

A aula interativa do **Bootcamp Arquiteto(a) de Software** começará em breve!

Atenção:

- 1) Você entrará na aula com o microfone e o vídeo DESABILITADOS.
- 2) Apenas a nossa equipe poderá habilitar seu microfone e seu vídeo em momentos de interatividade, indicados pelo professor.
- 3) Utilize o recurso Q&A para dúvidas técnicas. Nossos tutores e monitores estarão prontos para te responder e as perguntas não se perderão no chat.
- 4) Para garantir a pontuação da aula, no momento em que o professor sinalizar, você deverá ir até o ambiente de aprendizagem e responder a enquete de presença. Não é necessário encerrar a reunião do Zoom, apenas minimize a janela.

Fundamentos

SEGUNDA AULA INTERATIVA

PROF^a. PRISCILLA SCARPELLI

Fundamentos

SEGUNDA AULA INTERATIVA

PROF^a. PRISCILLA SCARPELLI

Nesta Aula



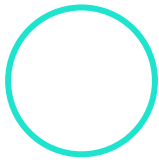
- ☐ Risco de Projeto
- ☐ 5W2H
- ☐ Frameworks Arquiteturais
- ☐ Modelagem Arquitetural
- ☐ Ciclo de Vida Projeto
- ☐ Estudo de Caso

O que é Risco?



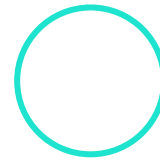
- ❑ Riscos são inerentes a projetos de software;
- ❑ Avaliar o quão impactante é uma situação de incerteza;
- ❑ Trabalhar em plano de ação;
- ❑ Risco é um evento ou condição incerta;
- ❑ Pode afetar um ou mais objetivos do projeto, como:
 - ❑ Escopo;
 - ❑ Cronograma;
 - ❑ Custo;
 - ❑ Qualidade.

Tipos de riscos



Positivo

- Também chamado de oportunidade;
- Pode estar relacionado a uma incerteza, mas que pode alavancar o valor gerado pela solução.

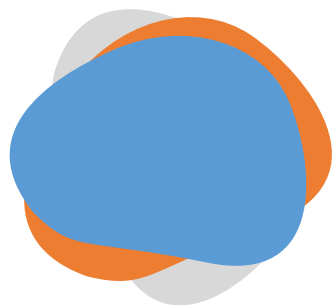


Negativo

- Também chamado de ameaças;
- Pode provocar atrasos na entrega do produto e aumentar os custos do projeto.

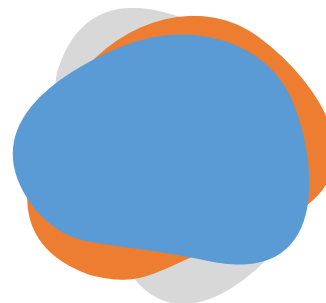


Fontes de Risco

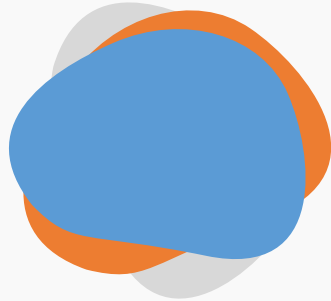


Capacitação técnica dos desenvolvedores.

Requisitos mal definidos.

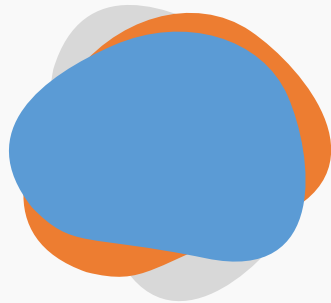


Fontes de Risco

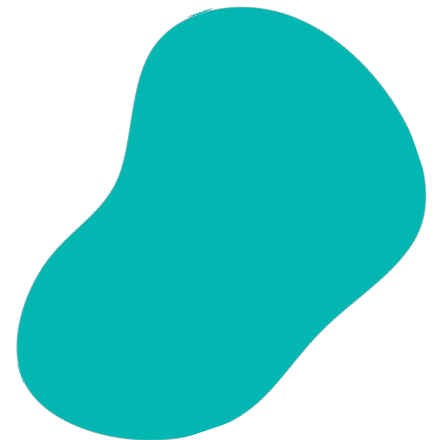
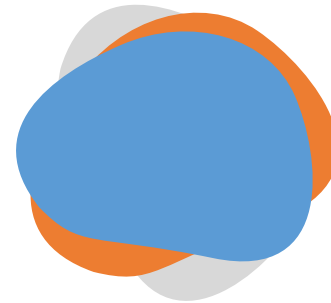


Prazos apertados.

Mercado de desenvolvimento aquecido.

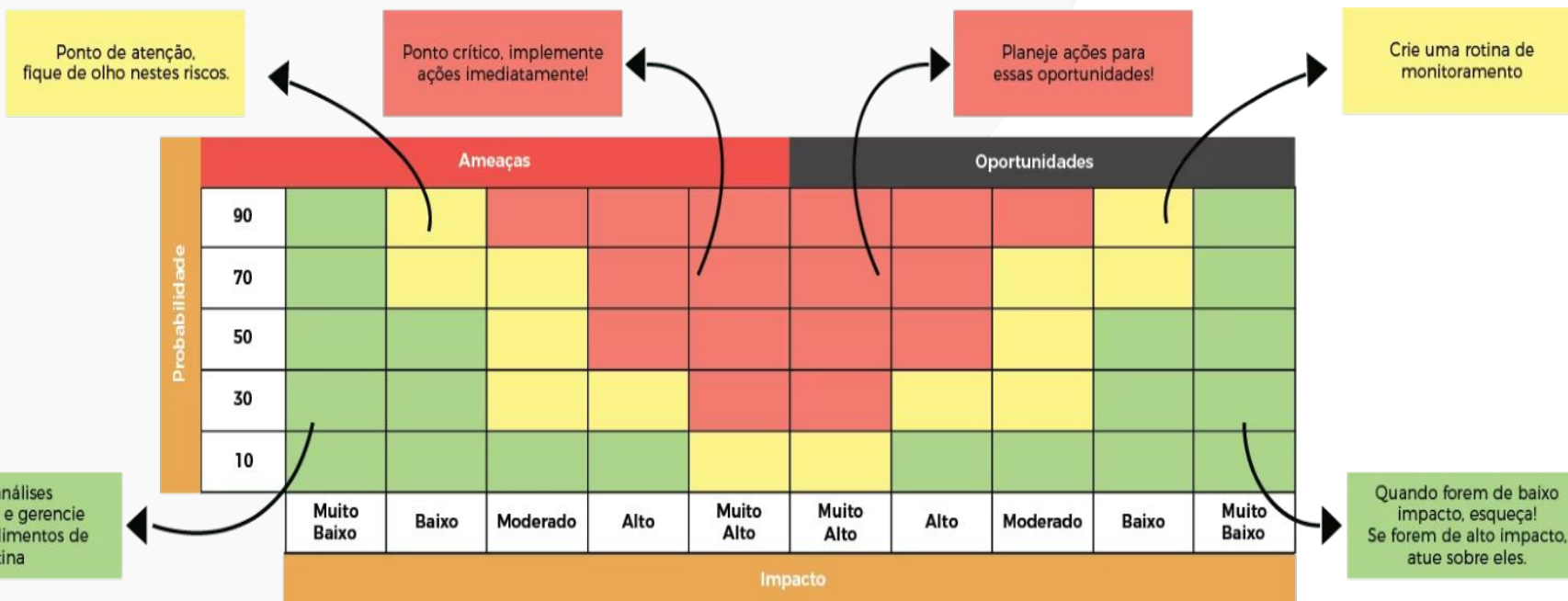


Constantes mudanças de priorização.

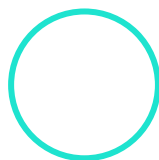


Matriz de impacto

IGTI

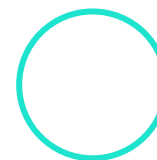


Análise da matriz



Oportunidade

- ☐ Vermelho: devemos gastar esforços e traçar planos para que aconteçam.
- ☐ Amarelo: monitore e tente estudar as de impacto grande.
- ☐ Verde:
 - ☐ baixo impacto, não perca tempo;
 - ☐ impacto alto, tente traçar planos de ação para que aconteçam.

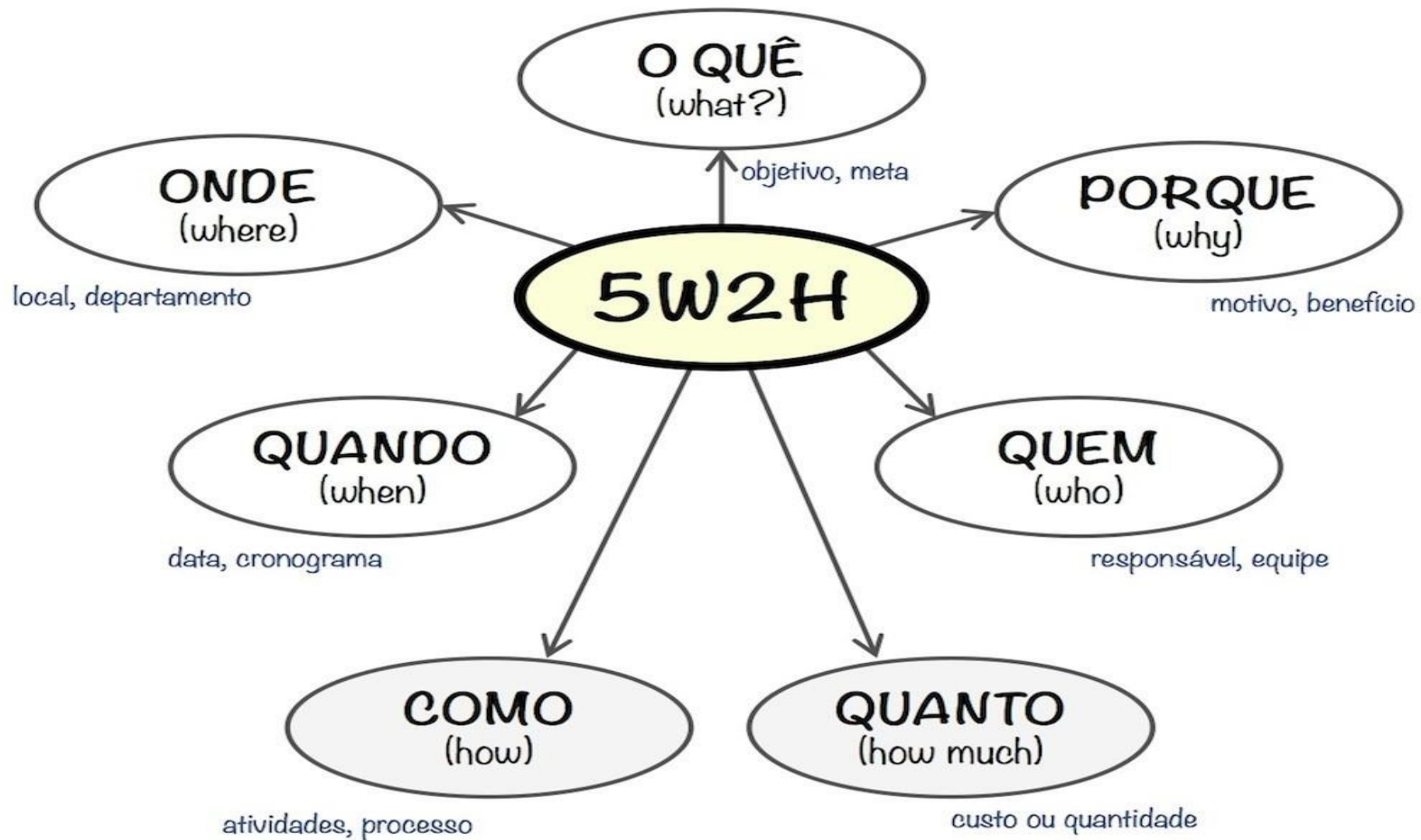


Ameaças

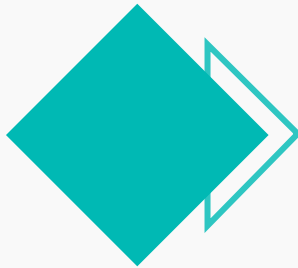
- ☐ Vermelho: foco na mitigação e na criação de ações.
- ☐ Amarelo: atenção para que eles não transitem para a região vermelha.
- ☐ Verde: acompanhados periodicamente.

5W2H

IGTI

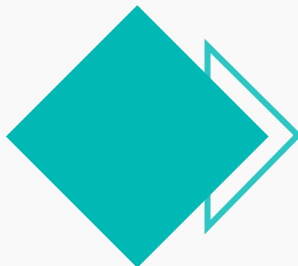
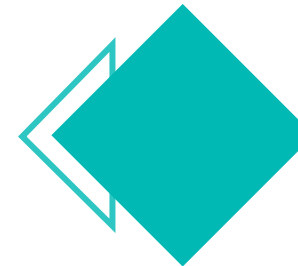


5W2H



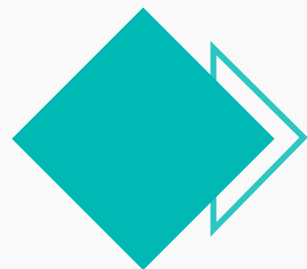
O quê? (What?): Descrição da ação ou atividade que será executada.

Quem? (Who?): Quem são as pessoas envolvidas.



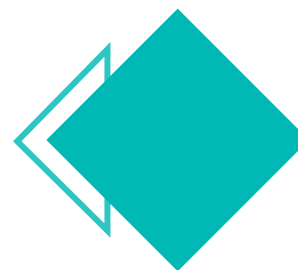
Onde? (Where?): Onde a ação será executada.





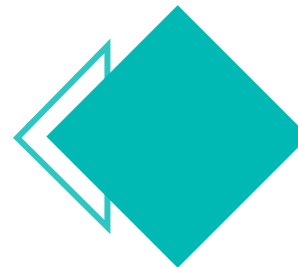
Quando? (When?): Aqui responderemos o prazo para execução.

Como? (How?): Descrição da forma de como será feito.



Quanto (How Much?): Qual o custo envolvido.

Por quê? (Why): Descrição da razão por tomar esta ação.

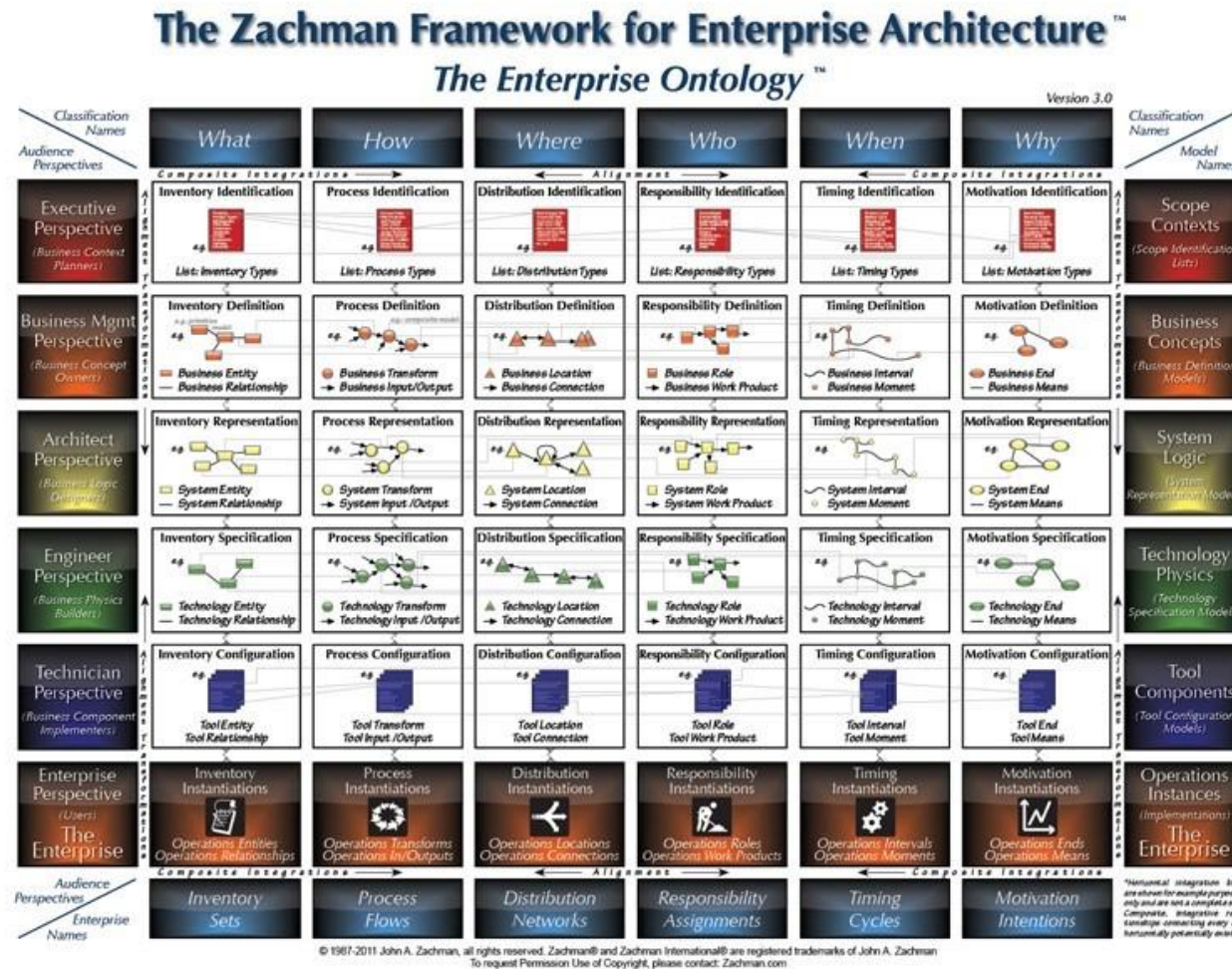


Zachman



- ❑ O modelo precisa se relacionar com o negócio;
- ❑ O negócio contém diferentes perspectivas pelos diferentes papéis;
- ❑ Cada perspectiva possui uma necessidade, e entender a fundo essas necessidades ajuda a construir um modelo mais realista.

Matriz Zachman

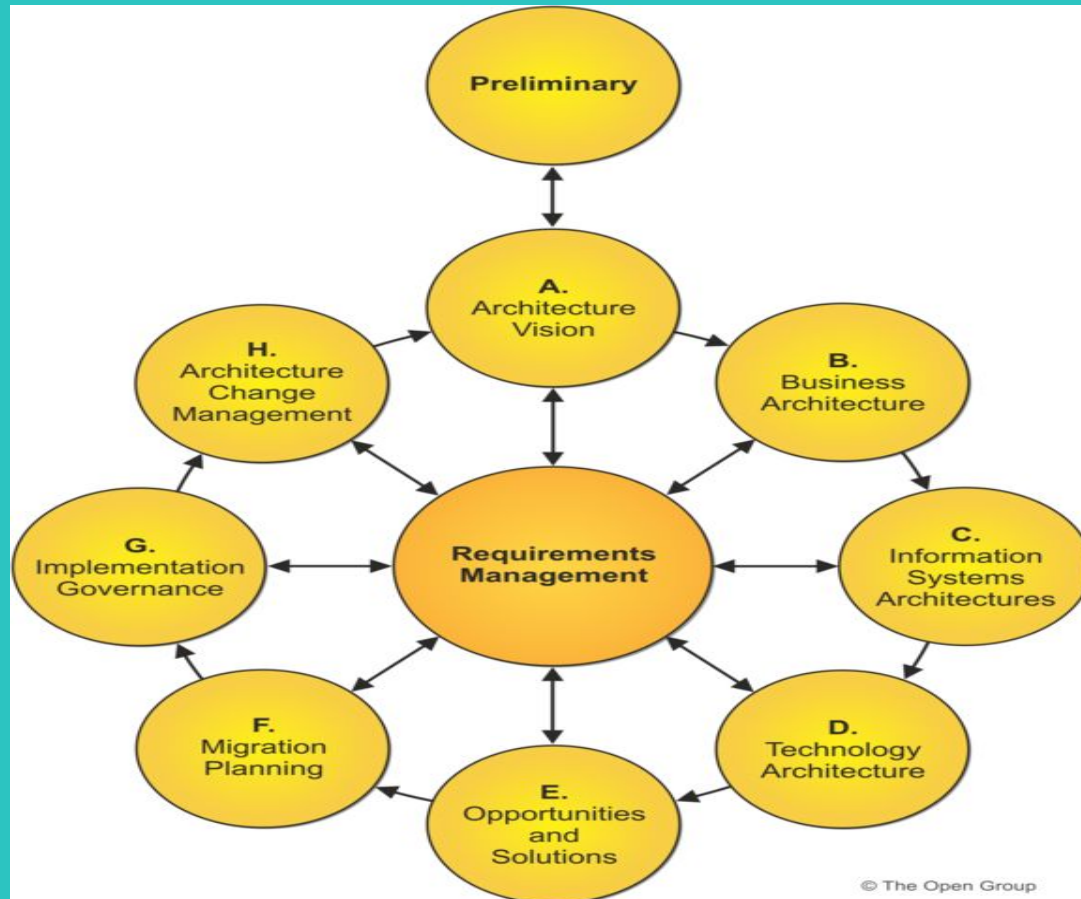


TOGAF



- The Open Group Architecture Framework (TOGAF);
- Framework que oferece uma estrutura de alto nível para o desenvolvimento de software;
- Ajuda a organizar o processo de desenvolvimento;
- Visa:
 - Reduzir erros;
 - Manter cronograma e orçamento;
 - Alinha a TI com o negócio.
- O TOGAF é aberto e disponibiliza uma metodologia customizável para orientar a arquitetura.

TOGAF ADM



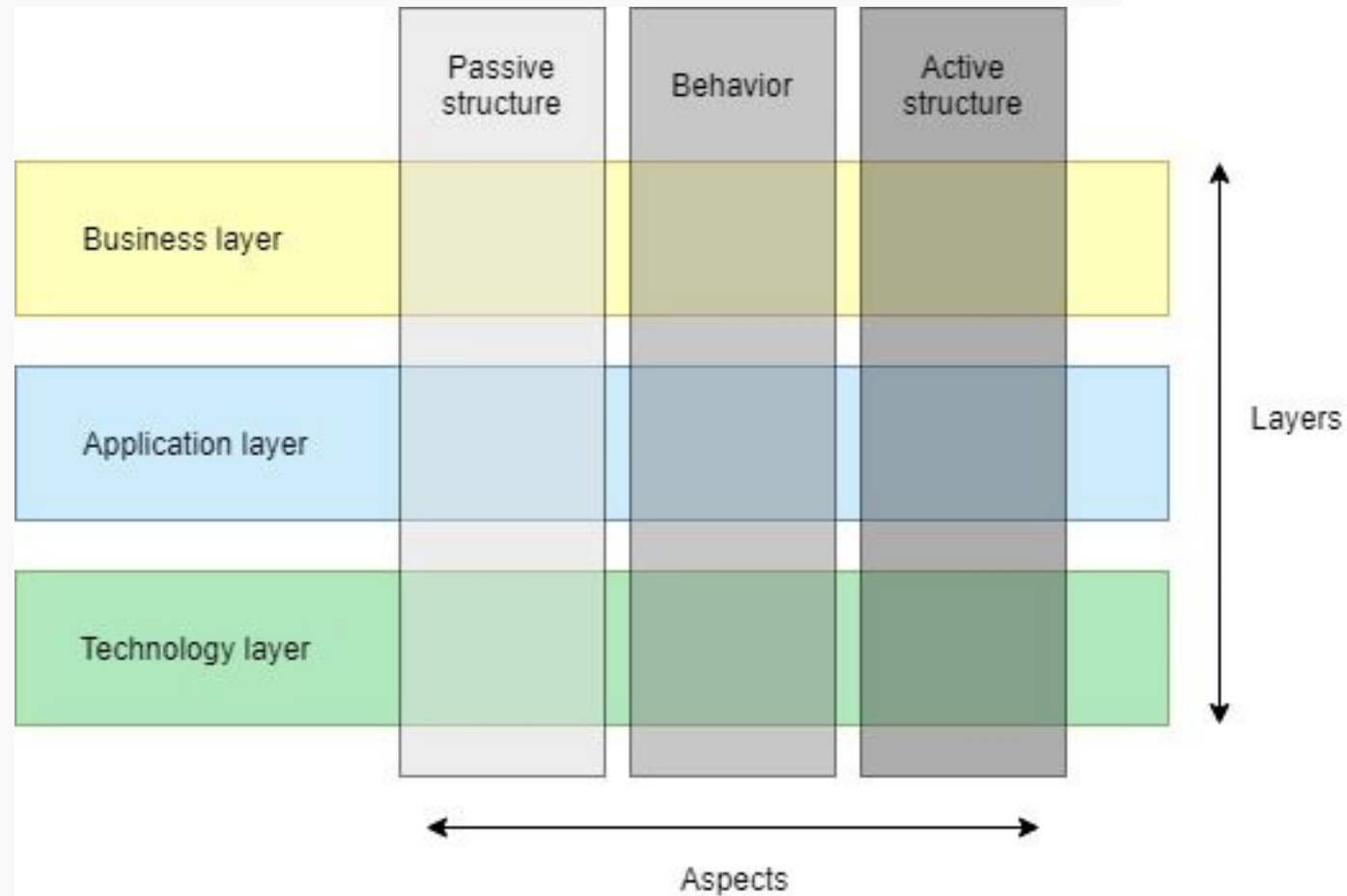
ArchiMate



- Linguagem visual;
- Foi desenvolvida e é mantida pelo “The Open Group”;
- Tem um fit natural com o TOGAF;
- É uma linguagem ampla e abrangente que foi projetada para auxiliar a descrever a arquitetura corporativa.

ArchiMate

IGTi



UML



- Na maioria dos casos, é a primeira linguagem de modelagem que aprendemos;
- A Linguagem Unificada de Modelagem (Unified Modeling Language – UML) é uma linguagem para a elaboração da estrutura do projeto;
- Pode ser utilizada para:
 - Visualização;
 - Especificação;
 - Construção;
 - Documentação de artefatos.

Diagramas Estruturais



- São diagramas utilizados para modelar aspectos estáticos e estruturais do software;
- Nestes diagramas, representaremos:
 - Estruturas de classes de software;
 - Interfaces com outros softwares e componentes pouco vulneráveis a mudanças.
- Exemplos de diagramas estruturais:
 - Diagrama de classe;
 - Diagrama de objetos;
 - Diagrama de componentes;
 - Diagrama de implantação;
 - Diagrama de pacote;
 - Diagrama de estrutura.

Diagrama Comportamental



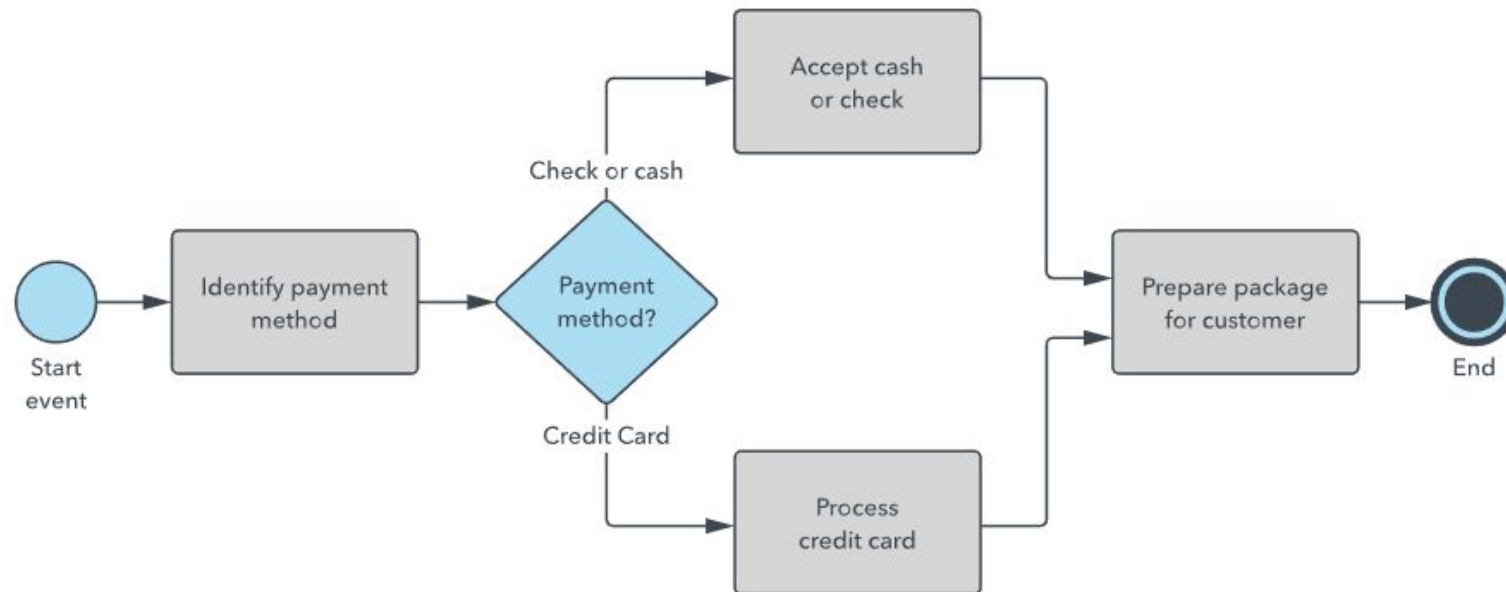
- Utilizados para modelar aspectos dinâmicos e de interação que envolvem o software;
- Representaremos ações de usuário, dinâmica de troca de mensagem e outras ações;
- Exemplos de diagrama comportamental:
 - Diagrama de casos de uso;
 - Diagrama de sequência;
 - Diagrama de colaboração;
 - Diagrama de estados;
 - Diagrama de atividade.

BPMN



- Business Process Model and Notation (BPMN);
- É um método que se utiliza de fluxogramas para documentar um processo de negócio de ponta a ponta;
- Fornece uma linguagem padrão comum para todas as partes interessadas:
 - Analista de negócio;
 - Participantes do processo;
 - Gerentes;
 - Desenvolvedores.

Diagrama BPMN

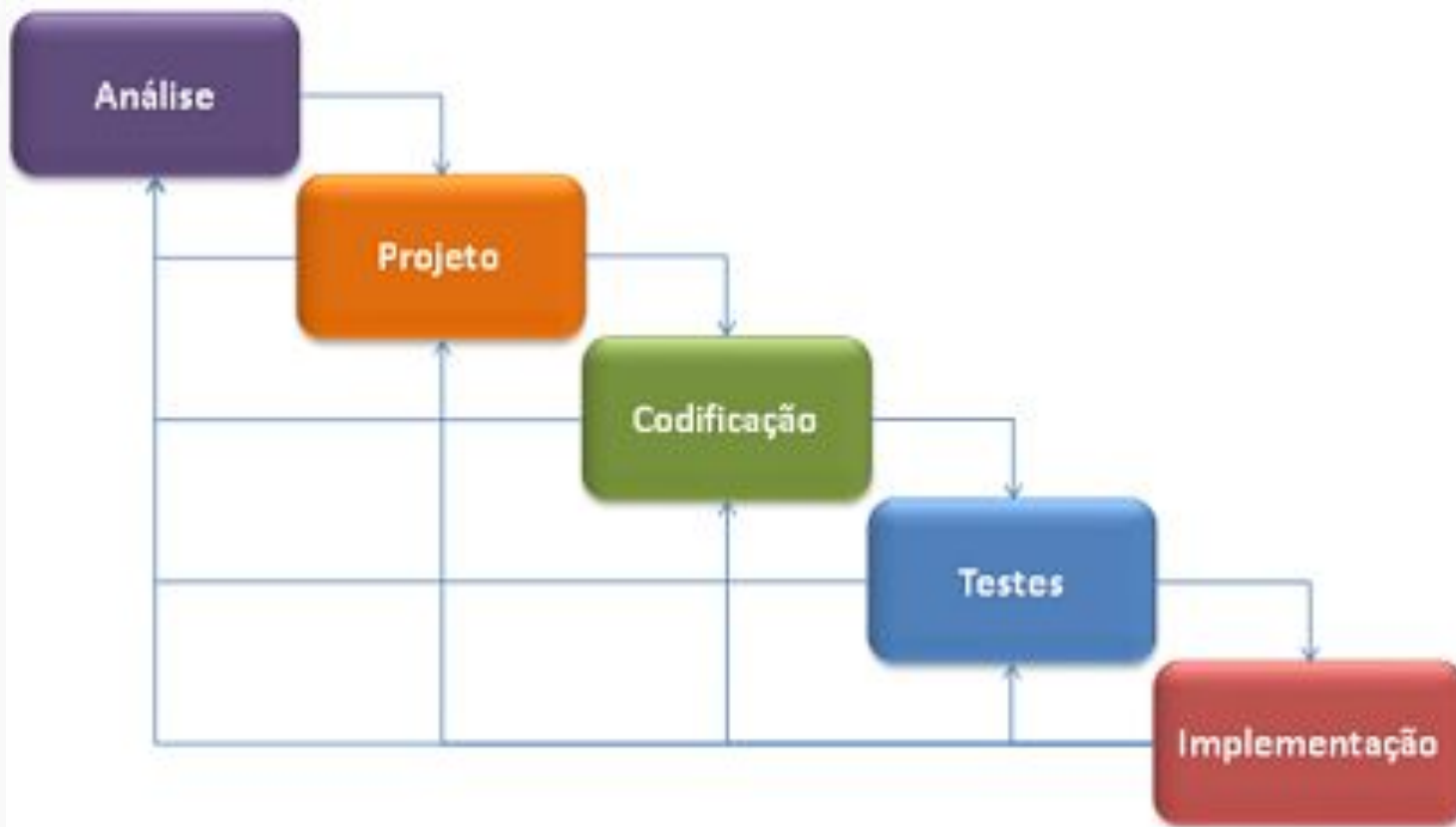


Ciclo de Vida



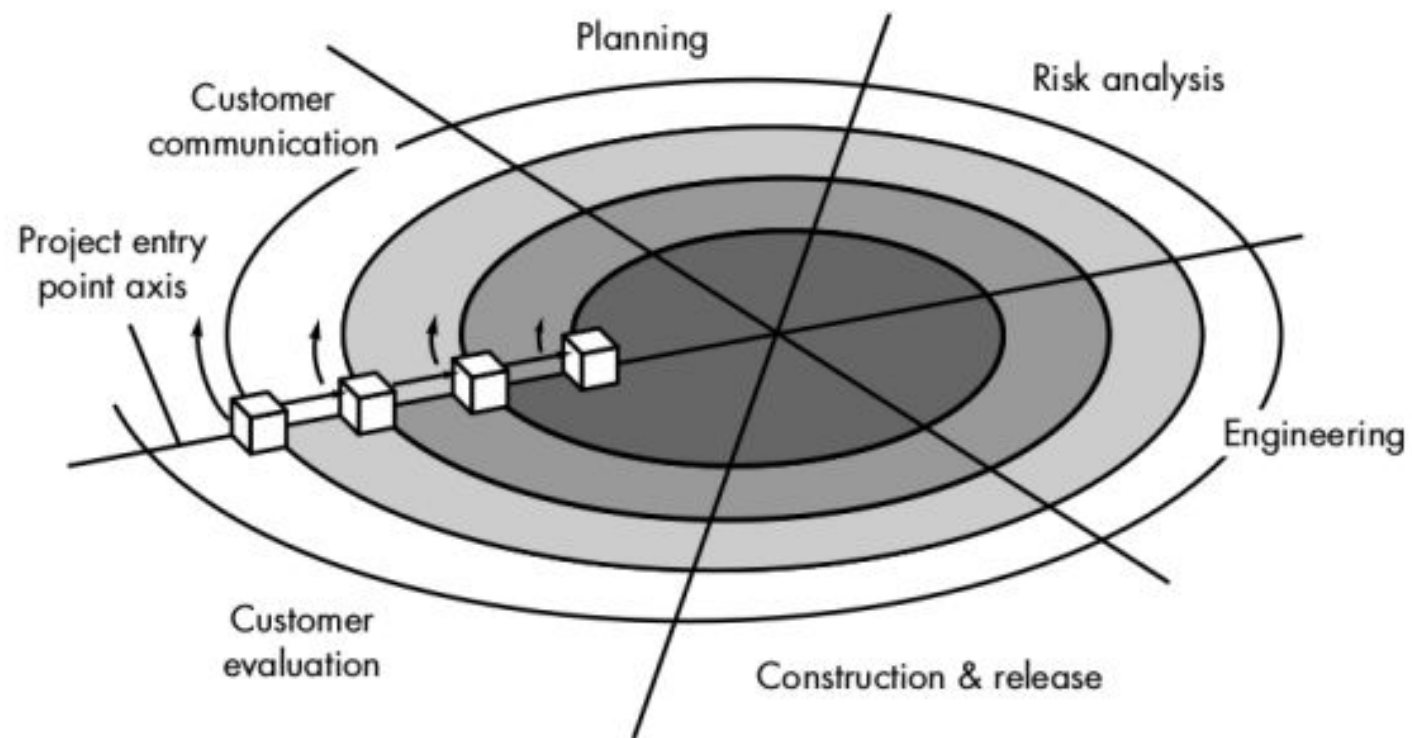
- O ciclo de vida de um software compreende os processos e atividades que o envolvem;
- Existem vários modelos de ciclo de vida de desenvolvimento;
- Cada modelo de ciclo de vida tem o seu esforço para poder gerenciá-lo.

Modelo Cascata

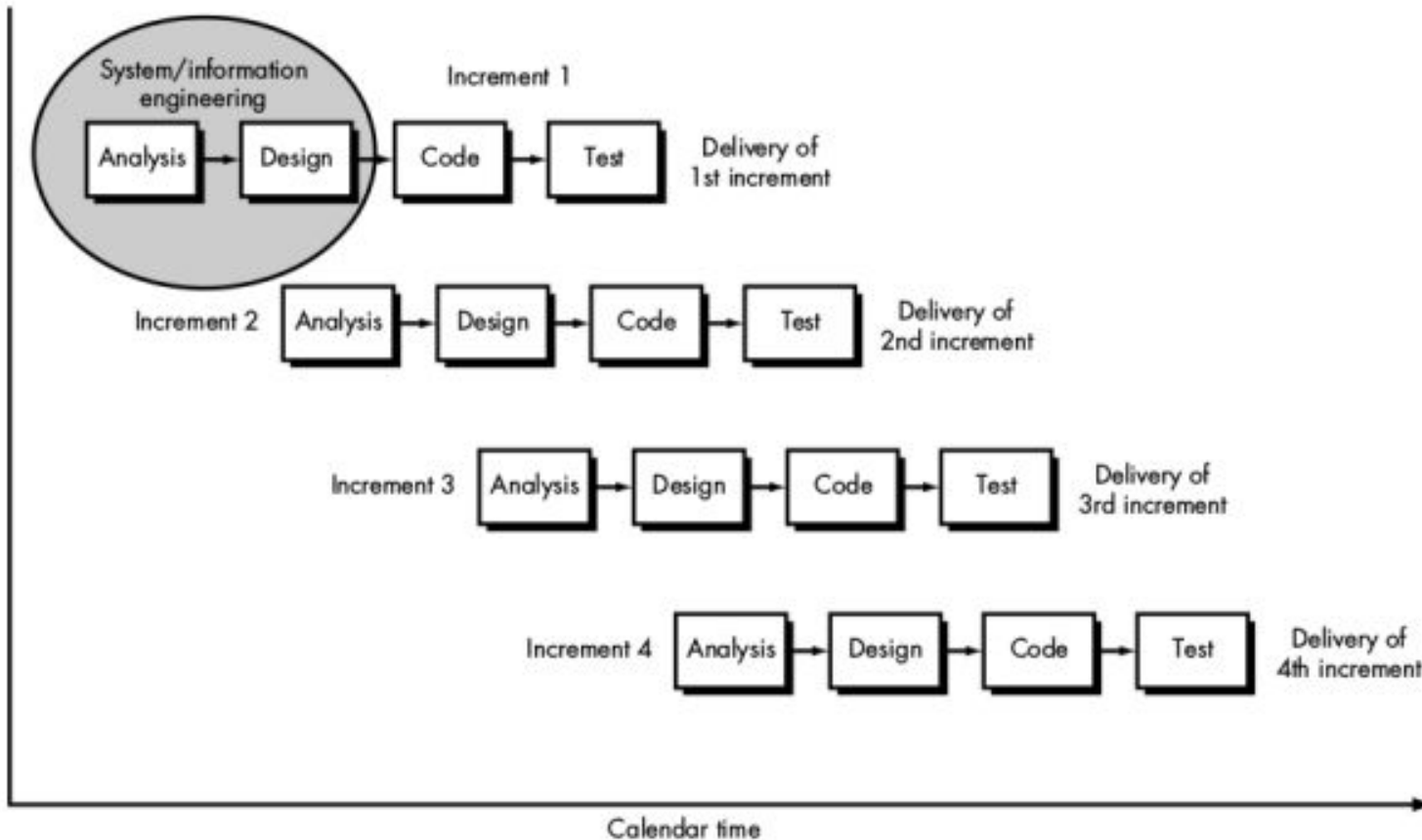


Modelo Espiral

iGTi



Modelo Incremental



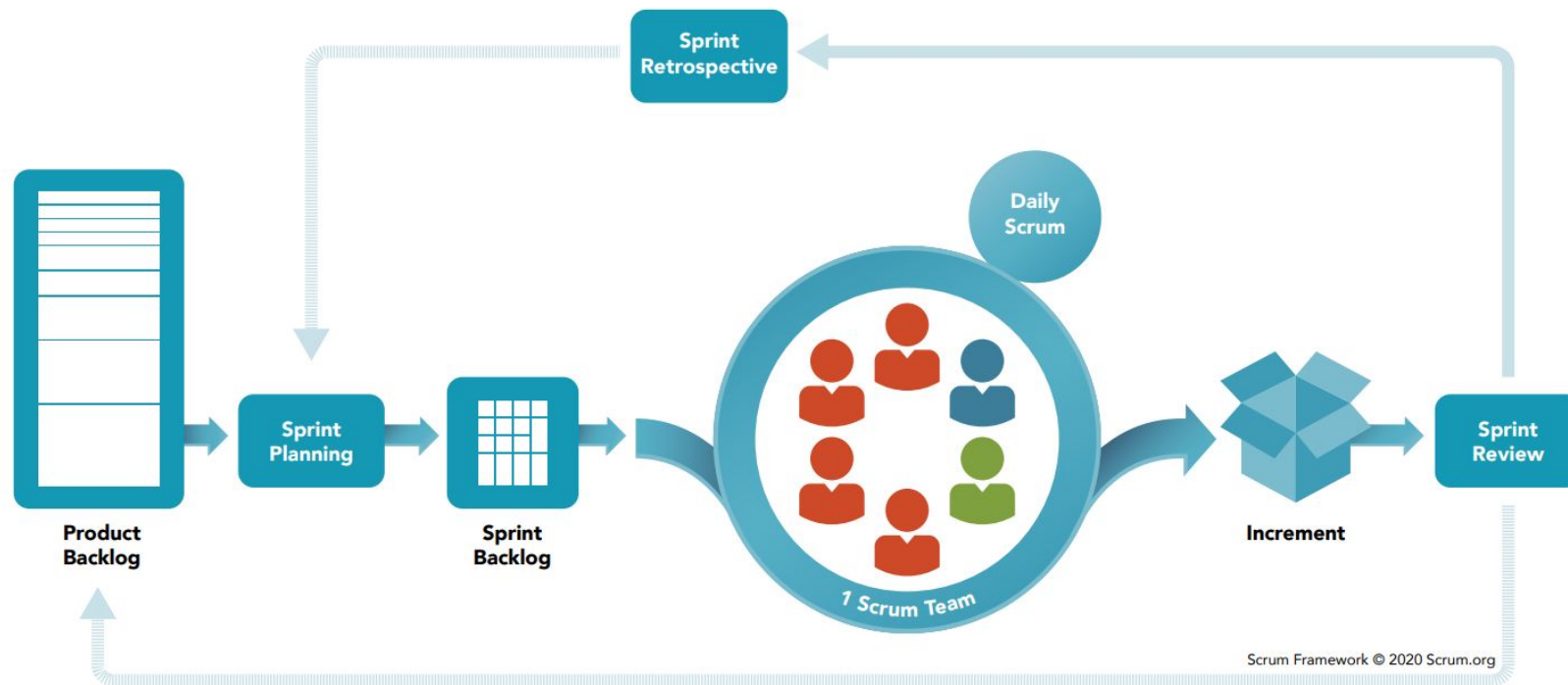
Scrum



- Framework que ajuda pessoas, equipes e organizações a gerar valor nas soluções;
- Inicialmente estava associado apenas a processos de desenvolvimento de software;
- Ferramenta para suporte no desenvolvimento de qualquer produto;
- Forma de gerenciamento de projetos complexos.

Scrum Framework

IGTi



DevOps

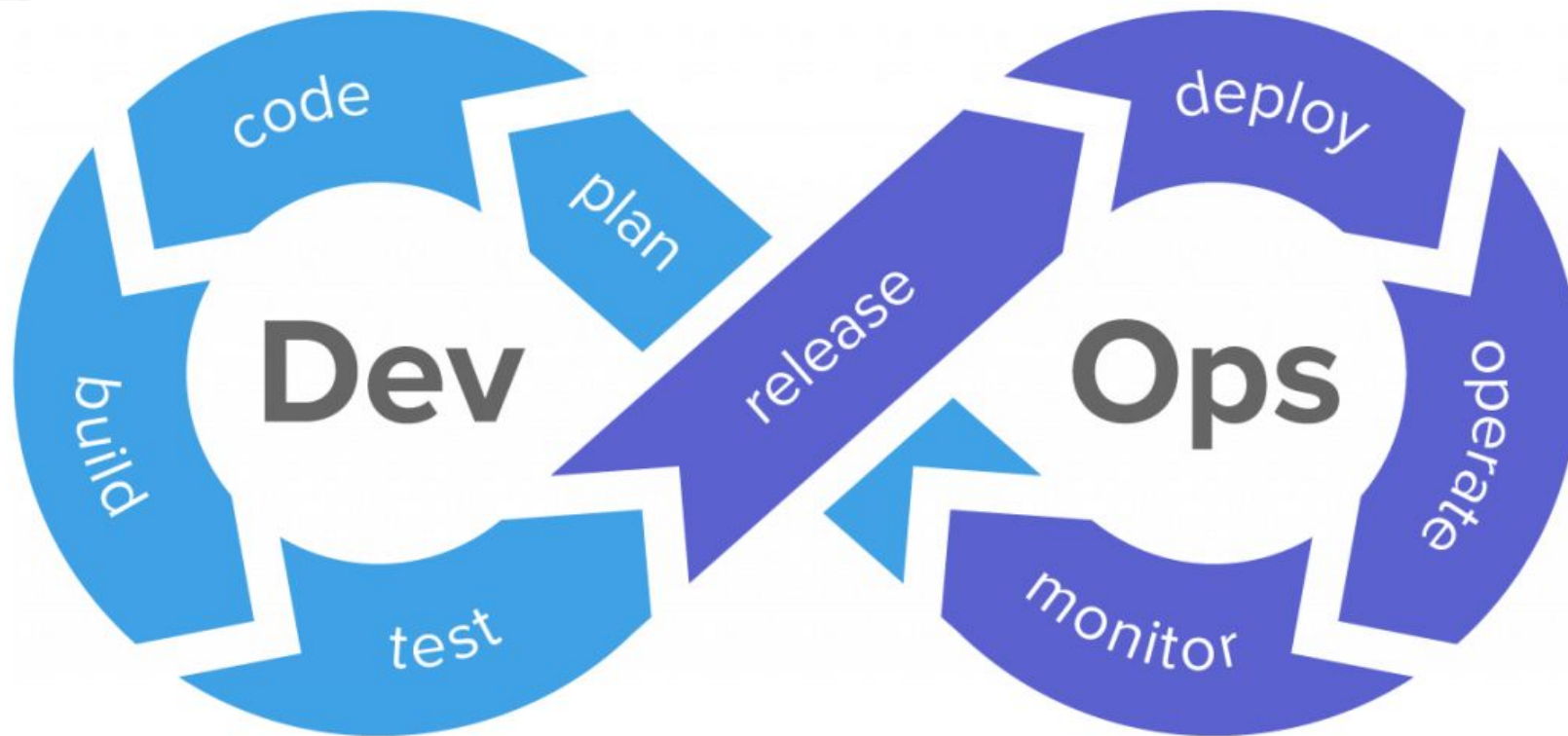
- É a união das palavras “Desenvolvimento” e “Operação”;
- É um conjunto de boas práticas de colaboração com o objetivo de realizar entregas de software mais seguras;
- Automatização de processos de delivery.



igti

Ciclo DevOps

iGTi



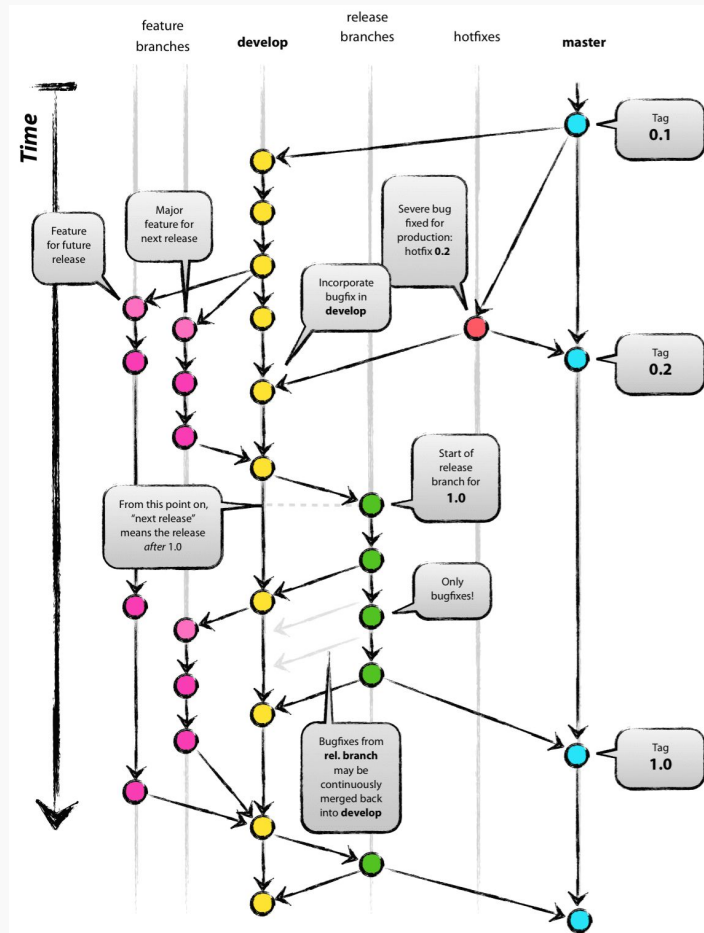
Versionamento Git



- Git é o sistema de controle de versão mais utilizado;
- Um projeto open source criado em 2005, por Linus Torvalds;
- Alguns conceitos básicos do git:
 - Commit;
 - Branch;
 - Merge;
 - Clone.

Gitflow

IGTI



Estudo de Caso

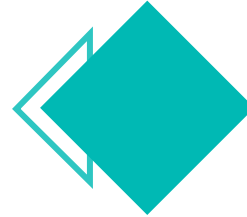


- ❑ Fomos contratados por uma empresa de desenvolvimento para desenvolver uma aplicação;
- ❑ O objetivo da aplicação é disponibilizar uma forma de os clientes fazerem operação na bolsa de valores brasileira;
- ❑ Devemos considerar a seguinte premissa:
 - ❑ O saldo do cliente deve ser obtido a partir do sistema de conta;
 - ❑ A integração com o sistema de conta será através de uma API Rest.

Requisitos do Projeto

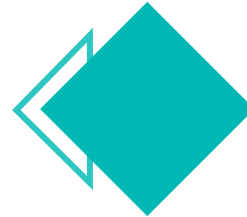
IGTI

O cliente poderá acompanhar a carteira de ações em tempo real.



O cliente deve conseguir realizar compra e venda de quaisquer ativos negociados na bolsa.

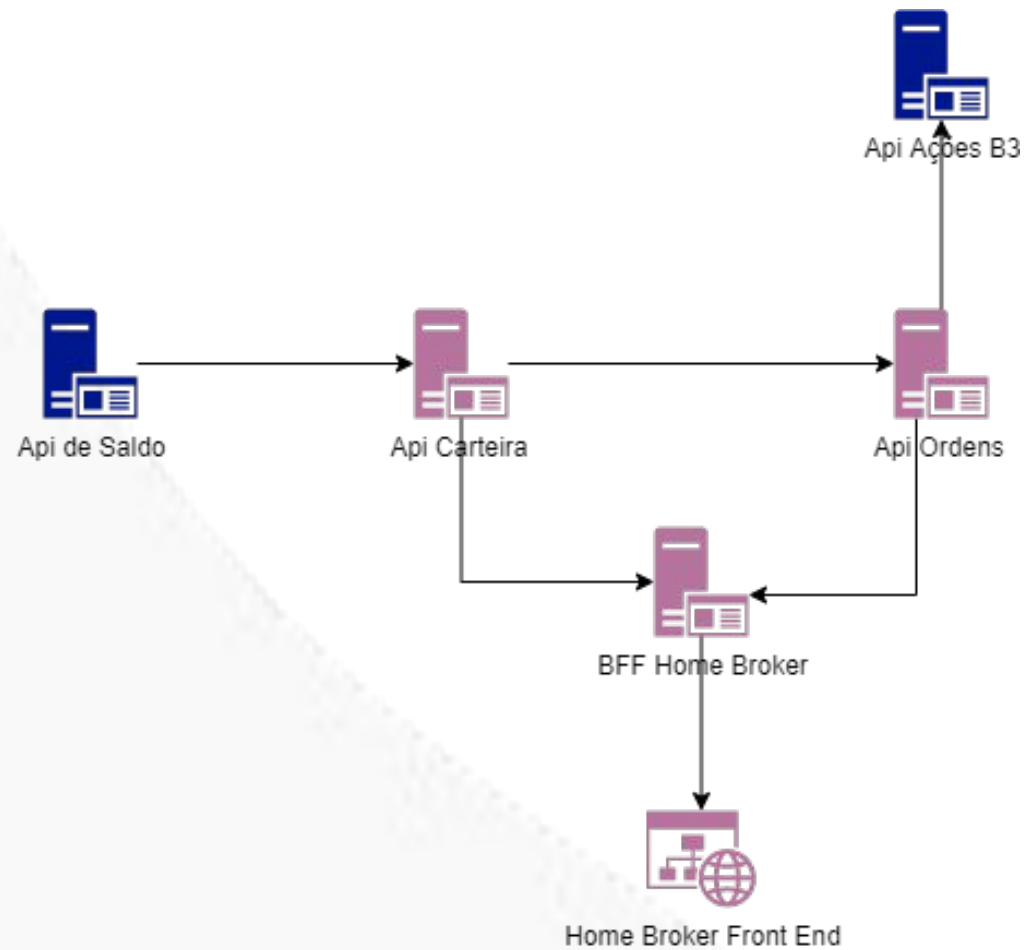
Ordens que forem colocadas e que não forem executadas devem ser reportadas para o cliente tentar novamente.



Todas as ordens requisitadas devem ser armazenadas para possíveis auditorias.

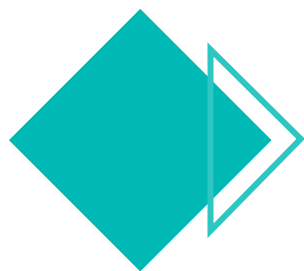
Arquitetura

iGTi



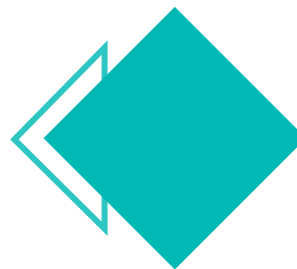


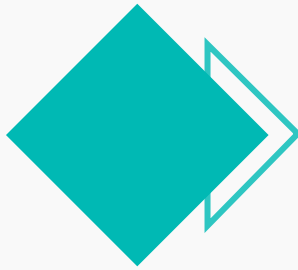
Arquitetura



A Api de Ações e a Api de Saldo são externas e fora do nosso controle.

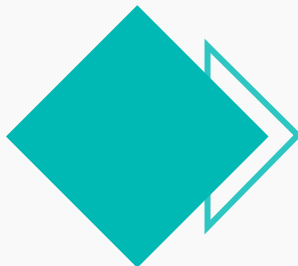
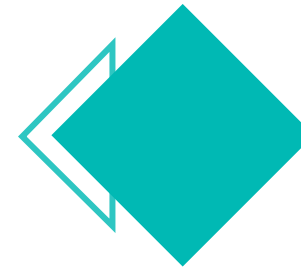
A Api Carteira tem como responsabilidade controlar o saldo disponível.



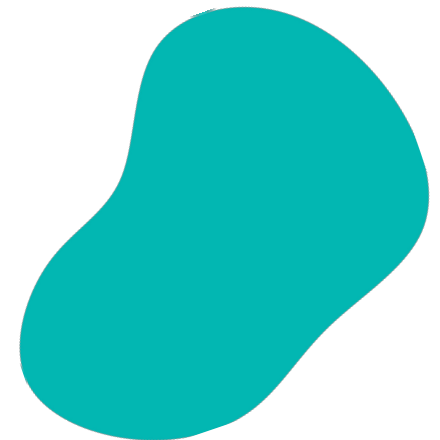


O BFF Home Broker é o “Back End for Front End” que irá fornecer as operações para o Home Broker.

O Home Broker Front End é onde efetivamente será a interface em que o usuário irá operar na bolsa.



A Api Ordens tem como papel controlar as ordens de compra e venda que são criadas

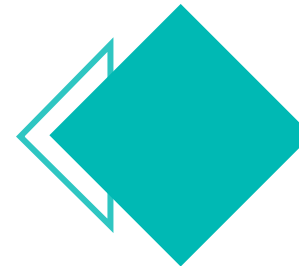


Riscos do Problema



Não existe um endpoint válido implementado para as informações que precisamos da API de Saldo.

O projeto não tem mão de obra suficiente para a entrega da solução.



Falta de entendimento da API da B3.

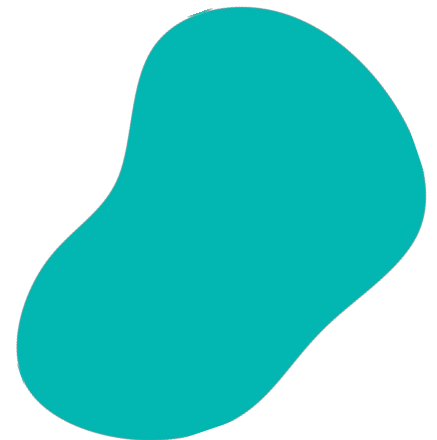
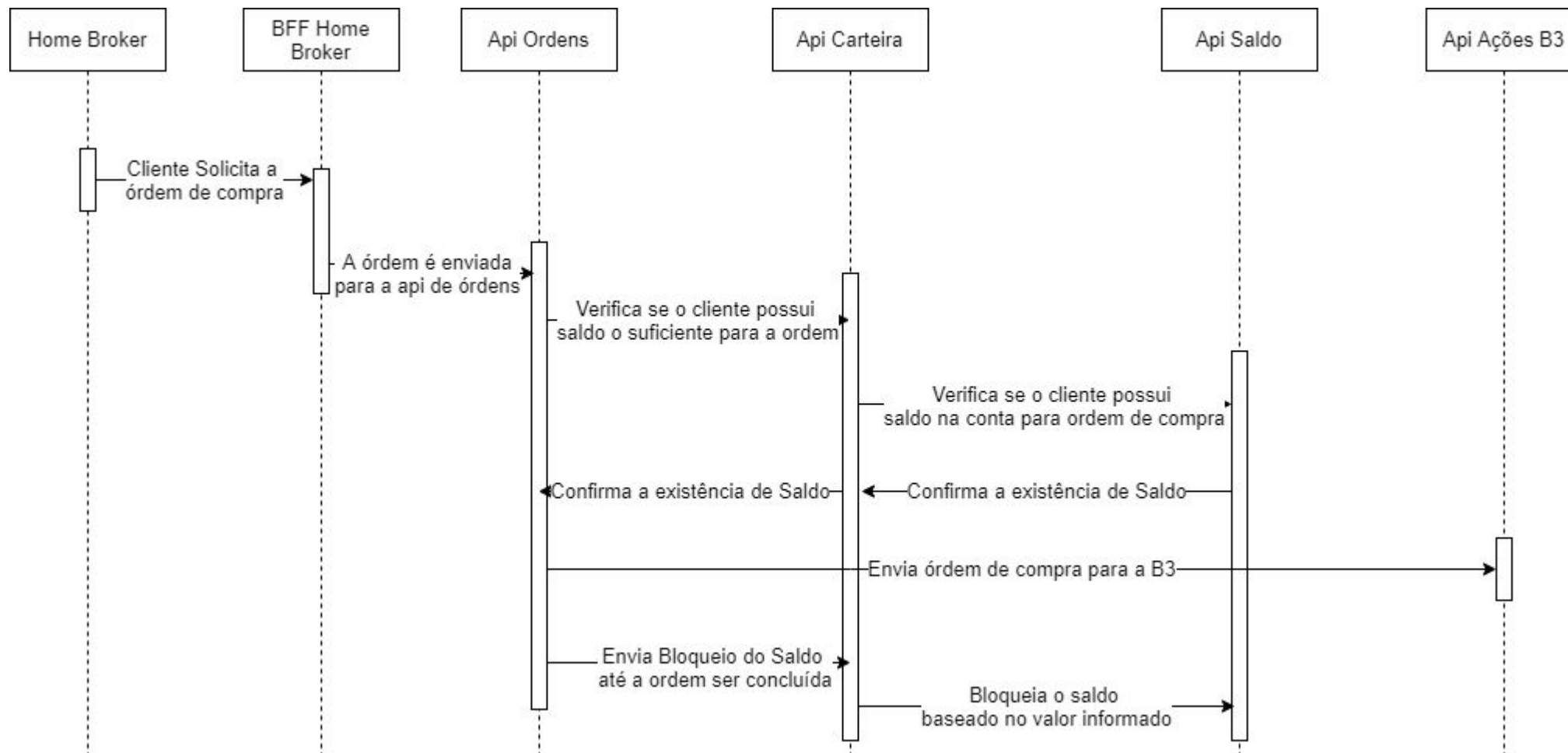


Diagrama Sequência



Conclusão



- Um arquiteto precisa saber identificar os riscos associados à solução proposta;
- Saber identificar o framework arquitetural que se adapta ao projeto ajuda no entendimento dos envolvidos;
- O arquiteto precisa ajudar na gestão do desenvolvimento do projeto e, para isso, precisa saber acompanhar o ciclo de vida de um desenvolvimento.