



SÃO
PAULO
TECH
SCHOOL



TECNOLOGIA DA INFORMAÇÃO - TI

Professores:

Rafael Petry – 1ADSB

Marcos Antonio – 1CCOA / 1CCOB

Monteiro – 1ADSA

Fábio Figueredo – SIS



S3

Sustentação

- Introdução a Processo de Desenv. De SW
- Governança
- ITIL (Incidentes, Problemas e Mudanças)
- Suporte de TI

- Fluxograma do suporte
- Ferramenta de Help Desk
- Documento de Mudança

Entrega: 26/05/2025

S2

Metodologia e Processos

- Metodologia de Gestão de Projetos
- Arquitetura de TI

- Diagrama da Solução
- Planilha Product Backlog
- Planilha Sprint Backlog

Entrega: 22/04/2025

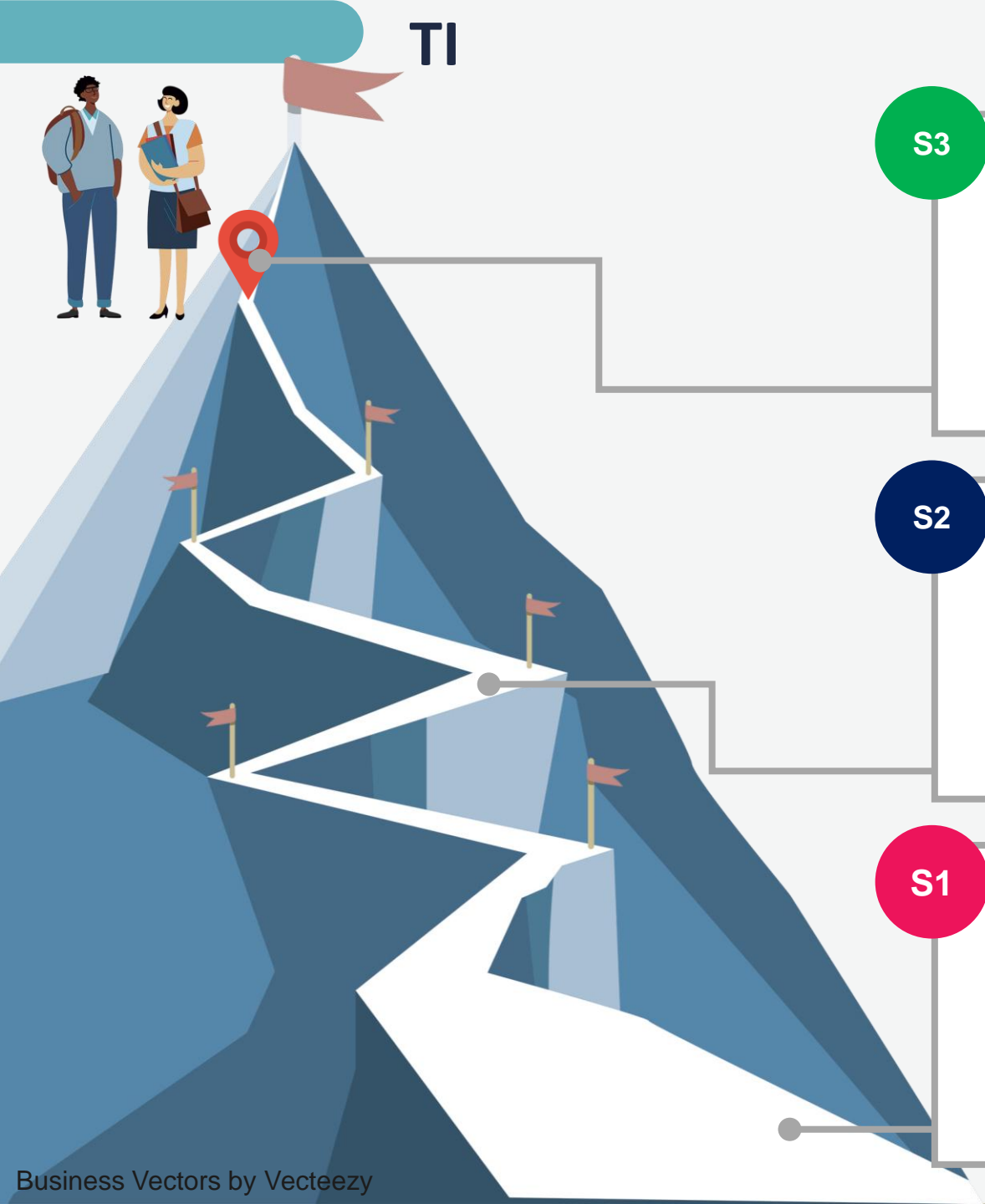
S1

Introdução + Planejamento

- Introdução a TI
- Projeto vs Processo
- Requisitos
- Documentação do projeto

- Requisitos na Ferramenta
- Ferramenta de Gestão
- Documentação do Projeto

Entrega: 17/03/2025



Planejamento para final de semestre



05 a 09/05 – Des. Soft+Gov+ITIL
12 a 16/05 – ITIL+Fluxo+GMUD
19 a 23/05 – V. Entregáveis
26 a 30/05 – AC3
02 a 06/06 – Sprint 3
09 a 13/06 – Dinâmica 1
16 a 20/06 – Dinâmica 2



Pontos de atenção

Colaboração
Comprometimento
Disciplina



Avaliações e Sprint

- Se prepare
- Se organize
- Grupo de estudo
- Dividir angustias



CHAMADA

SP3 – Aulas 1

Ciclo de Desenvolvimento de SW

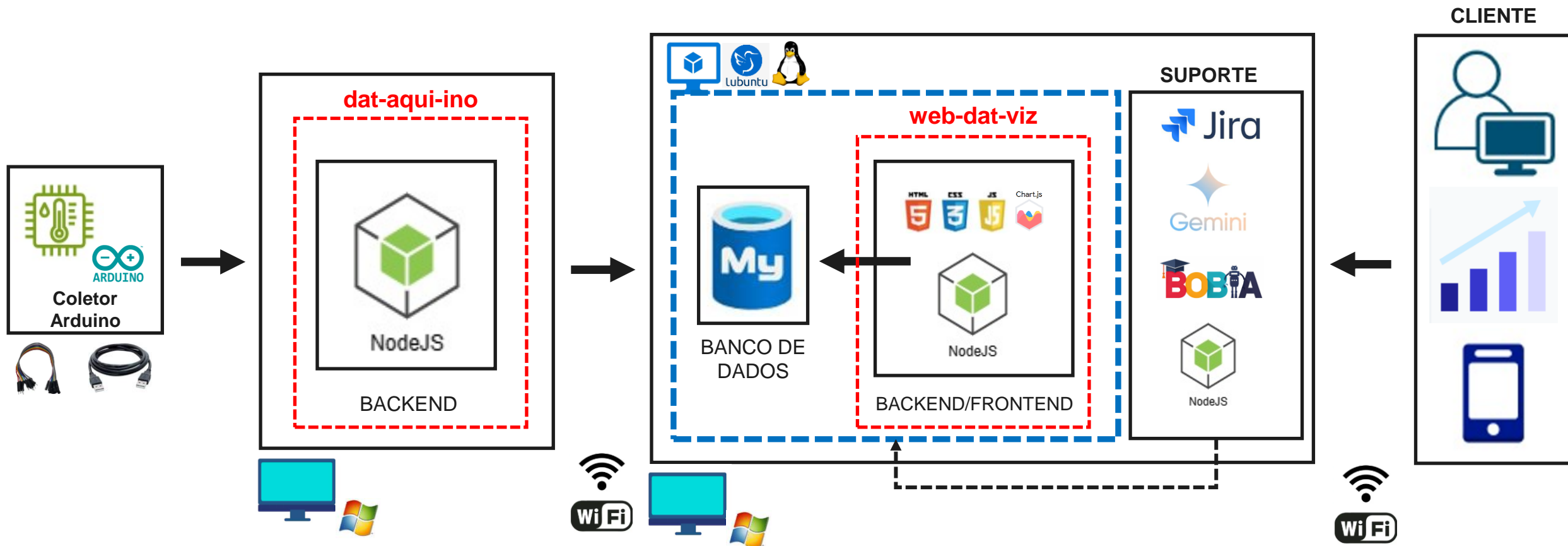
Governança

Projeto individual - Validação

Enviar por e-mail – Monteiro+Letícia

DIAGRAMA DE SOLUÇÃO

Diagrama de Solução - Tecnologia



CICLO DE DEV. DE SW

CICLO DE DESENV. DE SW

7 Stages of Software Product Development



O ciclo de vida do desenvolvimento de software é um **processo sequencial** que define as etapas de criação e implantação de um software. Ele fornece uma **estrutura para gerenciar o projeto de software** de forma eficiente e eficaz, **garantindo** que o produto final atenda aos requisitos do cliente.



Uma vez que o desenvolvimento de sistemas deve ser concluído dentro do tempo e custo pré-definidos, o Ciclo de Vida do Desenvolvimento de Software (SDLC) consiste num plano detalhado.

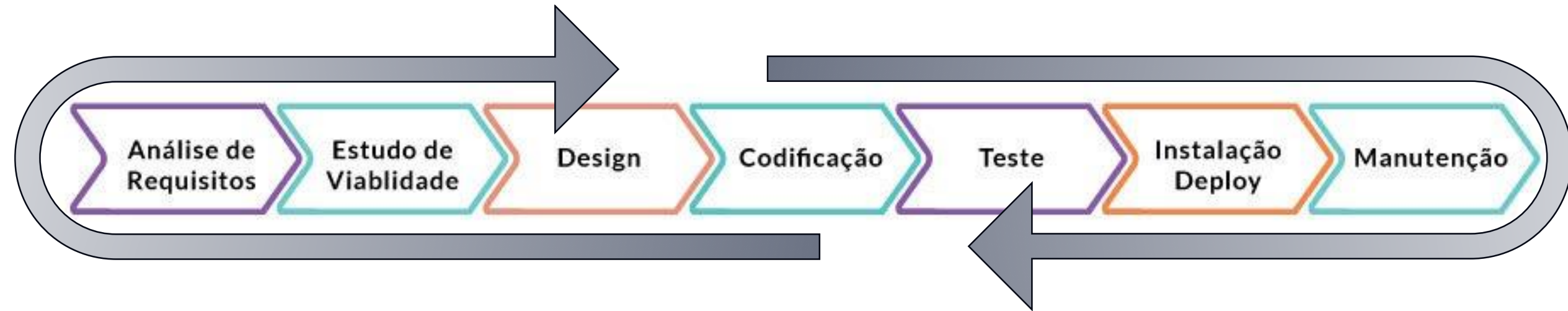
Ele explica como planejar, construir e manter software específico.

O Software Development Lifecycle (SDLC, em inglês) – tem como objetivo produzir software de alta qualidade.

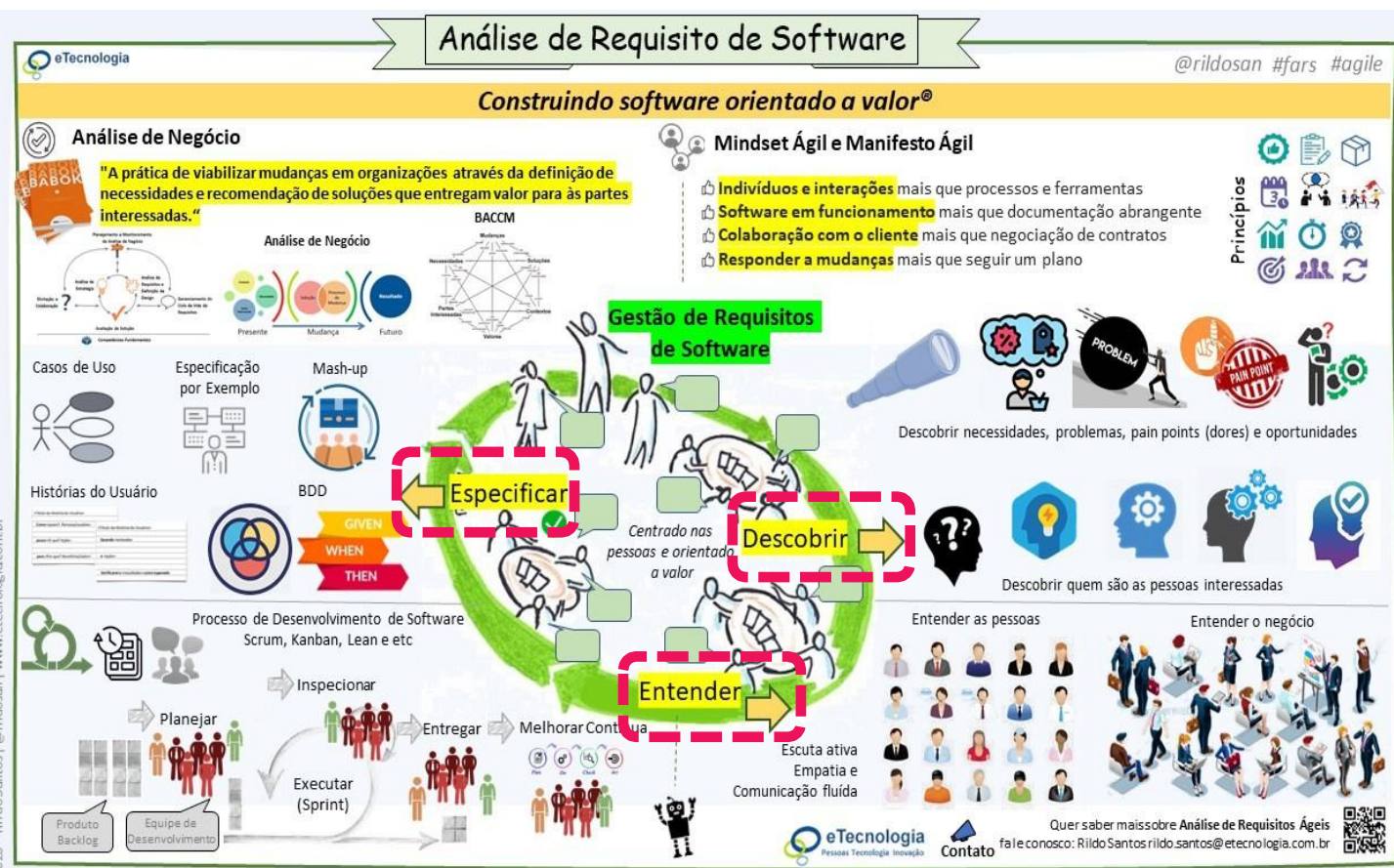
O Ciclo de Desenvolvimento de Software é um modelo composto por **sete fases**: Análise de Requisitos, Estudo de Viabilidade, Design, Codificação, Teste, Instalação, Deploy e Manutenção.



CICLO DE DESENV. DE SW



Análise de Requisitos



Esta primeira fase do **Ciclo de Vida de Desenvolvimento de Software** é uma visão geral e das diretrizes do projeto/software.

Análise de Requisitos

Todas as partes interessadas – incluindo clientes, vendedores, especialistas do setor, desenvolvedores de software, analistas de negócios e gerentes de projeto -, devem colaborar para juntar as informações necessárias sobre o que será construído.



Lembre-se de ser preciso ao descrever os requisitos; quanto mais detalhado for, melhor.

Análise de Requisitos

Checklist de informações essenciais para o cumprimento dos requisitos:

1. Perfil do usuário e como ele/ela se comporta ao usar sua solução
2. Feedback, pesquisas, entrevistas, questionários, testes e muito mais.
3. **Escopo e propósito do produto (problemas que seu software deve resolver)**
4. Listagem de todos os riscos envolvidos
5. Planejamento de cronogramas e calendários
6. **Pontos fortes e fracos do sistema atual, tendo como objetivo melhoria (SWOT)**
7. Custo e recursos necessários para implementação e lançamento
8. Equipes do projeto e estrutura de liderança.
9. Após essa etapa faremos a **Especificação de Requisitos de Software** para orientar o processo de desenvolvimento de software.

404

Houston, we have a problem.

Actually, the page you are looking for
does not exist. [Return home.](#)



Análise de
Requisitos

Estudo de
Viabilidade

Um estudo de viabilidade **é uma imagem clara do projeto**. Esta etapa do SDLC é uma das mais importantes e às vezes pode ser executada simultaneamente à primeira etapa.

É importante que todas as partes interessadas saibam exatamente todo o **contexto econômico, técnico, jurídico e de programação do projeto**, porque isso pode alterar escopo ou demonstrar se o software vai funcionar ou não.



Análise de
Requisitos

Estudo de
Viabilidade

É por isso que a análise como um estudo de viabilidade desempenha um papel relevante no Software Development Life Cycle. Durante o estudo de viabilidade, considere incluir informações como:

1. Descrição do produto ou serviço
2. Demonstrativos contábeis
3. Detalhes de operação e gerenciamento
4. Pesquisa e política de marketing
5. Dados financeiros e obrigações fiscais
6. Requerimentos legais
7. Plano de implementação do projeto
8. Tempo e orçamento disponíveis

VIABILIDADE TÉCNICA

VIABILIDADE ECÔNOMICA

VIABILIDADE OPERACIONAL

VIABILIDADE JURÍDICA/LEGAL

404

Houston, we have a problem.

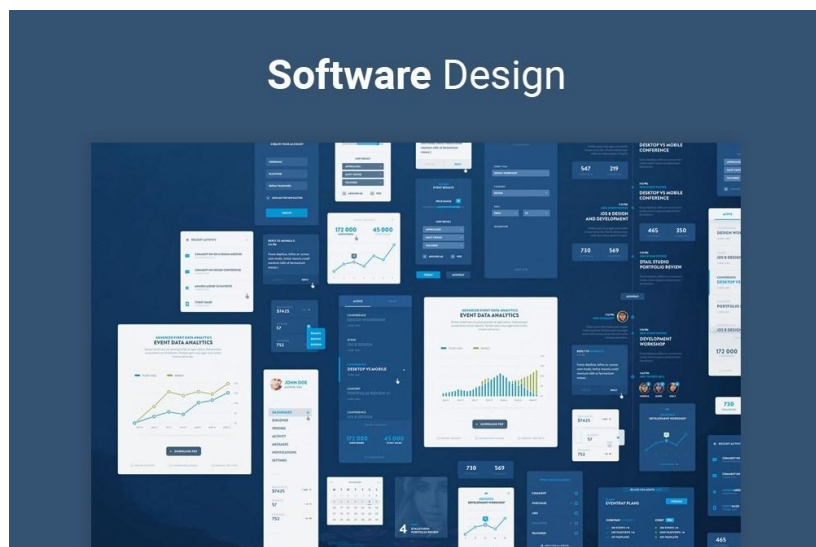
Actually, the page you are looking for
does not exist. [Return home.](#)



Análise de
Requisitos

Estudo de
Viabilidade

Design



É hora de projetar! Nesta etapa do SDLC, a equipe produzirá a DDS (Design Document Specification – Especificações de Documentação do Projeto) com base nos requisitos do usuário e na análise detalhada feita na fase anterior.

O documento DDS define a arquitetura geral do sistema e descreve todas as informações para os desenvolvedores começarem a trabalhar no produto, como recursos, input, output, bancos de dados, formulários, esquemas de códigos, especificações de processamento e tempo esperado para entregar o produto.

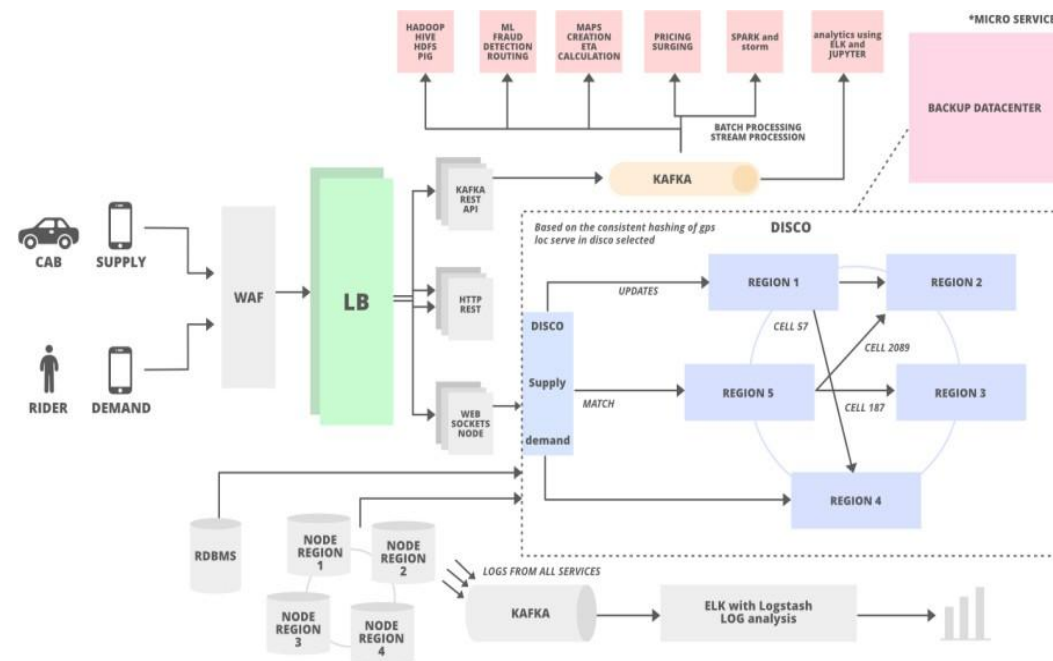
Análise de
Requisitos

Estudo de
Viabilidade

Design

Os documentos de design mais comuns usados nesta fase são Design de alto nível (**HLD – High-Level Design**) e Design de baixo nível (**LLD – Low-Level Design**).

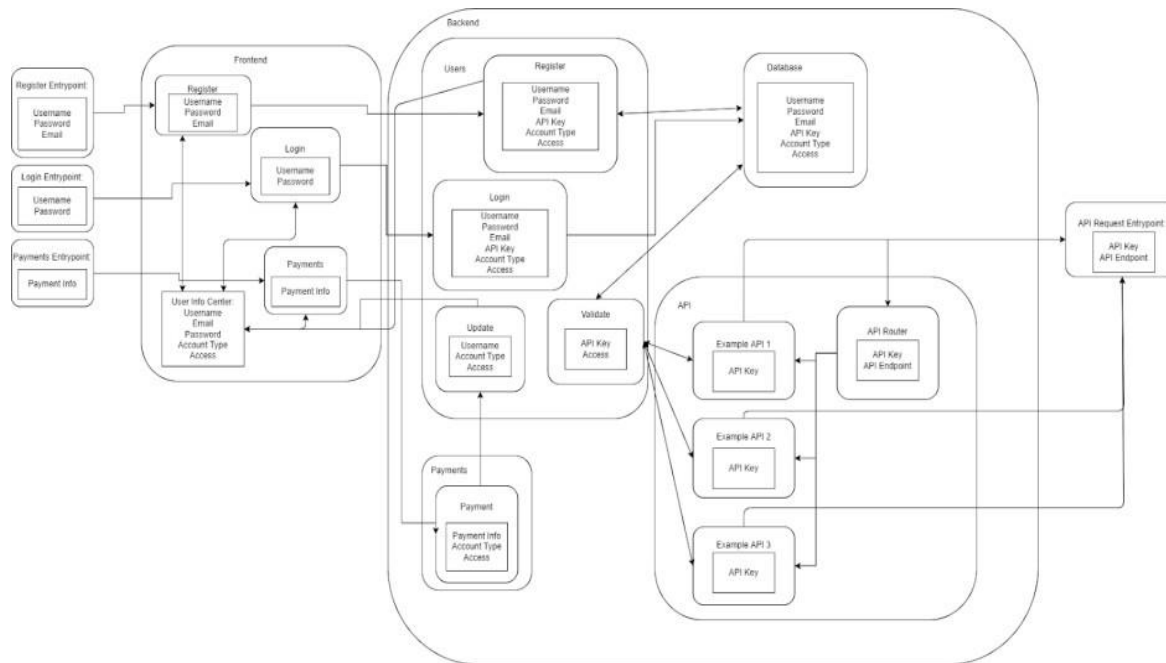
O High-Level Design (HLD) é uma breve descrição da funcionalidade de cada módulo e de como funcionará a relação de interface e dependências entre os módulos. Também inclui as tabelas de banco de dados, identificadas junto com seus elementos-chave, e os diagramas de arquitetura, junto com detalhes técnicos.



Análise de
Requisitos

Estudo de
Viabilidade

Design



O Low-Level Design (LLD) é um documento que descreve a lógica funcional dos módulos, tabelas de banco de dados (tipo e tamanho), detalhes da interface, tipos de problemas de dependência, lista de mensagens de erro e entradas e saídas para cada módulo.

Análise de
Requisitos

Estudo de
Viabilidade

Design

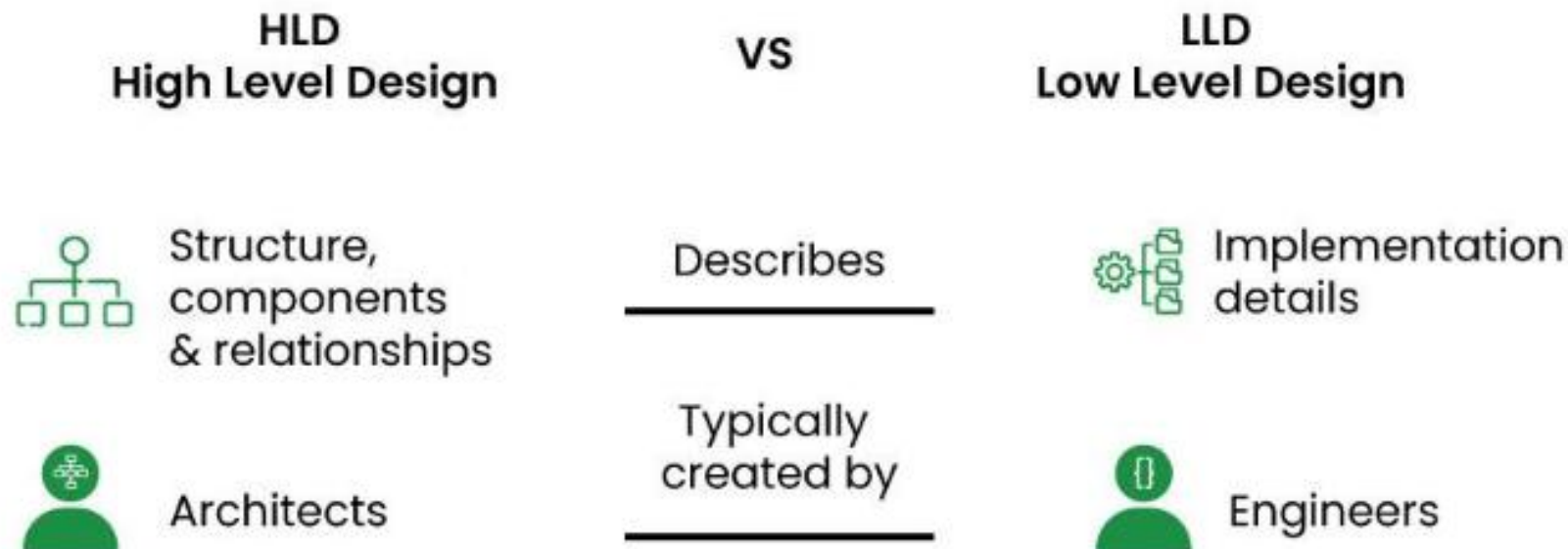
Nos dois tipos de documentos, é importante especificar detalhes sobre como a arquitetura deve ser construída, em termos de linguagem de programação, modelos ou padrões de projetos a seguir, etc, como é a comunicação entre o aplicativo com outros ativos e como os clientes devem interagir com a interface do usuário do software.

Além disso, será importante definir a plataforma **ou dispositivo no qual o software será executado (Mobile, Apps, Desktops, consoles de jogos, por exemplo)** e detalhes de programação, como métodos de resolução de problemas, realização de tarefas no aplicativo e detalhes de segurança como criptografia de tráfego SSL, proteção de senha e armazenamento seguro de credenciais de usuário.

Depois disso, começa a fase de prototipagem, que demonstrará uma ideia básica de como o aplicativo se parece e funciona. Este protótipo será mostrado às partes interessadas a fim de coletar feedback para melhorar o produto antes da fase de codificação.

CICLO DE DESENV. DE SW – DESIGN

High Level Design vs Low Level Design



Difference between High Level Design and Low Level Design



Design de Alto Nível é como fazer um grande plano para o software e ajuda a encontrar problemas antecipadamente, para que a qualidade do software possa ser melhor garantida.

Design de Baixo Nível é bem documentado, fica mais fácil para outros verificarem o código e garantirem sua qualidade durante a própria escrita do software.

CICLO DE DESENV. DE SW – CODIFICAÇÃO

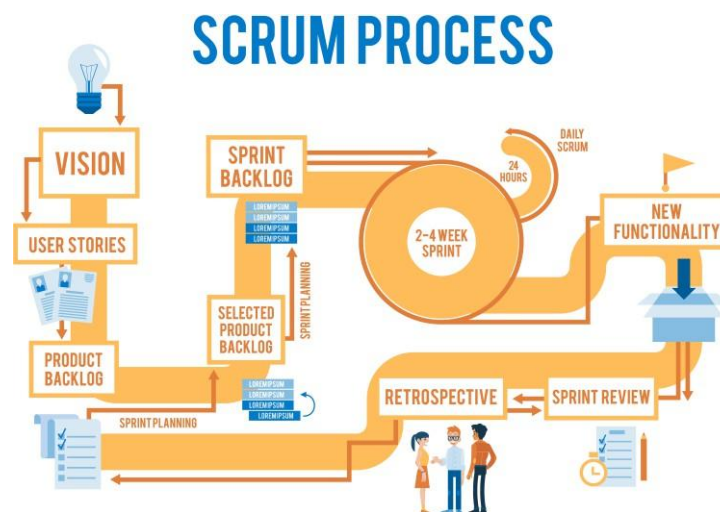
Análise de
Requisitos

Estudo de
Viabilidade

Design

Codificação

A próxima fase do SDLC é o Desenvolvimento, que também é a **mais longa**. Seguindo o DDS e as diretrizes de desenvolvimento, os desenvolvedores vão traduzi-los em código-fonte e linguagem de programação. Todos os componentes do software são implementados nesta fase.



404

Houston, we have a problem.

Actually, the page you are looking for
does not exist. [Return home.](#)



Análise de
Requisitos

Estudo de
Viabilidade

Design

Codificação

Teste

A próxima fase do SDLC é testar o produto desenvolvido. Pesquisas têm mostrado que o processo de teste frequentemente é **responsável por 40% do custo de desenvolvimento de software**. Com a crescente necessidade de alta qualidade e eficiência, é cada vez mais importante que as organizações aprimorem seus testes de software.

A principal função do teste de software é detectar bugs para descobri-los e detectá-los. O escopo do teste de software inclui a execução desse código em vários ambientes e também o exame dos aspectos do código.

Análise de
Requisitos

Estudo de
Viabilidade

Design

Codificação

Teste

TESTES

DEV e QA – Quality Assurance

- Teste unitário (dev)
- Teste integrado
- Teste de Performance
- Teste de Carga
- Teste de Stress
- Teste de Segurança
- PLANO DE VALIDAÇÃO (cenários e evidências)



HOMOLOGAÇÃO

(Cliente / QA contratado pelo cliente)

- PLANO DE VALIDAÇÃO
 - Validação dos Cenários de negócio
 - Validação de Performance
 - Teste de Carga
 - Teste de Stress
 - Teste de Segurança

CICLO DE DESENV. DE SW – INSTALAÇÃO & DEPLOY

Análise de
Requisitos

Estudo de
Viabilidade

Design

Codificação

Teste

Instalação
Deploy

A fase de lançamento e implantação concentra-se em observar **como o mercado reage ao seu produto**. É hora de lançar a versão final do software após os testes!

Durante a preparação e os procedimentos para a fase de lançamento, a equipe estabelece um procedimento operacional para organizar como o software deve funcionar no ambiente de TI e fornecer um plano de mitigação para apoiar o usuário final no reparo do problema.

Depois disso, é hora de programar cada parte do sistema. Esta fase inclui enviar o programa e programar cada site regional e cada sistema de computador.

Depois que sua equipe **implantar o aplicativo e entregá-lo aos usuários**, fique atento ao feedback e verifique se há problemas de implantação e o que precisam ser melhorados, **de acordo com a expectativa do cliente**.

Durante a fase de implantação, não se esqueça de identificar as principais equipes e funções envolvidas, como migrações e atualizações de software e configuração de permissões e funções de acesso, e tente limitar o impacto de quaisquer **problemas de configuração inicial usando projetos- piloto**.

CICLO DE DESENV. DE SW – MANUTENÇÃO

Análise de
Requisitos

Estudo de
Viabilidade

Design

Codificação

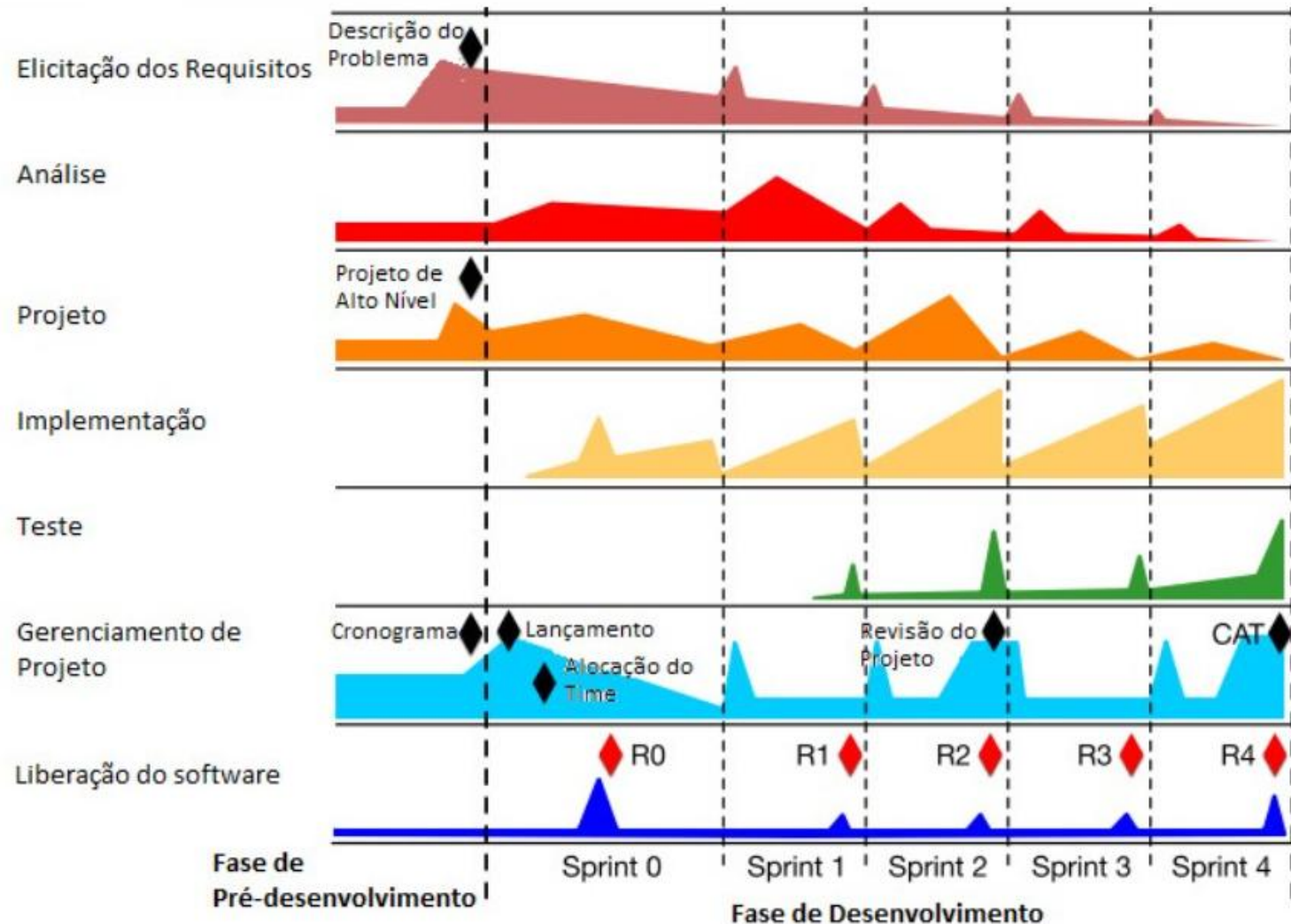
Teste

Instalação
Deploy

Manutenção

A fase de manutenção envolve **correção de bugs**, atualização do aplicativo para as versões mais recentes do software e aprimoramento, adicionando algumas novas especificações mencionadas na primeira fase. Nesse ponto, o ciclo de desenvolvimento está encerrado.





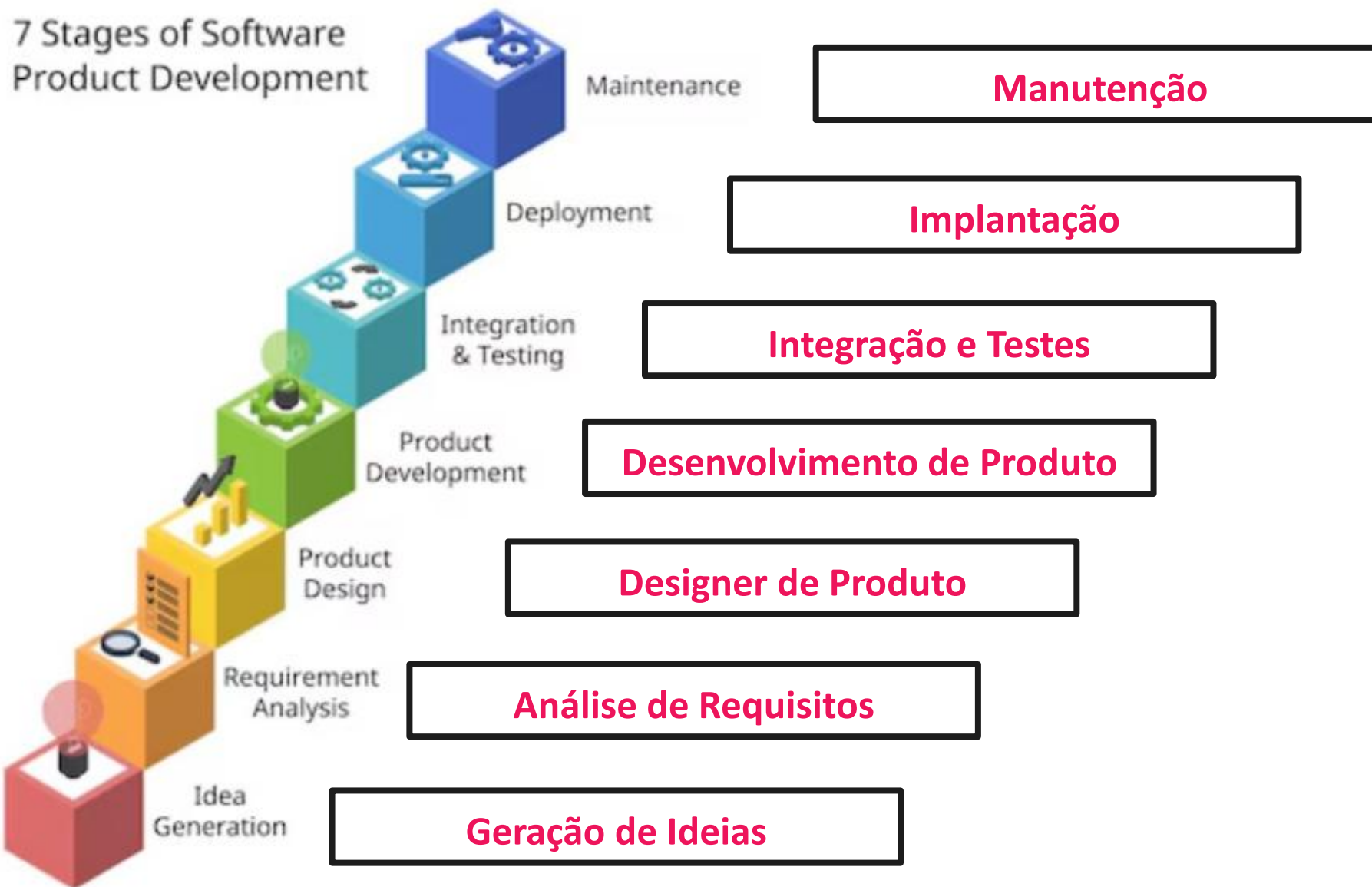
Fases da Sprint

Fase de Pré-desenvolvimento focada em Descrição do problema e Descrição de requisitos.

Todas as Sprints possuem a Liberação do SW para gerar garantir o entendimento do produto e validação do cliente –

Se tiver algum erro a correção é rápida.

CICLO DE DESENV. DE SW



**Processo Contínuo a
cada entrega.**

A cada versão entregue ao
cliente (V1/V2/V3...)

Deve se fazer a atualização da
documentação.



GOVERNANÇA DE TI

Governança: Garantir que políticas e estratégia sejam realmente implementadas e que os processos requeridos estejam sendo corretamente seguidos. **Governança inclui definir papéis e responsabilidades, medir e relatar, e tomar as ações para resolver quaisquer questões identificadas.** (ITIL Glossary of Terms, 2011)

Governança de TI: A especificação das normas e políticas e principalmente o framework de **responsabilidades** para estimular **comportamentos desejáveis** na utilização da TI. (Weill, 2006, pag. 8)



GOVERNANÇA DE TI

Estabelecer normas e políticas para o setor de tecnologia

Direcionar processos de TI

Promover a Transformação Digital

Estabelece objetivos estratégicos para a TI

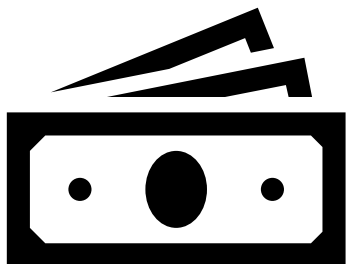
Monitorar o cumprimento das regras estabelecidas

POR QUE UMA EMPRESA PRECISA DE GOVERNANÇA DE TI?



Simples! Quando apenas existiam cavalos nas ruas, não nos preocupávamos com semáforos.





POR QUE UMA EMPRESA PRECISA DE GOVERNANÇA DE TI?

- Falhas em TI **podem causar impactos e colocar em risco a credibilidade** da empresa com os consumidores e investidores;
- Um acidente ou erro em TI pode afetar a **segurança da informação e confidencialidade de processos**;
- Diferentes projetos de TI sem um alinhamento estratégico e procedimentos para seguir podem trazer **prejuízos financeiros para um negócio**.

A Governança de TI entra em campo para solucionar problemas de **alinhamento organizacional em relação a TI e às estratégias do negócio.**



Os principais **INDICADORES** de uma Governança de TI bem sucedida são:

Alinhamento Estratégico

Entrega de valor

Gestão de risco

Gestão de recursos

Mensuração de desempenho



GOVERNANÇA DE TI

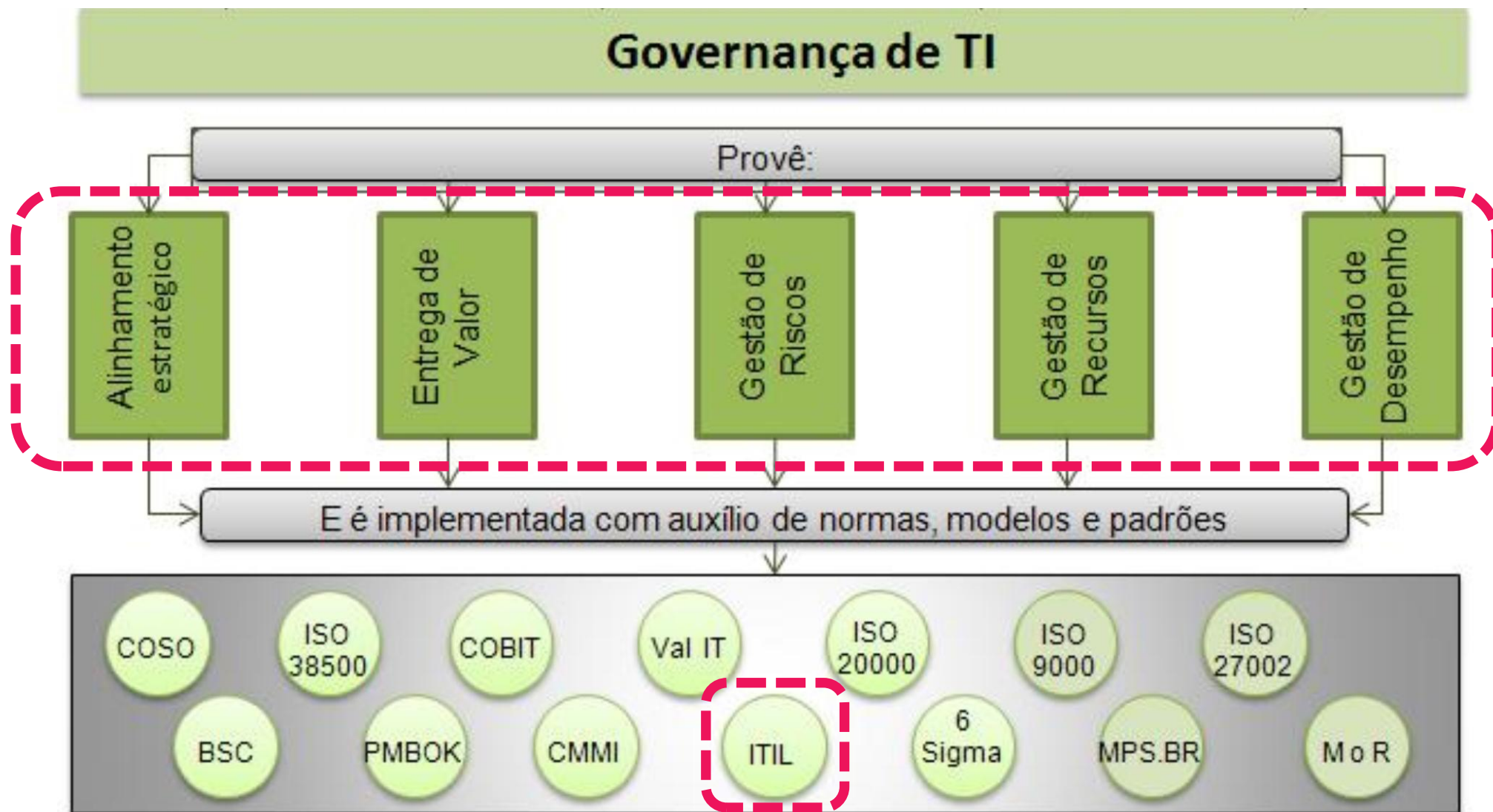


Imagem adaptada de: FEELY, Dan. Getting Governance Right. Info Source, v. VII, n. 3, dez 2007. Disponível em:
http://www.transforming.com/tsi_news/best_consulting_firm_newsletters_december07.html.
<http://transforming.com/2007/12/17/getting-governance-right/>

GOVERNANÇA DE TI - FRAMEWORKS

Os principais *frameworks* (conjunto de padrões, ferramentas e boas práticas) da Governança de TI são:

- Cobit (*Control Objectives for Information and related Technology*)
- **ITIL (*Information Technology Infrastructure Library*)**
- PMBOK (*Project Management Body of Knowledge*)
- Metodologias Ágeis (*SCRUM...*)
- Dentre outros...

Além da importância de **gerenciar os riscos** que o negócio está exposto e também **direcionar a tomada de decisões de forma assertiva**.

O QUE GOVERNANÇA DE TI DEFINE:

- Qual o papel de negócio da TI?
- Como garantir o uso eficaz de TI?
- Quem são as pessoas que devem decidir?
- Como garantir que as decisões estão sendo tomadas de forma correta?
- Quais são os comportamentos desejáveis?
- Como o orçamento deve ser investido?
- Como garantir que os sistemas vão se integrar? Qual o padrão?
- Quais são os serviços de TI que devem ser compartilhados?
- Quais são os serviços de suporte?

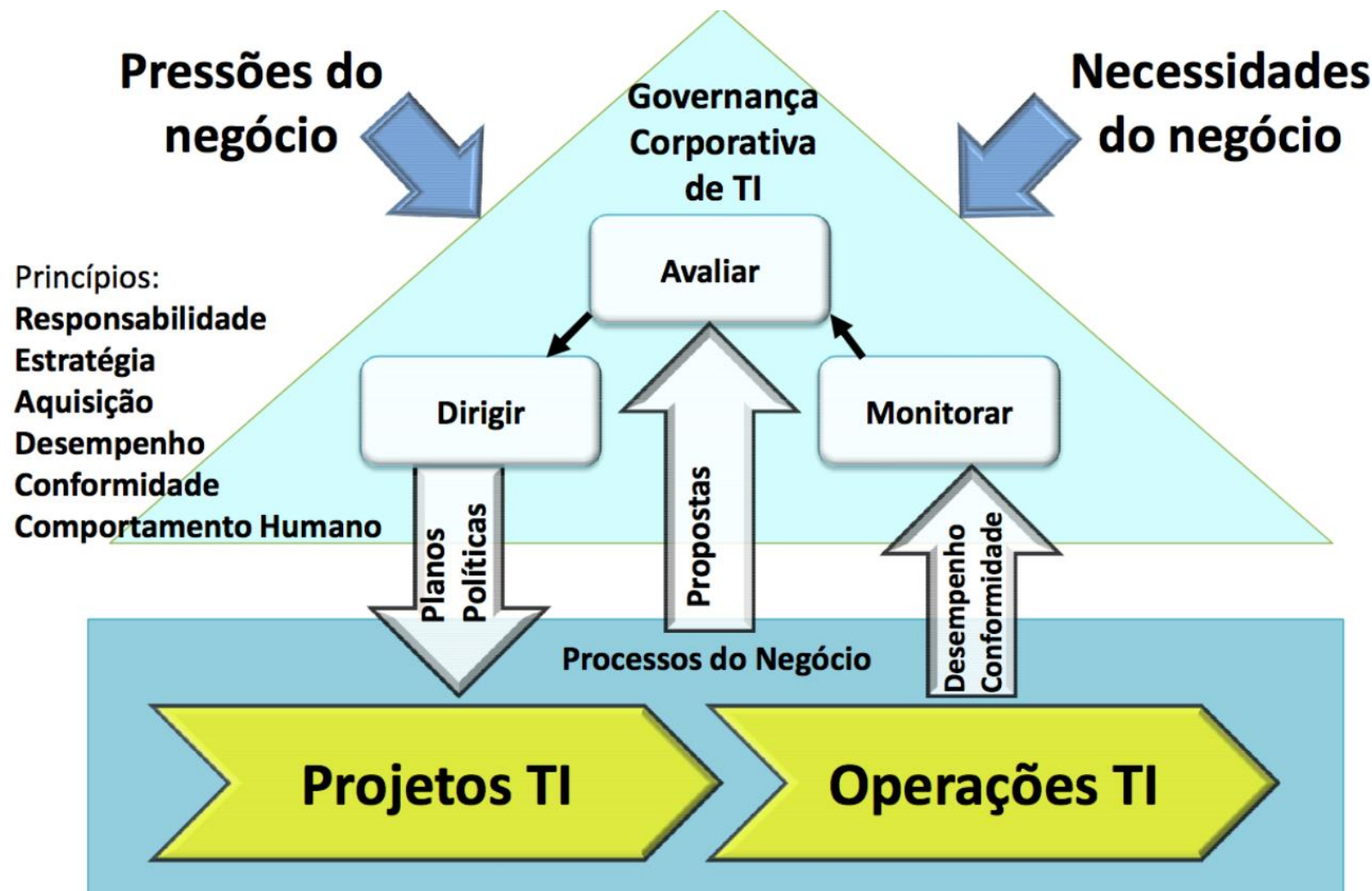
**AGREGAR VALOR
AO NEGÓCIO**

**CONTINUIDADE
DO NEGÓCIO**

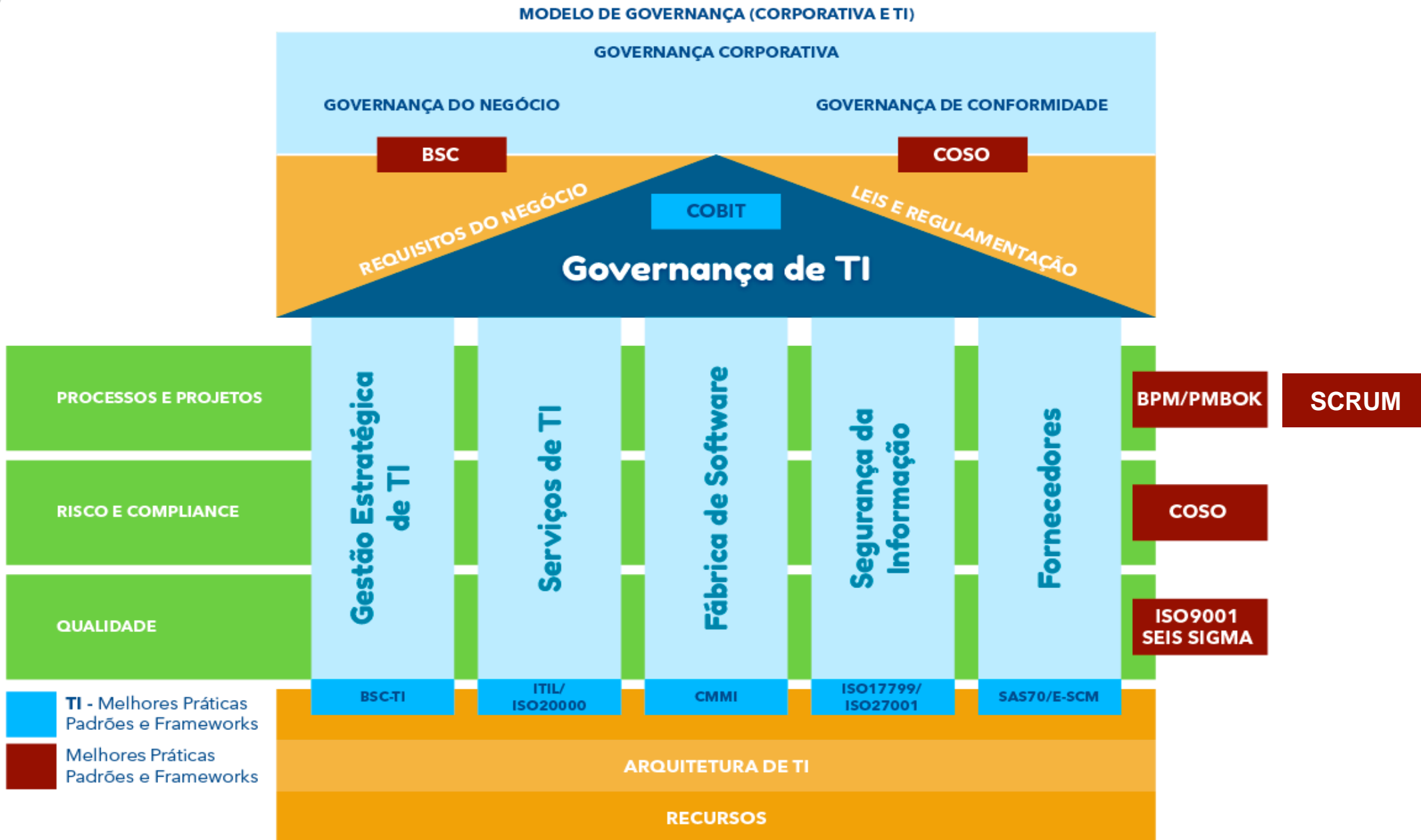


Para garantir um modelo de Governança de TI eficaz, é necessário ter o

envolvimento e apoio da alta gestão, com padrões estabelecidos e responsabilidades definidas para aplicação em toda a organização.



GUIAS DE BOAS PRÁTICAS



A **GOVERNANÇA DE TI** -

Precisa possuir participação ativa na organização da empresa.

NÃO PODE ser tratada como apenas mais uma operação do negócio, sim como um posicionamento estratégico.



Governança de TI X Empresas:

Qual(is) consequência(s) a falta de Governança de TI pode trazer para a empresa? E quais são os impactos?

Pesquisar um caso real de uma empresa que na sua visão, teve um problema de governança de TI (no caso, a falta dela) e os impactos gerados. Descrever o case detalhando os problemas relacionados à falta de governança e seus impactos.

- **Atividade Individual.**
- Subir no Moodle arquivo em formato **PDF**
- Nome do arquivo: Nome + RA + Atividade – **Ex: Marcos Antonio RA1225589 – Governança TI**
- **Prazo 14/05/2025 QUA – 23:59:59 (Hora do lobisomem)**

Agradeço a sua atenção!

RAFAEL PETRY
rafael.petry@sptech.school

MARCOS SANTOS
marcos.antonio@sptech.school

MONTEIRO
monteiro@sptech.school

FÁBIO FIGUEREDO
fabio.figueredo@sptech.school

SÃO
PAULO
TECH
SCHOOL