

The background is a gradient from dark purple at the top to deep blue at the bottom, speckled with white dots resembling stars. Overlaid on this are several faint, light-colored circular patterns. Some are solid lines, while others are dashed. Some circles have arrows indicating a clockwise direction. One large circle on the left has a scale with numbers: 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260.

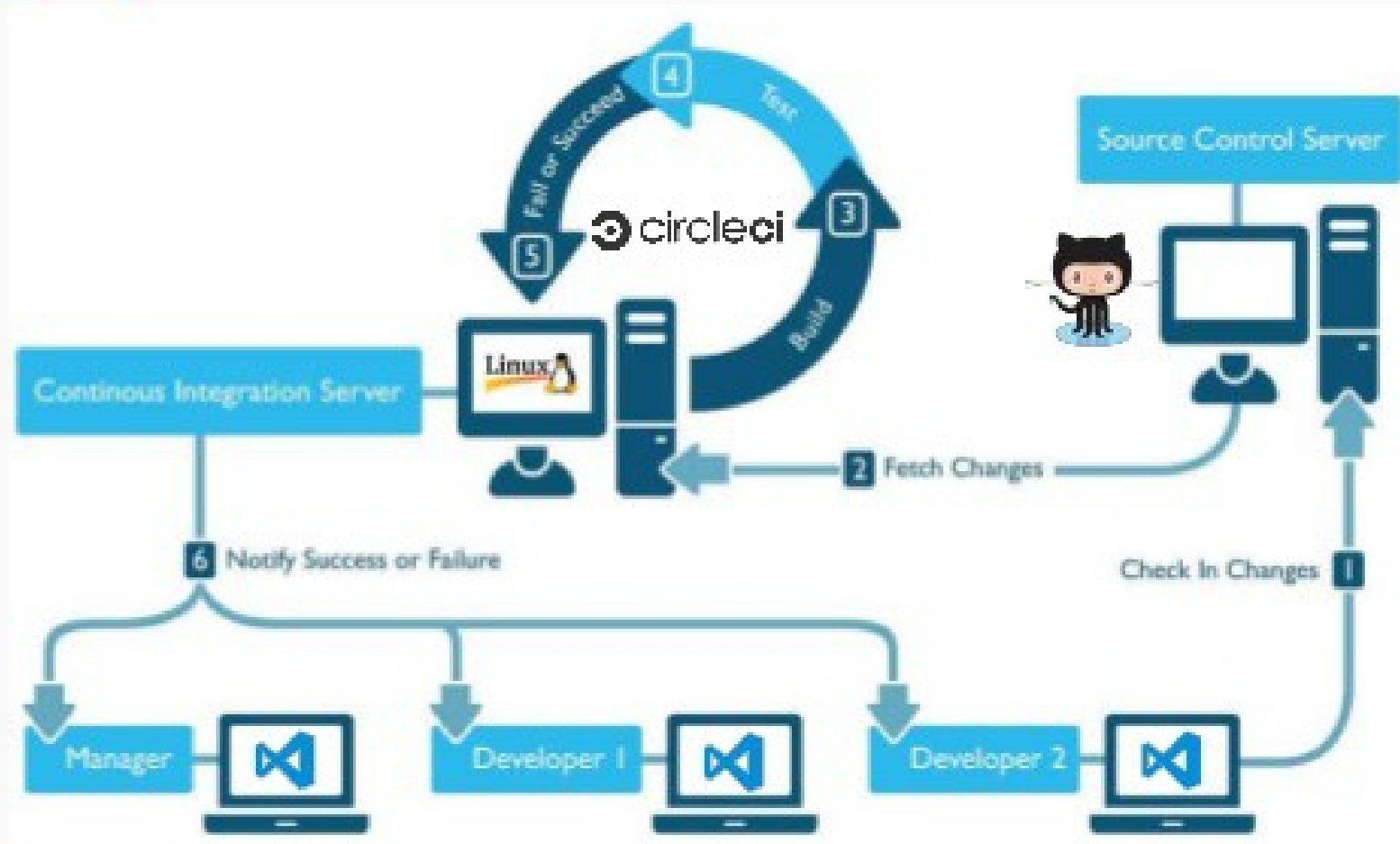
Methods

Metodologia

Integração contínua:

- **Conceito**
- Integração contínua (continuous integration) é um termo originado na metodologia ágil XP que refere a uma prática de desenvolvimento de software de DevOps para que os desenvolvedores, com frequência, juntem suas alterações de código em um Servidor central, que visa tornar a integração de código mais eficiente, através de builds e testes automatizados.
- **Para que?**
- No passado, os desenvolvedores de uma equipe podiam trabalhar isoladamente por um longo período e só juntar suas alterações à ramificação mestre quando concluíssem seu trabalho, que consequentemente poderia havia muito acúmulo de erros sem correção por um longo período, logo objetivo principal é de facilitar e agilizar as junções dos código. Utilizando a integração contínua é possível verificar se as alterações ou novas funcionalidades não criaram novos defeitos no projeto já existente assim podendo lançar atualizações para atender as demandas dos usuários com maior frequência.

Integração Contínua

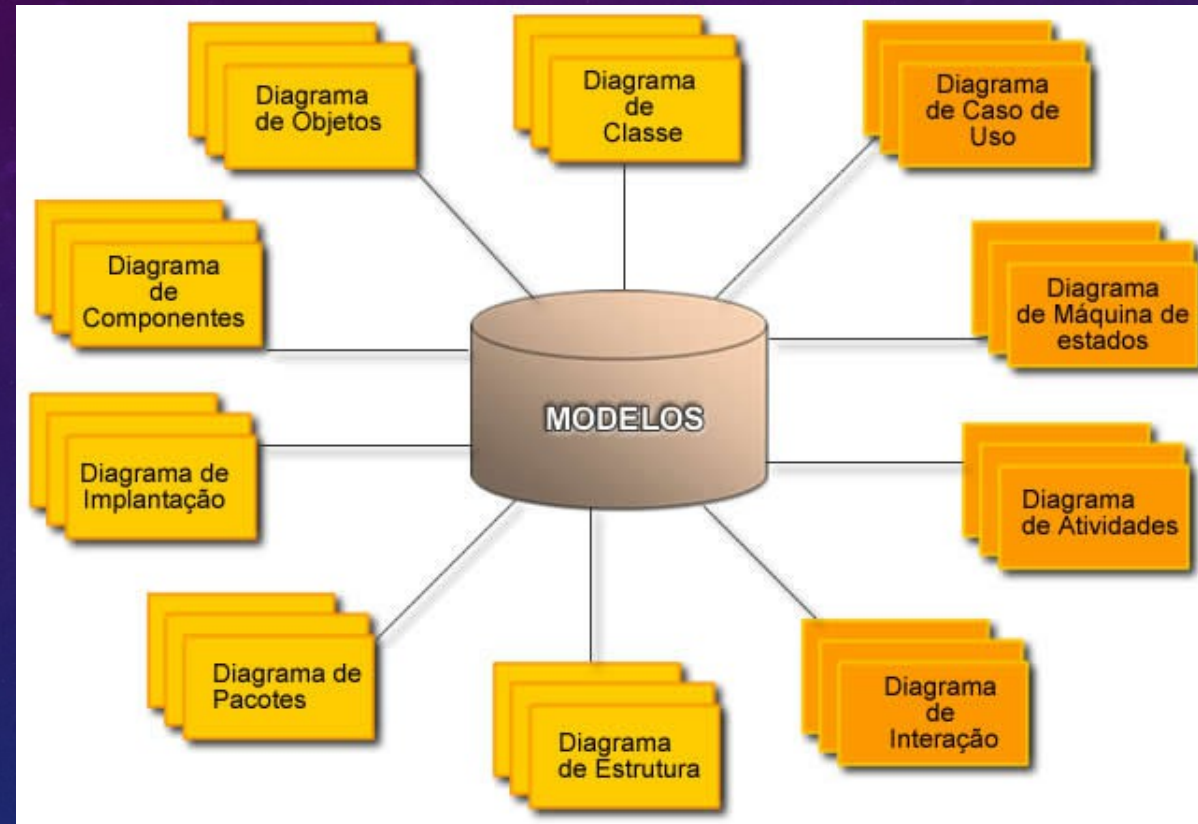


- **Controle de versão**
- Como parte crucial no processo de desenvolvimento e, para o sucesso da integração contínua, o controle de versão deve ser utilizado. Sendo que um dos objetivos do controle de versão é o trabalho colaborativo, em que diversos desenvolvedores trabalham em conjunto e compartilhando dados.
- O sistema de controle de versão resolve um grande problema quando trabalhamos em equipe:
 - como compartilhar as informações de forma a ter a última versão válida e ainda saber quem fez as alterações;
 - como prevenir que os desenvolvedores refaçam o trabalho já desenvolvido, etc.
- Para solucionar estes problemas, existe um conjunto de ferramentas para controle de versão centralizado, entre elas temos o CVS, Subversion, Git, entre outros. Estas ferramentas permitem aos desenvolvedores trabalharem em conjunto, possuindo um Servidor central responsável pelo versionamento do sistema e possibilitando que vários clientes possam acessar, visualizar, modificar e enviar novos códigos se for necessário.

UML - Linguagem de modelagem unificada

- É uma linguagem de notação (um jeito de escrever, ilustrar, comunicar) para uso em projetos de sistemas.
- Esta linguagem é expressa através de diagramas. Cada diagrama é composto por elementos (formas gráficas usadas para os desenhos) que possuem relação entre si.
- São divididos em dois tipos: Diagramas estruturais e diagramas comportamentais.
- Diagramas estruturais: Diagrama de pacotes, Diagrama de implementação, Diagrama de objetos, Diagrama de estrutura composta, Diagrama de componentes, Diagrama de classes.
- Diagramas comportamentais: Diagramas de atividade, Diagrama de máquina de estados, Diagrama de caso de uso, Diagrama de integração.

- Diagramas estruturais devem ser utilizados para especificar detalhes da estrutura do sistema (parte estática), por exemplo: classes, métodos, interfaces, serviços, como componentes devem ser instalados, como deve ser a arquitetura do sistema.
- Diagramas comportamentais devem ser utilizados para especificar detalhes do comportamento do sistema (parte dinâmica), por exemplo: como as funcionalidades devem funcionar, como um processo de negócio deve ser tratado pelo sistema, como componentes estruturais trocam mensagens e como respondem às chamadas.



Arquitetura Orientada a Serviço

- SOA - Arquitetura Orientada a Serviço é um modelo de planejamento de estratégia da área de tecnologia da informação, alinhando diretamente aos objetivos de negócios de uma organização que permite expor as funcionalidades os aplicativos em serviços padronizados e inter-relacionados. Tem como objetivo integrar as aplicações, disponibilizar maior flexibilidade para mudanças, suportar serviços independentes de plataforma e protocolos. Trata os requisitos de baixo acoplamento, desenvolvimento baseado em padrões, computação distribuída independentemente de protocolo, integração de aplicação e sistemas legados, seu componente mais importante é o ESB (Barramento de Serviço Corporativos), não implementa a arquitetura, mas oferece as funcionalidades para implementá-las. Seu maior foco é a construção e disponibilização de serviços de negócio, evitar replicação de dados, reuso e facilidade de manutenção de sistemas, integração entre sistemas, visão e controle do processo de negócio, agilidade na mudanças
- SERVIÇOS- é uma atividade ou conjunto de atividades que são intangíveis, sendo um relacionamento de quem fornece o serviço e de quem consome. Tem características de ser reutilizáveis, baixo acoplamento, capazes de se compor de cada serviço independente.

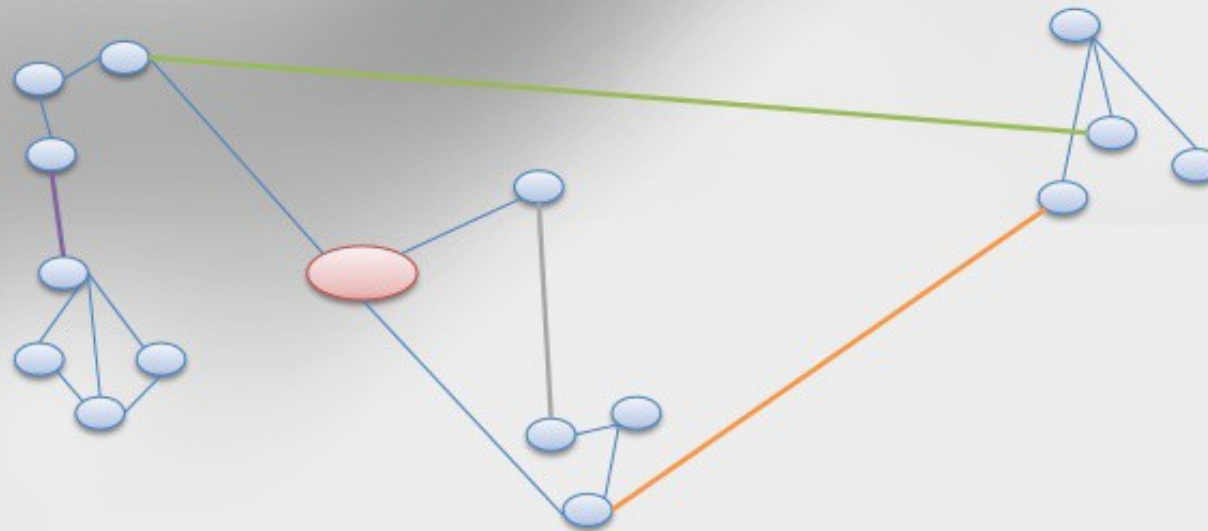
- WEB SERVICE - é a disponibilização de serviço pela internet podendo ser acessado de qualquer lugar, por diversas aplicações. Tem que ser visto por diversos conjuntos: HTTP, XML, UDDI, SOAP, WSDL.
- BARRAMENTO CORPORATIVO - é um barramento de serviços que fornece abstração das camadas, implementação empresarial de sistema de mensagem, onde tem troca de informações, parâmetros entre outros. Combina com a abordagem orientada a eventos e a serviços simplificando integrações de negócios e plataforma heterogêneas, e os ambientes em que não se comunicavam. Algumas de suas funções é roteamento e tradução de mensagem.
- Vantagens de SOA: Reutilização, produtividade, flexibilidade, interoperabilidade, e governança.
- Desvantagens de SOA: complexibilidade, performance, robustez, disponibilidade, segurança.
- Para Implementar o SOA:
 - > Seguir os padrões do mercado
 - > Usar ESB (Barramento de serviços corporativos)
 - > Seguir os princípios do SOA (fraco acoplamento, contato de interfaces, serviços reutilizáveis e entre outros)
 - > Criar serviços que tenham operações atômicas. (Não deve preocupar com dados de outros serviços)
 - > Construa seus serviços iterativamente.

O que é SOA?

Compras

Vendas

Financeiro



"SOA é um paradigma para organização e utilização de competências distribuídas que estão sob controle de diferentes domínios proprietários"

eXtreme Programming - XP

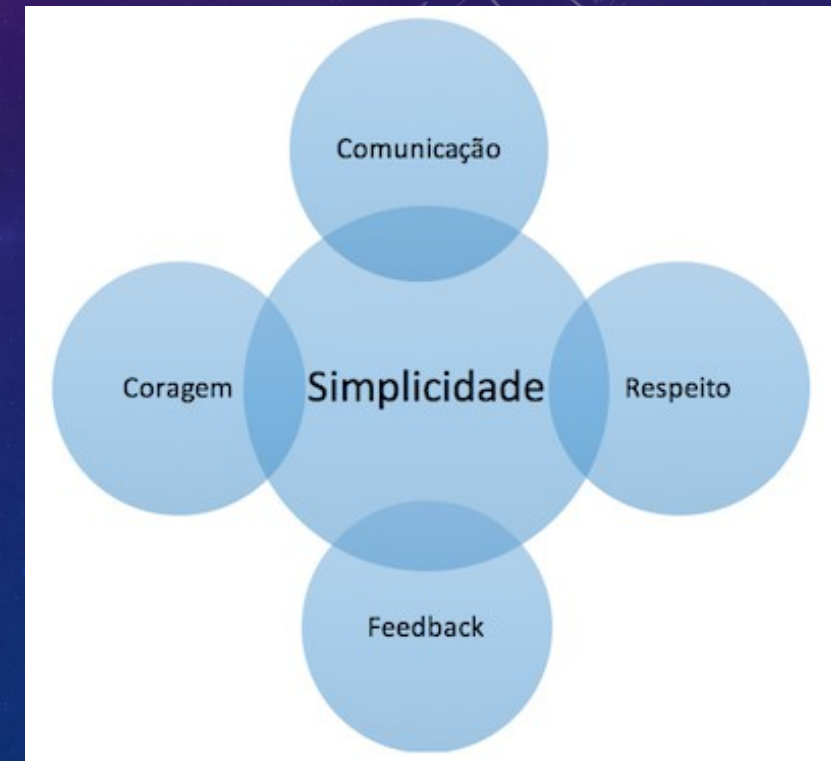


Valores:



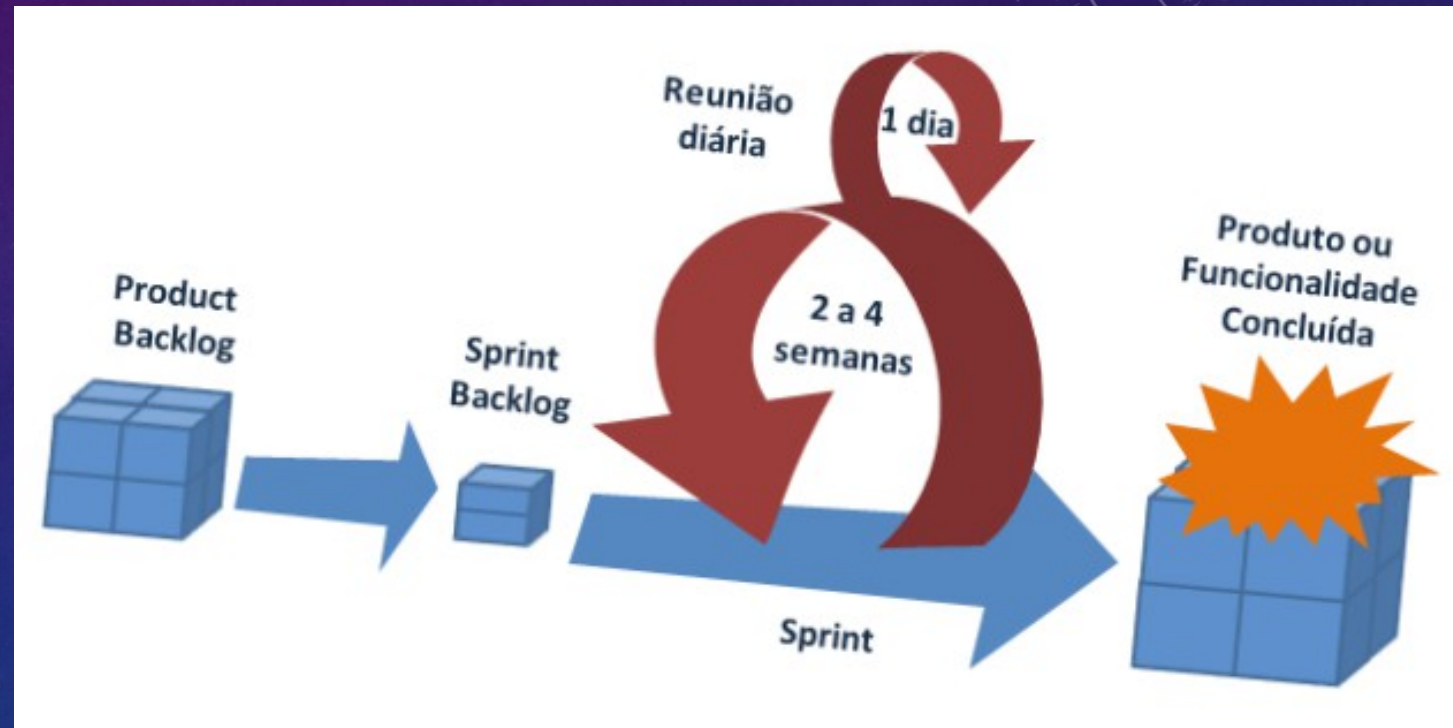
1. Comunicação
2. Feedback
3. Simplicidade
4. Respeito
5. Coragem

De acordo com os valores do XP, os programadores extremos devem sempre estar se **comunicando** tanto quanto com os outros programadores quanto com o próprio cliente, para do segundo receber o **feedback** dos testes diários que devem ser feitos. Mantendo o design sempre **simples** eles devem entregar o projeto ao cliente o mais cedo possível, **respeitando** as mudanças que cada parte da equipe sugerir e tendo **coragem** para enfrentar mudanças na codificação, nas tecnologias utilizadas, etc.



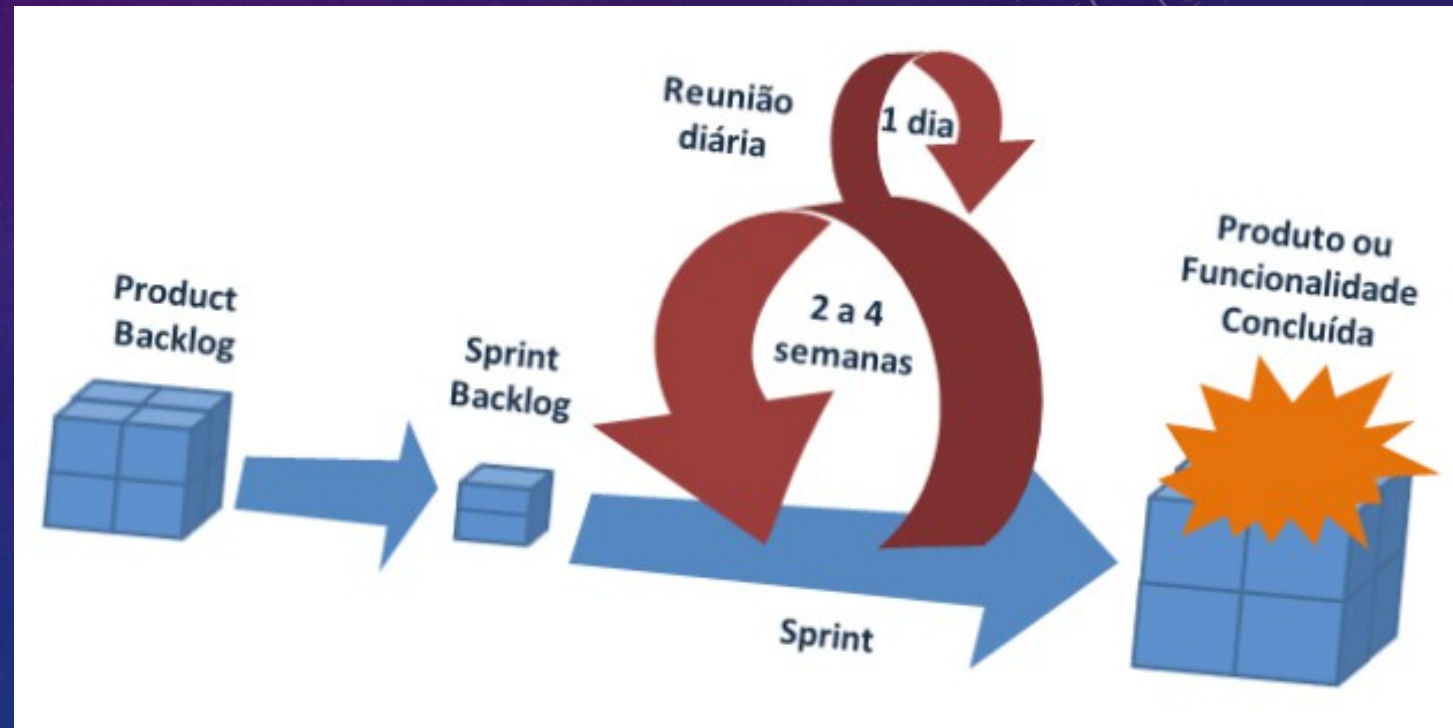
Scrum

- É uma metodologia ágil para gestão e planejamento de projetos de software.
- No Scrum, os projetos são divididos em ciclos de iterações (geralmente de 2 a 4 semanas) chamados de Sprints. O Sprint representa um Time Box dentro do qual um conjunto de atividades deve ser executado.
- A cada dia de uma Sprint, a equipe faz uma breve reunião (normalmente de manhã), chamada Daily Scrum. O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia.

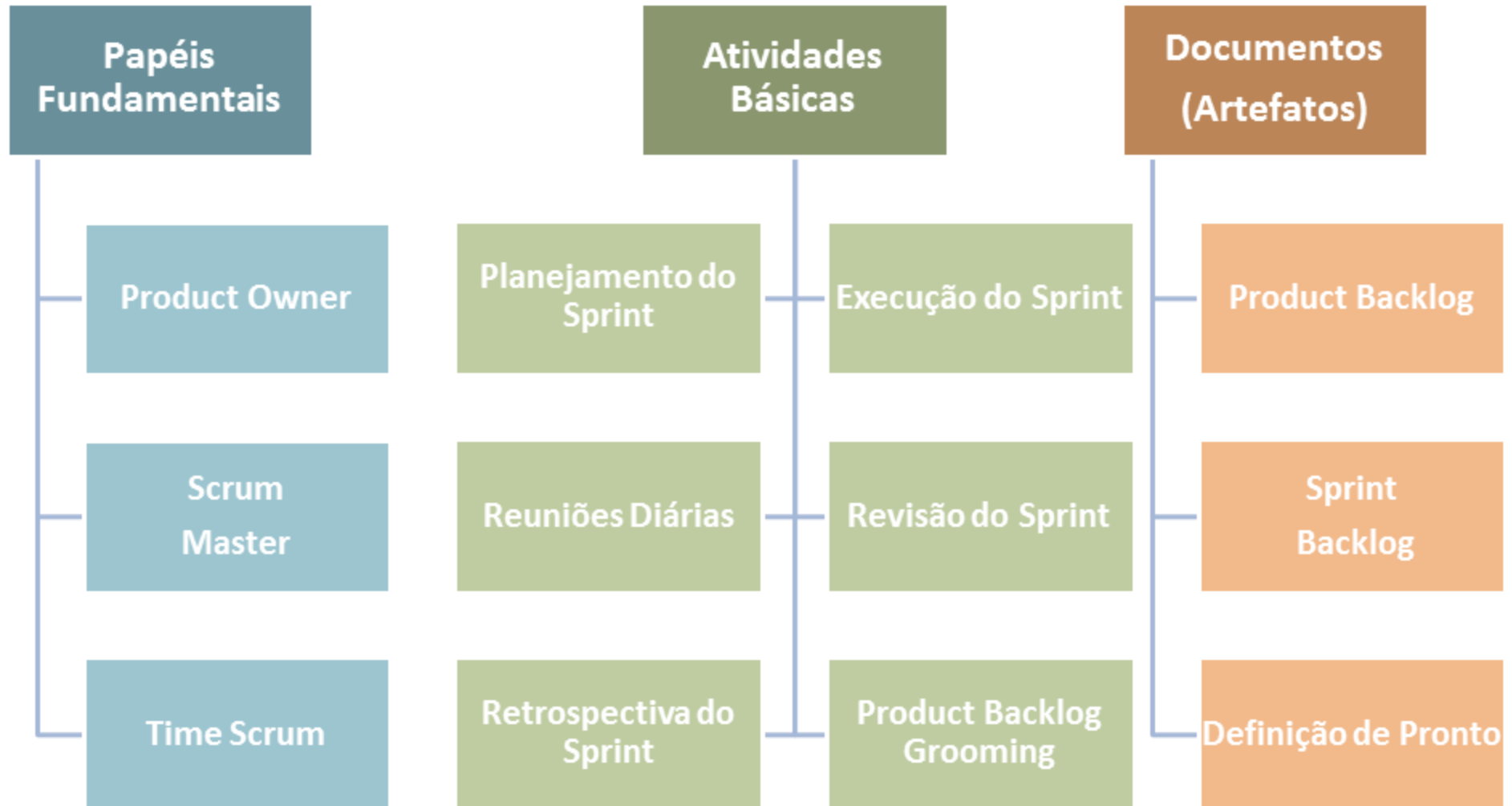


Scrum

- É uma metodologia ágil para gestão e planejamento de projetos de software.
- No Scrum, os projetos são divididos em ciclos de iterações (geralmente de 2 a 4 semanas) chamados de Sprints. O Sprint representa um Time Box dentro do qual um conjunto de atividades deve ser executado.
- A cada dia de uma Sprint, a equipe faz uma breve reunião (normalmente de manhã), chamada Daily Scrum. O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia.



Práticas do Scrum



FDD

Feature Driven Development(FDD): é uma metodologia ágil de desenvolvimento de software. Esta metodologia permite desenvolver um sistema de forma rápida e permite facilmente introduzir novas features (funcionalidades) a este. Uma nova feature demora entre duas horas a duas semanas a ser concluída. O facto de no FDD usar-se processos simples proporciona uma rápida exposição do projeto a novos elementos que venham a fazer parte da equipa, assim como torna o trabalho menos monótono, uma vez que se esta sempre a iniciar/acabar um processo.

- Descrição dos processos:

A metodologia FDD consiste em cinco processos sequenciais. O uso de processos tem várias vantagens: O facto de nos FDD's usarem processos simples e bem definidos faz com que o mover novas utilidades (features) para projectos grandes seja fácil. Pois é como fazer algo simples várias vezes. No desenvolvimento de processos complexos, deve-se dividir em N processos simples, o permite que o progresso do projecto seja mais rápido. - A integração de novos elementos nas equipas é rápida, pois o facto de os processos serem simples faz com que seja fácil a compreensão destes por parte dos novos membros. - O facto de os processos serem simples, faz com que a equipa esteja sempre concentrada no trabalho uma vez que estão constantemente a acabar/iniciar um novo processo. Não causa uma monotonia do trabalho.