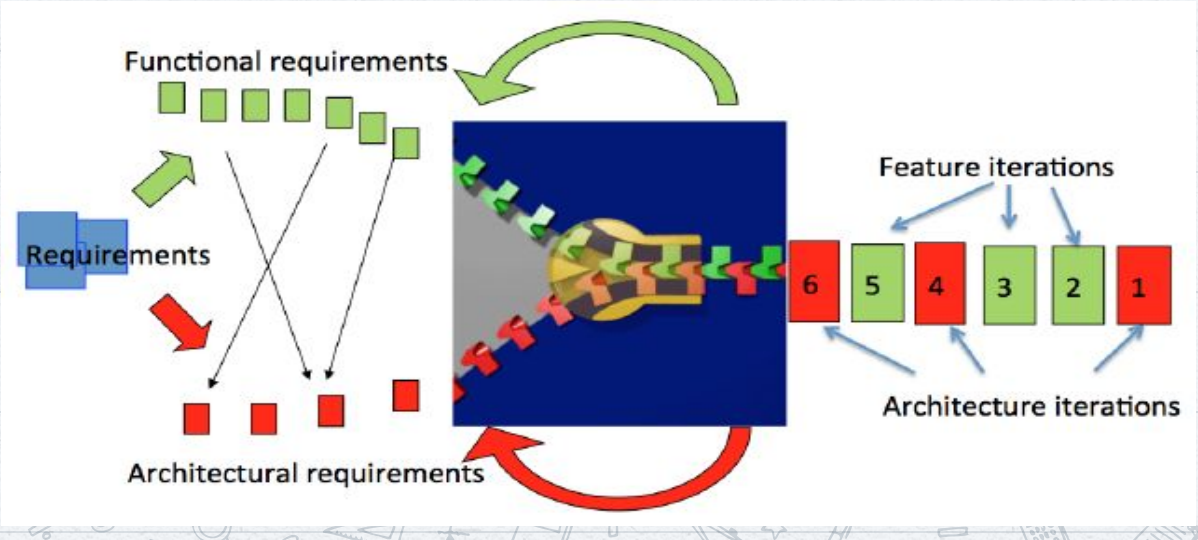
O que é? Vantagens e desvantagens.

O QUE É ARQUITETURA ÁGIL ?

- ✘ Flexibilidade
- ✘ Adaptabilidade
- ✘ Colaboração entre equipes multidisciplinares
- ✘ Evolução do Manifesto Ágil
- ✘ Atender as necessidades do usuário de forma incremental

ARQUITETURA ÁGIL E O ZÍPER

- ✘ Requisitos Funcionais e Requisitos de Arquitetura
- ✘ Alinhamento



Vantagens

- ✘ Flexibilidade e Adaptabilidade
- ✘ Entrega incremental de valor
- ✘ Identificação precoce de problemas

Desvantagens

- ✘ Risco de falta de planejamento
- ✘ Complexidade da gestão de dependências
- ✘ Risco de falta de documentação adequada

DevOps

Explicando a filosofia DevOps, incluindo a automação, integração contínua (CI) e entrega contínua (CD)

Definição

DevOps é uma metodologia de desenvolvimento de software ágil que busca promover a colaboração entre as equipes de desenvolvimento e operações para entregar software de alta qualidade de forma rápida e confiável.

Integração Contínua (CI) e Entrega Contínua (CD)

A integração contínua (CI) é um processo que envolve a automação da compilação, testes e implantação de código para um ambiente de teste. Isso ajuda a detectar problemas mais cedo no processo de desenvolvimento e a acelerar a entrega do software.

A entrega contínua (CD) é uma extensão da CI e envolve a automação do processo de implantação de código em produção. Isso permite que as alterações sejam entregues com mais frequência e de forma mais confiável.

Testes Ágeis

Mas afinal,o que seria testes ágeis?

Os testes ágeis são uma forma de validar as funcionalidades do software através da perspectiva dos clientes, facilitando assim a correção de bugs e a aplicação de melhorias.

Os tipos de testes?

Testes unitários, automatizados e exploratório.

Gerenciamento de requisitos ágeis

Como abordar de forma ágil, incluindo o backlog do produto e o backlog do sprint

Definição

O gerenciamento de requisitos ágeis é a prática de gerenciar os requisitos de um projeto de forma colaborativa e iterativa, para garantir que os requisitos do cliente sejam atendidos de forma eficaz e eficiente.

Backlog do Produto e do Sprint

O Backlog do Produto é uma lista de requisitos priorizados que precisam ser atendidos para que o projeto seja concluído. O backlog do produto é uma ferramenta fundamental no gerenciamento de requisitos ágeis, pois ajuda a garantir que a equipe de desenvolvimento esteja trabalhando nas funcionalidades mais prioritárias.

O Backlog do Sprint é uma lista de itens selecionados a partir do Backlog do Produto que a equipe se compromete a entregar em um Sprint. O Sprint é um período fixo de tempo, geralmente entre uma e quatro semanas, durante o qual a equipe trabalha para entregar um conjunto de itens do Backlog do Sprint.

Metricas Ágeis

- ✘ Velocidade da equipe
- ✘ Burndown chart
- ✘ Retrospectiva

Velocidade da equipe

- ✘ O que é velocidade da equipe e porque ela é importante.
- ✘ Como calcular a velocidade da equipe.
- ✘ Como a mesma pode ser usada para prever o tempo de entrega do projeto.

Burndown chart

- ✘ O que é o burndown chart.
- ✘ Como ele é utilizado.
- ✘ Como o burndown pode ser utilizado para prever o tempo de entrega do projeto.

Retrospectiva

- ✘ O que é retrospectiva.
- ✘ Porque ela é importante.
- ✘ Como a retrospectiva pode ser utilizada para melhorar o processo de desenvolvimento.

Exemplos de sucesso de empresas que adotaram a abordagem ágil em seus projetos de software

Exemplos

Spotify: Adotou a metodologia ágil para melhorar a qualidade do seu software e acelerar o tempo de lançamento. A equipe de desenvolvimento trabalha em sprints e usa uma variedade de práticas ágeis, como o Scrum e o Kanban, para gerenciar seus projetos.

Google: Adotou a metodologia ágil para melhorar a eficiência e a qualidade de seus projetos de software. A equipe de desenvolvimento da Google trabalha em sprints e usa uma variedade de práticas ágeis, como o Scrum e o Lean, para gerenciar seus projetos.

Referências:

Arquitetura Ágil:

- AGILITY AND ARCHITECTURE The phrase “Agile architecture” evokes two concepts. [s.l: s.n.]. Disponível em: <https://resources.sei.cmu.edu/asset_files/Article/2014_101_001_493902.pdf>..
- Ambler, S. (2002) Agile Modeling: Effective Practices for Extreme Programming and the Unified Process. John Wiley & Sons, Hoboken.
- KRUCHTEN, P. Agility and Architecture Agility and Architecture –A Clash of Two Cultures? [s.l: s.n.]. Disponível em: <https://www.ieee.li/pdf/viewgraphs/agility_and_architecture.pdf>..
- BABAR, M.; BROWN, A.; MISTRIK, I. Agile Software Architecture. [s.l: s.n.].
- LARMAN, C. Agile and iterative development : a manager’s guide. Boston: Addison-Wesley, Cop, 2012.

Referências:

Kanban:

- https://www.youtube.com/watch?v=WjZBnYa58B4
- https://www.alura.com.br/artigos/metodo-kanban

Referências:

DevOps:

- Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. IEEE Software, 33(3), 94-100. doi:10.1109/ms.2016.68
- "DevOps: A Software Architect's Perspective" de Len Bass e Ingo Weber
- "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" de Jez Humble e David Farley.
- S. Mysari and V. Bejgam, "Continuous Integration and Continuous Deployment Pipeline Automation Using Jenkins Ansible," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1-4, doi: 10.1109/ic-ETITE47903.2020.239.

Referências:

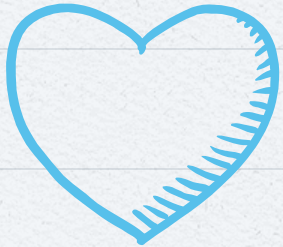
Gerenciamento de requisitos ágeis:

- L. Cao and B. Ramesh, "Agile Requirements Engineering Practices: An Empirical Study," in IEEE Software, vol. 25, no. 1, pp. 60-67, Jan.-Feb. 2008, doi: 10.1109/MS.2008.1.
- Aqsa Rasheed, Bushra Zafar, Tehmina Shehryar, Naila Aiman Aslam, Muhammad Sajid, Nouman Ali, Saadat Hanif Dar, Samina Khalid, "Requirement Engineering Challenges in Agile Software Development", Mathematical Problems in Engineering, vol. 2021, Article ID 6696695, 18 pages, 2021. https://doi.org/10.1155/2021/6696695

Referências:

Exemplos de sucesso de empresas que adotaram a abordagem ágil em seus projetos de software:

- "Scaling Agile @ Spotify" de H. Kniberg e A. Ivarsson.
- "How Google Tests Software" de J. Whittaker.



THANKS!

Perguntas?