

ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS - EXATAS

Laboratório de Software

Organização da Disciplina

Prof. Daniel Luis Notari dlnotari@ucs.br

Março - 2025

Índice

- Conceito de projeto temático/integrador
- Conceito de Design Thinking
- Etapas do Projeto
- Modelagem de Negócio

Projeto Temático / Integrador

- O que você entende por uma disciplina no formato Projeto Temático / Integrador?

Objetivo

- Planejar e implementar uma solução computacional que atenda os requisitos de um problema real considerando as etapas de levantamento das necessidades do software, concepção e elaboração da solução e sua implementação e validação.

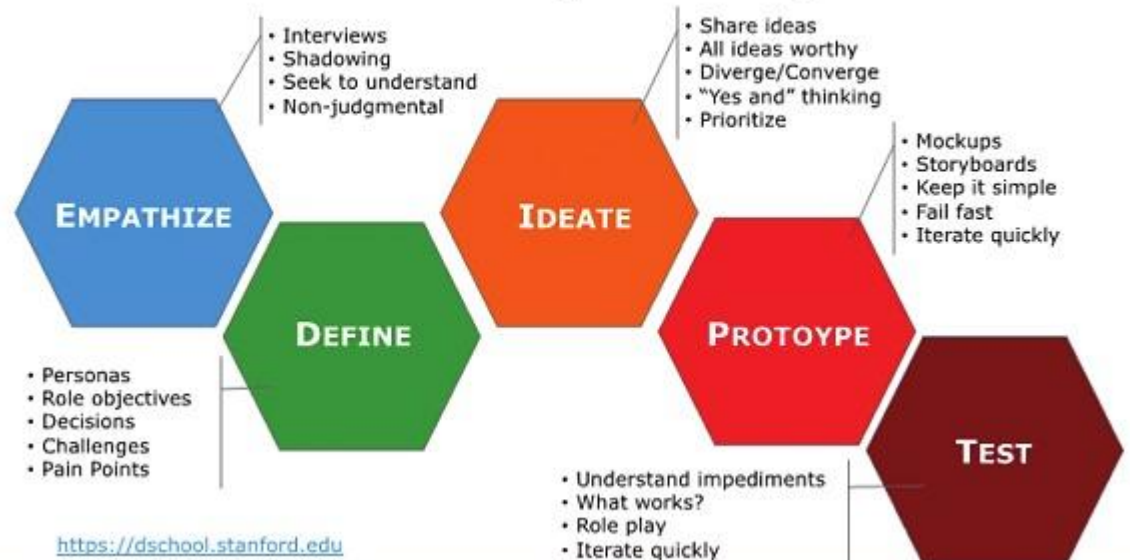
Competências

- Compreensão do problema e suas implicações
- Modelagem da aplicação computacional
- Implementação da aplicação computacional

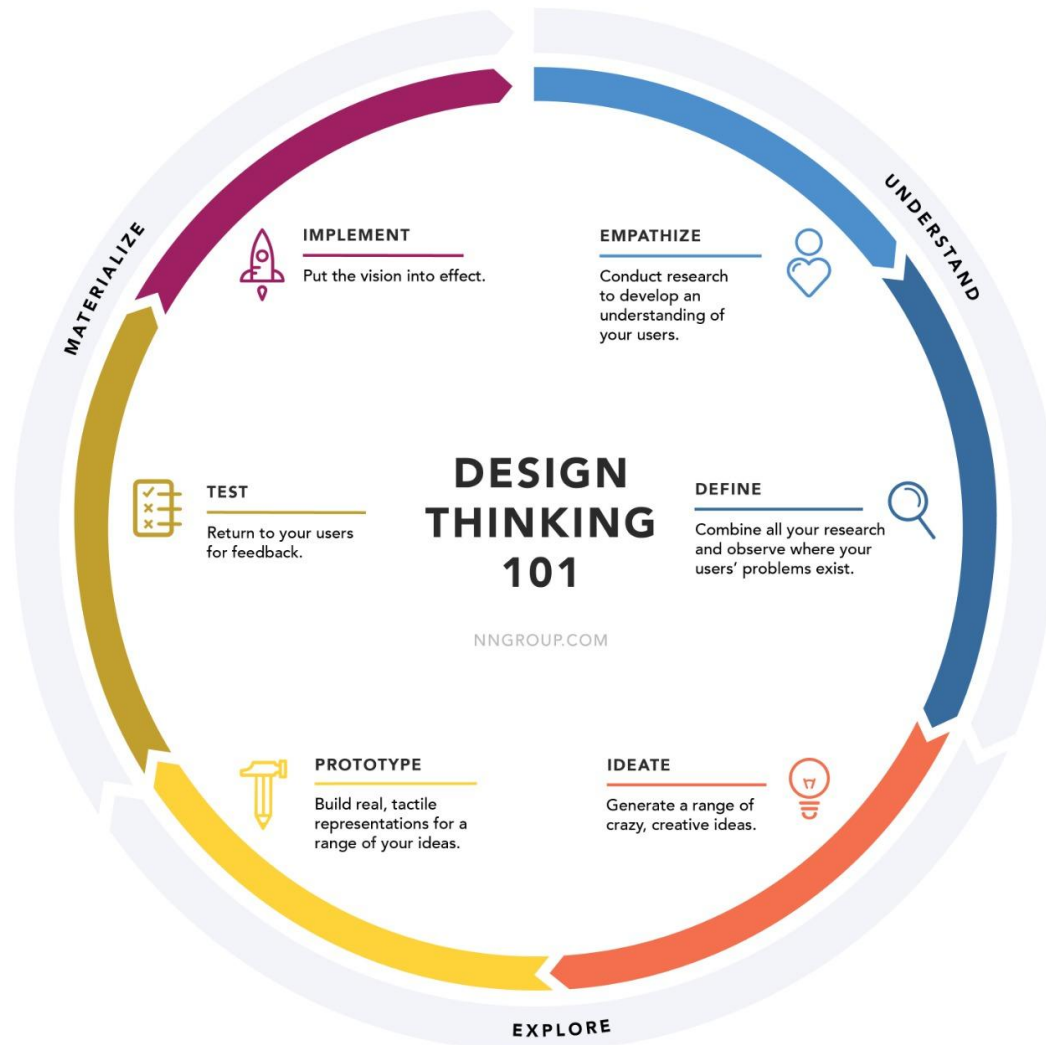
Design Thinking



Stanford d.school Design Thinking Process



Design Thinking



Projeto de Software

- Sucesso de um projeto depende:
 - Entendimento do problema
 - Correta especificação das necessidades
 - Investigação de como é feito hoje (se existir)
 - Investigação das possíveis soluções
 - Projetar e simular as soluções
 - Desenvolver e testar
- Fluxo de Informações

Projeto de Software

- Erros mais comuns
 - Aumentar o escopo do projeto mantendo o prazo e o custo iniciais
 - Permitir que o cliente atrase suas entregas
 - Querer ser perfeccionista em tudo
 - Cobrar um preço fixo pelo projeto
 - Não fazer um contrato
 - Querer entregar mais do que o combinado

Projeto de Software

Custo da descoberta do erro



Projeto de Software

- Custo da Descoberta de um erro
 - Quanto antes for descoberto, menos caro para consertar
- Como evitar a propagação de erros
 - Uso adequado de um processo de desenvolvimento
 - Entendimento correto do problema e das necessidades do cliente
 - Planejar e cumprir prazos
 - Especificação correta da necessidade e da solução
 - Comunicação entre as pessoas
 - Entregas curtas (ciclos de desenvolvimento pequenos)

Conteúdo Programático

- Design Thinking
- Scrum
- BDD
- DDD
- TDD
- FDD
- MSF
- Kanban

BDD

- Behavior Driven Development,
- é um processo colaborativo que envolve múltiplos membros do time, trabalhando em conjunto com o PO (Product Owner) para descobrir e refinar os requisitos usando, para isso, conversas estruturadas sobre exemplos de uso e comportamento de um sistema ou funcionalidade, buscando o entendimento compartilhado.

DDD

- Domain-Driven Design,
- Trata-se de uma abordagem de design de software de forma disciplinada, abordando uma série de conceitos e técnicas sempre com foco no domínio do software.

TDD

- Test Driven Development
- Ele se baseia na aplicação de pequenos ciclos de repetições. Em cada um deles, um teste é aplicado.
- O objetivo é desenvolver uma função que permita que esse teste tenha um resultado positivo. Ou seja, a função está pronta para ser implementada.

FDD

- Feature-Driven Development
- significa desenvolvimento guiado a funcionalidades, assim como o XP, ASD, Scrum e AUP, faz parte das metodologias ágeis originais, sendo este um modelo incremental e iterativo do processo de desenvolvimento de software que tem como lema resultados frequentes, tangíveis e funcionais.

MSF

- Microsoft Solutions Framework
- é um guia de boas práticas de desenvolvimento de softwares, criado pela Microsoft
- Um projeto MSF é regido por ciclos ou iterações.
- A Microsoft não classifica o MSF como uma metodologia, mas sim como uma disciplina.
- Basicamente, serve como um guia de boas práticas, sem aprofundamento dos detalhes. O que pode parecer uma deficiência, mas permitiu uma abordagem simples e direta das técnicas apresentadas. Permitindo uma fácil compreensão tanto por parte da equipe como do cliente, além de ser bastante flexível em sua aplicação.

Kanban

- é uma metodologia ágil que permite fazer o acompanhamento de tarefas e identificar gargalos no fluxo de trabalho.
- O método kanban pode ser adotado por qualquer tipo de equipe, por ser intuitivo, visual e adaptável a diferentes processos e projetos.

Roteiro da Disciplina

- Apresentação e exercícios de cada método
 - Escolha do assunto para o trabalho
 - Aplicar alguns dos métodos no trabalho
- dividido em dois sprints com troca de especificação/implementação entre os grupos

Avaliação

- Sistema de Avaliação:
 - TDE (Exercícios feitos em sala de aula + a Avaliação; aplica-se a média aritmética dessas tarefas) - Peso 1
 - Trabalho 1 (Desenvolvimento do Sprint 1) - Peso 2
 - Trabalho 2 (Desenvolvimento do Sprint 2) - Peso 3
 - Aplicar a média Harmônica com a seguinte fórmula =
$$6 / ((1 / \text{TDE}) + (2 / \text{Trabalho 1}) + (3 / \text{Trabalho 2}))$$

Bibliografia

- Design Thinking
- Engenharia de Software: Scrum, Processo Unificado, Modelo Espiral, Modelo Cascata
- Diagramas UML
- Artigos: SBES, SBSI, IEEE, ACM, BDBComp