



SÃO  
PAULO  
TECH  
SCHOOL



# **TECNOLOGIA DA INFORMAÇÃO**

**Professores:**

**Rafael Petry – SIS, CCO e ADS-A**

**Marcos Antonio – ADS-B**

**Fernando Brandão – ADS-C**



# REVISÃO



**S3**

## Sustentação

- Governança
- ITIL (Incidentes, Problemas e Mudanças)
- Suporte de TI
- Fluxograma do suporte
- Ferramenta de Help Desk
- Documento de Mudança

Entrega: ~03/06/2024

**S2**

## Metodologia e P

- Metodologia d
- Projetos
- Introdução a F
- Desenv. De S'
- Arquitetura de



grama da Solução  
ilha Product Backlog  
ilha Sprint Backlog

Entrega: ~29/04/2024

**S1**

## Introdução + Pla

- Introdução a T
- Projeto vs Pro
- Requisitos
- Documentaçã
- Requisitos na Ferramenta
- Ferramenta de Gestão
- Documentação do Projeto



Entrega: ~18/03/2024

**SP3 – Aulas 1**

**Prova AC2**

**Ciclo de Desenvolvimento de SW**

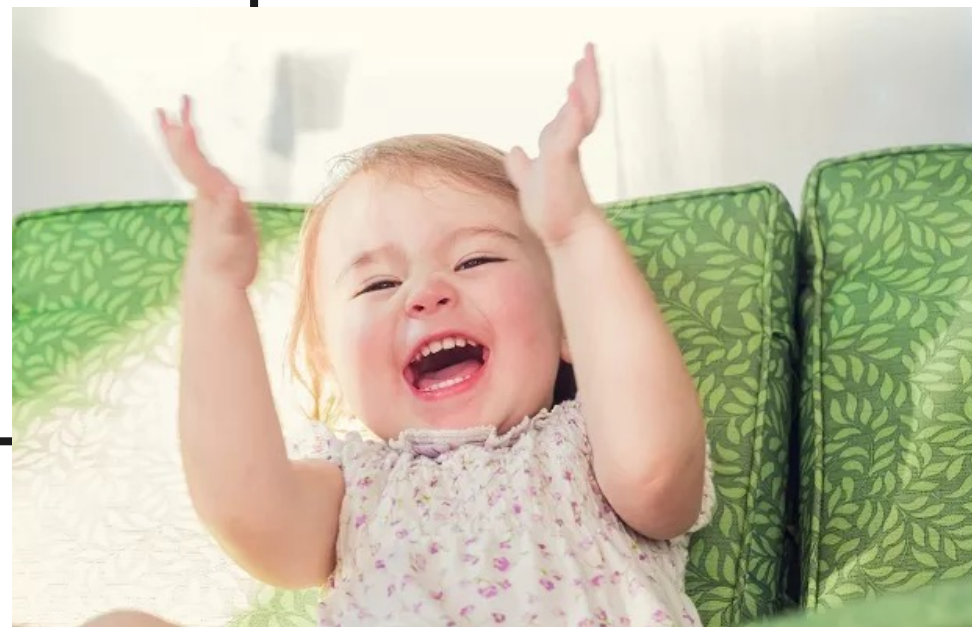
# PROVA AC2

65 ALUNO FIZERAM A PROVA

MÉDIA 7,90

63 ACIMA DE 6,00

2 ABAIXO DE 6,00



NOTA MAIS ALTA 10

Bruno Yuji Takahashi



**Aluno**   
**Nota 10!**



# PROVA AC2

## Pra que serve o Diagrama de Solução:

I - Mostrar ao cliente a complexidade dos projetos;

II - O DSN (Desenho de Solução de Negócio) para dar uma visão ao pessoal mais técnico e o Diagrama de Solução Técnico para as equipes mais ligadas ao negócio;

**III - Servir como documentação usada durante e após a implementação do Projeto, apresentando a visão geral, entendimento e organização dos elementos técnicos.;**

IV - Ter um relatório dos requisitos Funcionais do projeto,

A I está correta

B I e a IV estão corretas

C II está Correta

**D III está correta**

## A Arquitetura de TI é:

A Uma função da TI para analisar e reduzir custos da empresa.

B Atribuição da área de Negócio responsável por analisar as necessidades de TI e os espaços de instalação de uma empresa para planejar, estruturar estrategicamente a alocação e distribuição dos ativos tecnológicos.

**C Atribuição da área de TI responsável por analisar as necessidades de negócio e os espaços de instalação de uma empresa para planejar, estruturar estrategicamente a alocação e distribuição dos ativos tecnológicos.**

D Uma atribuição da área de TI para atender as demandas de infraestrutura de clientes de acordo com os projetos.

**"Processo de dividir um servidor físico em vários servidores virtuais, únicos e isolados, por meio de um aplicativo de software." Esta definição pertence ao conceito de:**

A Cloud Computing

**B Virtualização**

C Data Centers

D SAAS, PAAS e IAAS

**Seu gerente está preocupado com o armazenamento e perda das informações de RH que são fundamentais para a empresa. Qual processo você indicaria para que ele fique tranquilo em relação as informações:**

A Aumentar a capacidade de armazenamento dos servidores.

B Política de atualização de software e Anti-vírus.

**C Plano de Backup das informações armazenadas.**

D Avaliar o tipo de contratação do serviço: IAAS, SAAS ou PAAS

**Em uma arquitetura onde temos o Sistema Operacional, Banco de Dados, Aplicação e Middleware na nossa gestão e pagamos como serviço todo restante da infraestrutura:**

**I - Temos um IAAS pois todo hardware será de responsabilidade da empresa contratada;**

**II - Temos um IAAS com PAAS**

**III - Temos serviços On-Premises com custos sob responsabilidade do contratante**

**IV - Temos um SAAS com custo compartilhado**

**A Apenas a I está correta**

B A I e a II estão Corretas

C Apenas III está correta

D Apenas IV está correta

**Numa estrutura organizada de elementos de TI, por exemplo, Datacenter, como é chamado o conjunto de processos para "cuidar" deste ambiente, para que tudo siga funcionando conforme planejado e siga alinhado com as estratégias e objetivos do negócio?**

## **A Governança de TI**

B Processos de Desenvolvimento de Software

C Metodologia Ágil ou Cascata

D Política de Backup e Segurança da Informação

E Política de Segurança e Disponibilidade

**O exemplo abaixo que mais se assemelha ao modelo SAAS de comercialização de serviços em TI é:**

A Delivery de Pizza, onde só precisamos da mesa, cadeiras, do refrigerante e talheres para consumir

B Compra de servidores e instalação em espaço apropriado dentro de casa ou escritório

C Assinatura de sites de jogos para download e uso no videogame

**D Assinatura de sites de conteúdo como Netflix, Amazon Prime Video, dentre outros**

E Compra de um carro



## A metodologia Ágil Scrum é importante pois:

A Elimina Riscos do Projeto

B As pessoas envolvidas no projeto controlam seus próprios cronogramas

**C O processo desta metodologia gera uma proximidade maior com cliente**

D Garante a arquitetura da solução que será entregue

**Quem normalmente tem mais relacionamento com os Clientes na Metodologia Ágil SCRUM:**

A Toda a equipe

B Desenvolvedores

C Gerente do Projeto

**D Product Owner**

## Quais destas funções fazem parte da Metodologia Ágil Scrum

A Scrum Master, Gerente de Projetos, Arquiteto de Solução, Product Owner e Team (ou Dev Team)

B Scrum Master, Gerente de Projetos, Product Owner e Team (ou Dev Team)

**C Scrum Master, Cliente, Team (ou Dev Team), Product Owner**

D Scrum Master, Team (ou Dev Team), Product Owner, Arquiteto de Solução

## Qual a melhor metodologia a ser usada em desenvolvimento de um Software?

A Scrum, pois é possível se concentrar na entrega de um produto de valor mínimo (MVP) e ir aprimorando a cada sprint

B Cascata, pois há interdependência clara de tarefas (não dá pra avançar até que a etapa anterior seja finalizada, exemplo, criação do banco de dados vs disponibilização do servidor de banco de dados) e é necessário ter um cronograma de entrega claro para o cliente

C Scrum, metodologia desenvolvida a partir de experiências inovadoras e "enxutas", evitando desperdício de recursos

**D Depende de como será o projeto, participação do cliente, cultura da empresa e maturidade da equipe e dos processos**

E Cascata pois é necessário preeditividade quanto as funcionalidades do escopo, além de foco total no produto final desejado

**Cite as principais habilidades que um Product Owner deve ter na metodologia Ágil Scrum:**

- I - Habilidade de relacionamento com o cliente;**
- II - Conhecimento do Negócio;**
- III - Dar treinamento da metodologia para toda a equipe;**
- IV - Escolha da tecnologia a ser utilizada no projeto;**
- V - Eliminar obstáculos para o time de DEV seguir com o desenvolvimento.**

A I e IV estão corretas

B A I, II e III estão corretas

C A II, III e IV estão erradas

**D Apenas a I e II estão corretas**

**Um dos principais objetivos da Metodologia Scrum é a entrega do MVP.**

**Entende-se por MVP:**

A Máximo de Valor do Produto, ou seja, o máximo possível de requisitos de valor na versão finalizada do produto.

**B Produto Mínimo Viável, uma primeira versão da entrega funcionando com requisitos de valor, mesmo que ainda para testes e feedback.**

C Técnica para elencar os principais requisitos de valor e seu tamanho, para compor a primeira sprint.

D Produto entregue na primeira Sprint

**Considerando o método ágil SCRUM, o Daily Scrum Meeting é :**

A Um brainstorming eventual para definir as possíveis atividades e alternativas que devem ser aprovadas pelo comitê do projeto.

B Uma reunião exclusiva para tratar do cronograma do projeto.

**C Uma reunião diária com toda a equipe para verificar as pendências desde a última reunião, as atividades que serão realizadas no dia, e os impeditivos existentes.**

D Uma reunião quinzenal entre o Scrum Master e o Product Owner para responder aos riscos do projeto.

**Qual item NÃO faz parte do enfoque do SCRUM ?**

A Colaboração

B Transparência

C Compartilhamento de conhecimento

**D Individualismo.**



**Veja abaixo algumas afirmações sobre as características de um Product Backlog:**

- 1. O Product Backlog deve ser definido entre o Product Owner e o Cliente;**
- 2. O Sprint é derivado apenas dos Requisitos Funcionais e Não Funcionais acordados com Cliente e o Scrum Master;**
- 3. Um Product Owner só pode controlar um Product Backlog;**
- 4. Os Requisitos de Negócio são descritos no Product Backlog e o Scrum Master ajuda a definir as Sprints.**

**Sobre as afirmações acima, é correto dizer que :**

**A As afirmações 1, 3 e 4 estão corretas**

B As afirmações 3 e 4 estão corretas

C As afirmações 2 e 3 estão corretas

D Somente a afirmação 4 está correta

**Considerando o método ágil SCRUM, o Product Owner deve investir a maior parte do seu tempo em :**

A Preparar as reuniões diárias

B Dar apoio e ajuda exclusiva aos Devs

C Resolver impedimentos e fazer Coaching com a equipe

**D Cuidar do relacionamento com os clientes dos projetos para entender o Requisitos e construir o Product Backlog**

**Em um projeto onde a data de entrega do primeiro Sprint esta em risco de atraso você, como Product Owner, tomaria qual decisão para minimizar impactos com cliente:**

**A Voltaria ao cliente e tentaria renegociar as entregas do Sprint**

B Colocaria mais DEVS sem conversar com cliente, para não gerar atrito.

C Negociaria com a equipe fazer hora extra

D Quebraria o Sprint em dois

## Qual a melhor definição para a Retrospectiva da Sprint?

- I - Momento para DEVs e SCRUM MASTERS entrarem em um acordo quanto as melhores linguagens para desenvolvimento;**
- II - Momento para estudar novas tecnologias a serem empregadas nos próximos sprints;**
- III - Momento para aprender com erros e melhorar processos para os próximos sprints ;**
- IV - Momento para preparar o Cronograma das próximas entregas.**

A A I e a II estão corretas

B A II e a IV estão corretas

**C Somente a III esta correta**

D Somente a I esta incorreta

**Um cronograma com a data final do projeto, no Scrum poderá ser definido:**

A A partir do planejamento (antes da primeira sprint), em reunião com dev team, PO e Scrum Master

B A partir do planejamento após a primeira sprint, após conhecer as competências técnicas do time

C A partir do conhecimento das competências técnicas e limitações do time

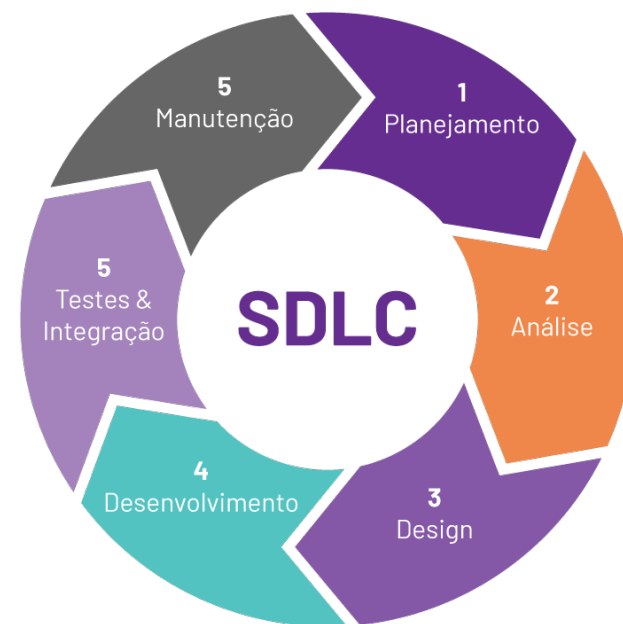
**D A partir da execução de algumas sprints, após o Scrum Master medir a velocidade de entrega de pontos a cada sprint**



# CHAMADA

# CICLO DE DEV. DE SW

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.







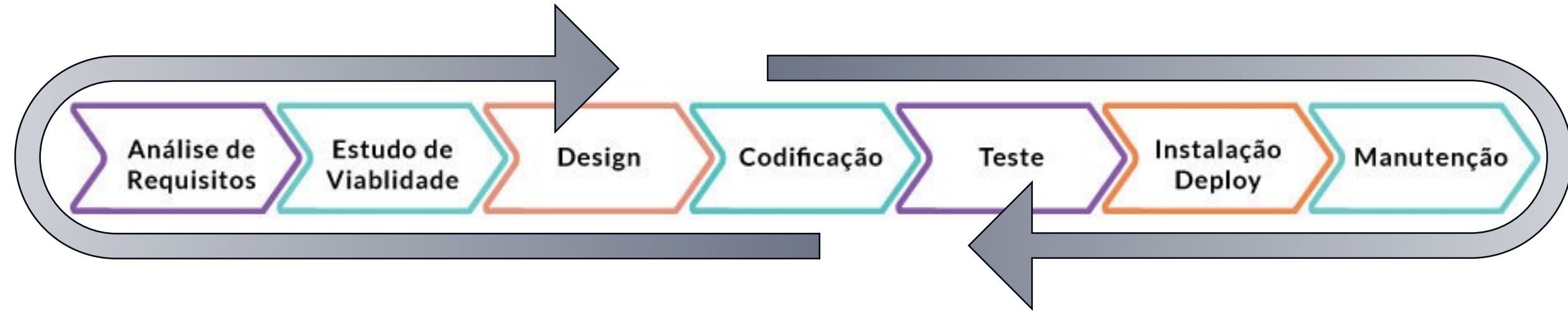
Uma vez que o **desenvolvimento de sistemas deve ser concluído dentro do tempo e custo pré-definidos**, o **Ciclo de Vida do Desenvolvimento de Software (SDLC)** consiste num plano detalhado.

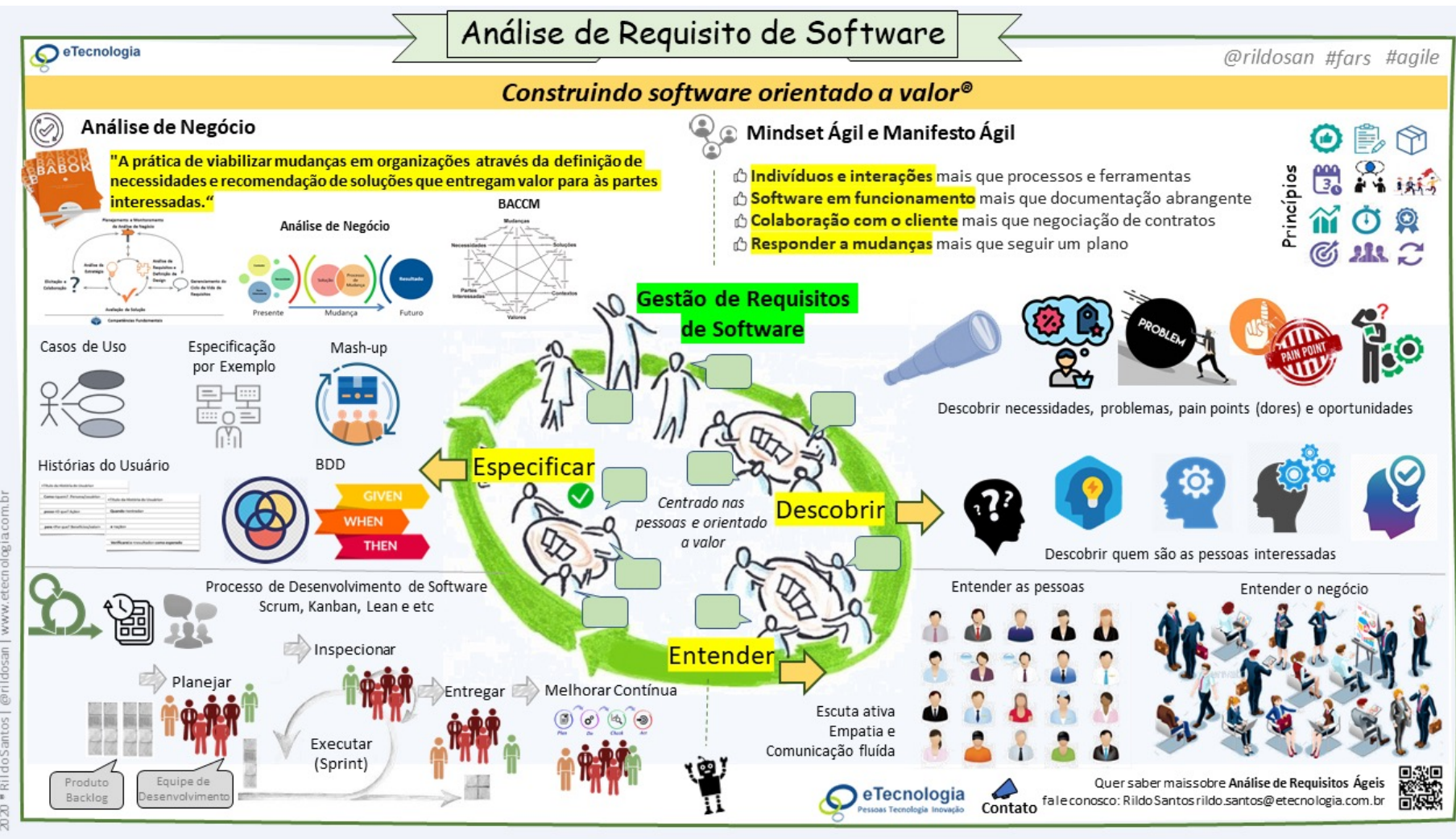
Ele explica como planejar, construir e manter software específico. O Software Development Lifecycle (SDLC, em inglês) – tem como objetivo produzir software de alta qualidade.

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





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





**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.





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.](#)





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.







É por isso que a análise como um estudo de viabilidade desempenha um papel relevante no Software Development LifeCycle. 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.](#)





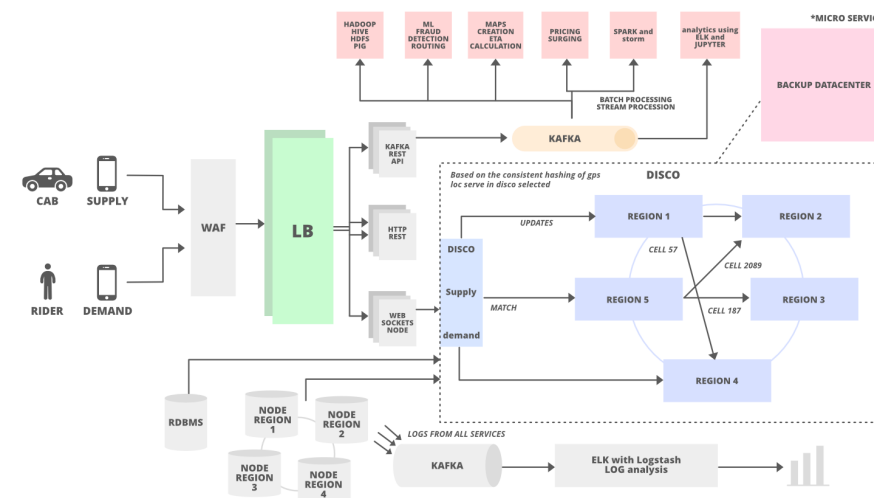
É hora de projetar! Nesta etapa do SDLC, a equipe produzirá a **DSS (Design Document Specification – Especificações de Documentação do Projeto)** com base nos requisitos do usuário e na análise detalhada feita na fase anterior.

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.



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  
RequisitosEstudo de  
Viabilidade

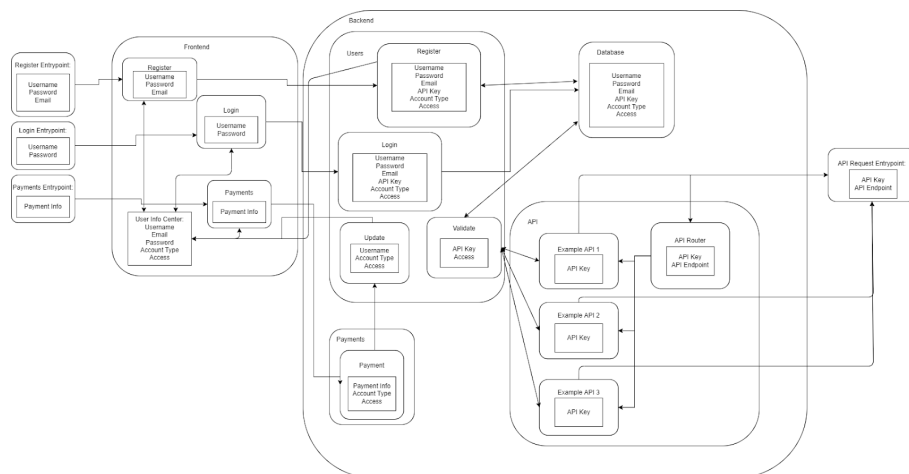
Design

Codificação

Teste

Instalação  
Deploy

Manutenção



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.



Nos dois tipos de documentos, é importante especificar detalhes sobre como a arquitetura deve ser construída, em termos de linguagem de programação, modelos ou boilerplates 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.

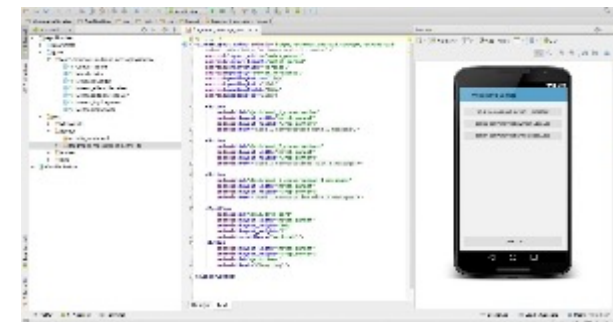
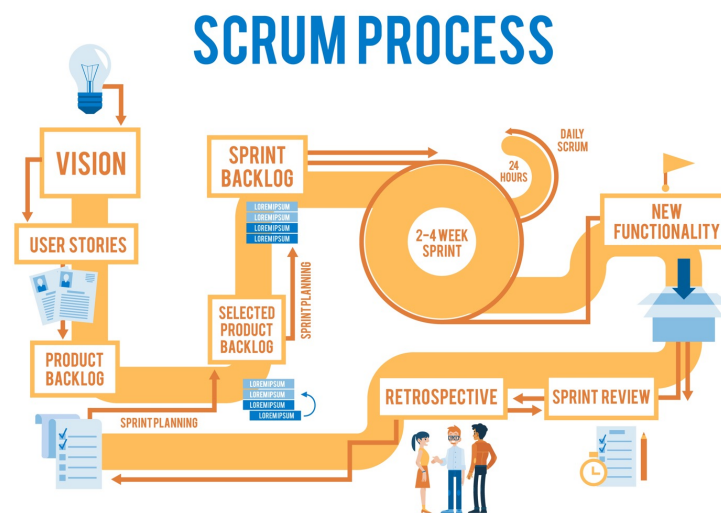
Depois disso, começa a fase de prototipagem, que demonstrará uma ideia básica de como o aplicativo se parece e funciona. Este protótipo será mostrado às partes interessadas a fim de coletar feedback para melhorar o produto antes da fase de codificação.



# CICLO DE DESENV. DE SW – 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 traduzí-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.](#)



# CICLO DE DESENV. DE SW – TESTES & HOMOLOGAÇÃO



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.



## TESTES

### DEV e QA – Quality Assurance

- Teste unitário (unidades)
- Teste integrado (módulos)
- Teste de Performance (Usuário)
- Teste de Carga (mundo real)
- Teste de Stress (limites)
- 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



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.

# CICLO DE DESENV. DE SW – INSTALAÇÃO & DEPLOY



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.

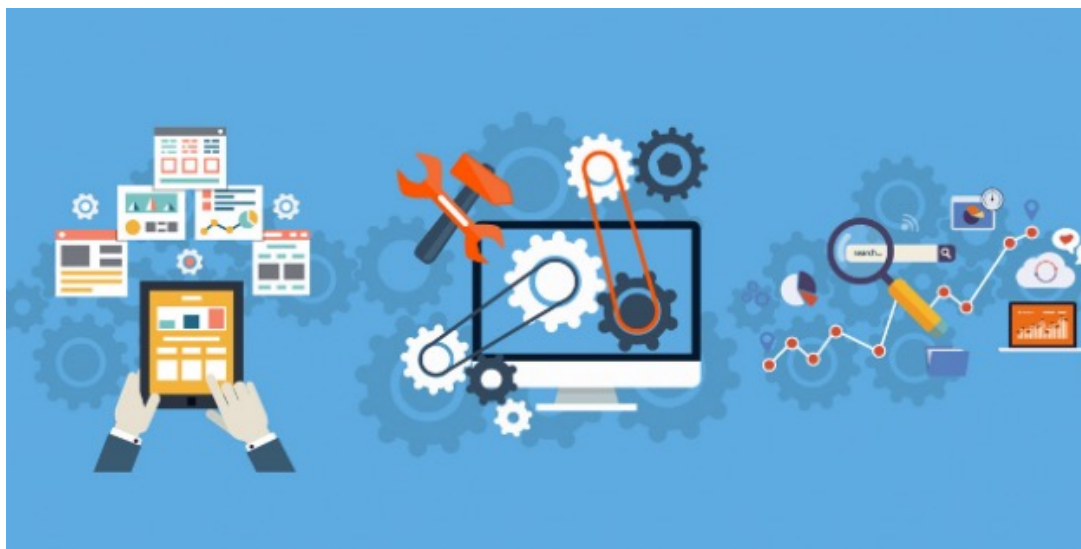
Depois que sua equipe **implantar o aplicativo e entregá-lo aos usuários**, fique **atento ao feedback** e verifique se há problemas de implantação e o que precisam ser melhorados, de acordo com a expectativa do cliente.

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.

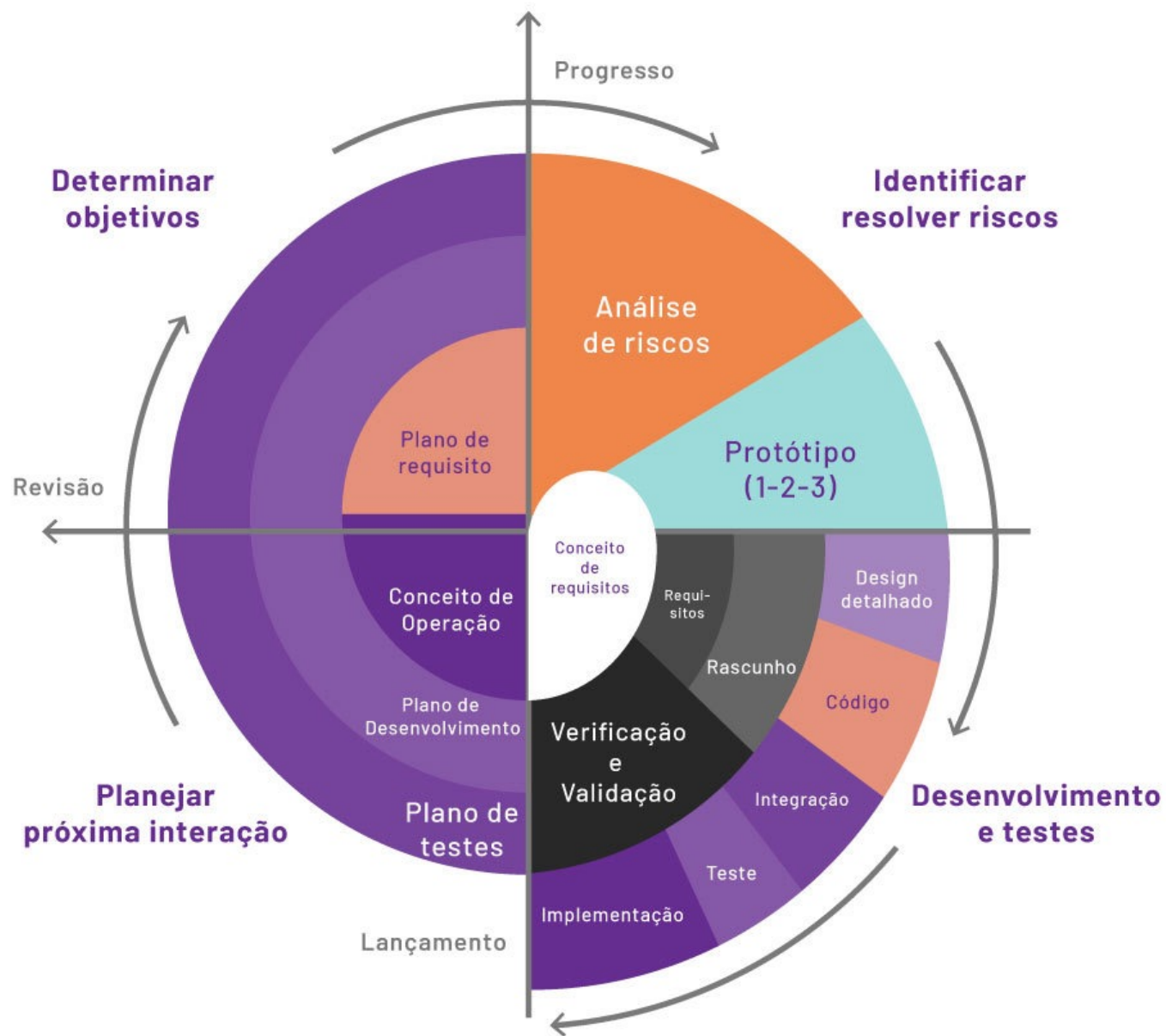
# CICLO DE DESENV. DE SW – MANUTENÇÃO



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.







# 404

---

**Houston, we have a problem.**

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





**Agradeço**  
a sua atenção!

**RAFAEL PETRY**

rafael.petry@sptech.school

SÃO  
PAULO  
TECH  
SCHOOL