CENTRO ESTADUAL DE EDUCAÇÃO TECNOLOGICA PAULA SOUZA

FACULDADE DE TECNOLOGIA DE SÃO PAULO

LUCAS EDUARDO ORMOND DE SOUSA

**DESENVOLVIMENTO DE SOFTWARE PARA GESTÃO DE PROJETOS EM SCRUM**

SÃO PAULO/SP

2022

LUCAS EDUARDO ORMOND DE SOUSA

**DESENVOLVIMENTO DE SOFTWARE PARA GESTÃO DE PROJETOS EM SCRUM**

Monografia apresentada a FATEC – São Paulo como parte dos requisitos para obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Aristides Novelli Filho.

SÃO PAULO/SP

2022

|  |
| --- |
|  |

LUCAS EDUARDO ORMOND DE SOUSA

**DESENVOLVIMENTO DE SOFTWARE PARA GESTÃO DE PROJETOS EM SCRUM**

Monografia apresentada a FATEC – São Paulo como parte dos requisitos para obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

|  |
| --- |
|  |

Prof. Dr. Aristides Novelli Filho

FATEC – São Paulo

Dedico este trabalho à minha família, que sempre me apoiou e esteve ao meu lado nos momentos mais difíceis, incentivando e renovando as minhas energias.

**AGRADECIMENTOS**

À Deus, por me guiar durante a minha jornada.

À minha família por todo o incentivo e apoio incondicional durante os meus estudos.

Aos meus amigos por todo o incentivo, e pela contribuição direta ou indiretamente à minha graduação.

Ao Prof. Dr. Aristides por ter aceitado ser meu orientador, e por todo o apoio durante o processo de elaboração do TCC.

À minha empresa atual, pela oportunidade de experiência prática no ciclo de desenvolvimento de software.

À Prof. M. Edmea pelas sugestões.

À instituição FATEC – SP juntamente com o CPS pela oportunidade de aprendizado e estudos.

A todos os outros professores da FATEC – SP pela experiência e conhecimento compartilhado durante o processo.

“Sentir é criar. Sentir é pensar sem ideias, e por isso sentir é compreender, visto que o Universo não tem ideias.”

(Fernando Pessoa, 1966)

**RESUMO**

Diferente do modelo em cascata, a metodologia ágil Scrum se baseia em diversos ciclos de atividades. Em cada ciclo, o projeto pode passar por alterações e ser aperfeiçoado de acordo com a interpretação obtida referente ao contexto e situação atual do projeto. O propósito deste trabalho foi analisar conceitos da engenharia de software juntamente com conceitos da metodologia ágil Scrum, para então iniciar o desenvolvimento de um sistema que permita auxiliar e gerir projetos que adotam essa metodologia em seu dia a dia. Foram abordadas as etapas necessárias que um software deve passar para ter êxito em seu processo de desenvolvimento, como especificação dos requisitos funcionais e não-funcionais do sistema, modelagem de software (diagramas de caso de uso, diagramas de classe, diagramas de evento, diagramas de atividade e diagramas de sequência), dentre outros tópicos importantes. Também foram abordados os tópicos principais da Scrum como os papéis da equipe (Product Owner e Scrum Master), Product Backlog, Sprint, dentre outras coisas.

**Palavras-chave:** Engenharia de software; metodologia; Scrum; desenvolvimento; projeto.

**LISTA DE FIGURAS**

FIGURA 1 – Modelo em cascata 16

FIGURA 2 – Modelo em espiral 18

**LISTA DE TABELAS**

**LISTA DE ABREVIATURAS E SIGLAS**

**SUMÁRIO**

[INTRODUÇÃO 13](#_Toc113731615)

[1 ENGENHARIA DE SOFTWARE 15](#_Toc113731616)

[1.1 Definição 15](#_Toc113731617)

[1.2 Ciclo de vida do software 16](#_Toc113731618)

[1.3 Requisitos funcionais e não-funcionais 18](#_Toc113731619)

# INTRODUÇÃO

**Contextualização**

O software está presente em diversas áreas, e por esse motivo sua existência é de extrema importância no mundo moderno. Grande parte de infraestruturas, serviços, indústrias, manufatura, dentre outras coisas, já funcionam de forma totalmente informatizada. Devido a esse fator, a engenharia de software se mostra fundamental para o funcionamento da sociedade (Sommerville, 2011).

Há diversos tipos de sistemas de softwares, e cada um possui a sua própria particularidade. Por esse motivo, apesar dessas aplicações precisarem de engenharia de software, não existem métodos ou técnicas universais a serem utilizadas. Portanto, é necessário observar a necessidade de cada sistema para abordá-lo de forma adequada (Id., 2011).

É importante ressaltar que a engenharia de software engloba diversos processos, não se limitando somente a desenvolver um simples programa de computador. Segundo Sommerville (2011, p. 3) “[...] quando falamos de engenharia de software, não se trata apenas do programa em si, mas de toda a documentação associada e dados de configurações necessários para fazer esse programa operar corretamente.”.

Tendo em vista esses conceitos e a importância da engenharia de software, ao desenvolver um sistema que consiga auxiliar na gestão de projetos em Scrum, ou qualquer outro sistema, é necessário levar em conta toda a estrutura e modelagem necessária para que ele consiga funcionar de maneira correta e consiga atender todas as expectativas.

**Objetivos Gerais**

Pesquisar sobre as etapas de desenvolvimento de software, modelagem de software, além de também entender a estrutura, funcionamento e organização da metodologia ágil Scrum.

**Objetivos Específicos**

Aplicar de forma prática todos os conhecimentos pesquisados, para modelar e estruturar um software que permita configurar e auxiliar projetos que adotam a metodologia ágil Scrum durante o seu ciclo de vida.

**Justificativa**

A proposta na presente pesquisa vai servir para aprofundar os conhecimentos pesquisados referentes a engenharia de software juntamente com a metodologia Scrum. Há ainda a aplicação prática desses conhecimentos, no qual será modelado um software capaz de auxiliar na gestão de projetos em Scrum, permitindo assim uma visão desses conceitos que vai além do estudo da parte teórica.

# 1 ENGENHARIA DE SOFTWARE

Este capítulo apresenta detalhes sobre os fundamentos da engenharia de software. Serão abordados tópicos sobre modelagem de software, envolvendo diagrama de classes, diagrama de caso de uso, diagrama de atividades e diagrama de eventos. Além disso, o capítulo também irá contemplar conceitos como especificação de requisitos funcionais e não-funcionais, ciclo de vida de software, dentre outros assuntos.

## 1.1 Definição

Segundo Stephen Schach (2009, p. 4) “[...] a engenharia de software é uma disciplina cujo objetivo é produzir software isento de falhas, entregue dentro do prazo e orçamento previstos, e que atenda às necessidades do cliente. Além disso, o software deve ser fácil de ser modificado quando as necessidades do usuário mudarem.”.

“Engenharia de Software trata da aplicação de abordagens sistemáticas, disciplinadas e quantificáveis para desenvolver, operar, manter e evoluir software. Ou seja, Engenharia de Software é a área da Computação que se preocupa em propor e aplicar princípios de engenharia na construção de software.” (Marco Túlio, 2020, p. 2).

“Engenharia de software é uma disciplina de engenharia cujo foco está em todos os aspectos da produção de software, desde os estágios iniciais da especificação do sistema até sua manutenção, quando o sistema já está sendo usado” (Sommerville, 2011, p. 5).

Diante das citações acima, conclui-se que a engenharia de software é uma disciplina extremamente ampla e que aborda diversos temas referente ao desenvolvimento de sistemas. Existe uma preocupação desde as etapas iniciais até a sua manutenção, sendo que o software deve cumprir com o que foi especificado, e estar livre de falhas que possam o comprometer. Em suma, a engenharia de software utiliza-se de técnicas e abordagens que permitem que o produto final, no caso o software, consiga atender às necessidades do cliente, considerando funcionalidade, prazo e orçamento previsto.

## 1.2 Ciclo de vida do software

O modelo de ciclo de vida do software é uma descrição das etapas que devem ser seguidas ao desenvolver um sistema. Por ser mais fácil realizar uma sequência de tarefas menores do que uma grande tarefa, o ciclo de vida é subdividido em uma série de etapas menores que são denominadas fases. Dependendo do modelo de ciclo de vida a ser adotado, o número de fases pode variar para mais ou para menos (Schach, 2009).

**Modelo cascata**

Foi o primeiro modelo de ciclo de vida de software a ser publicado. No modelo em cascata, as fases são executadas em sequência. O estágio seguinte não deve ser iniciado até que a fase anterior seja concluída. Nesse modelo, as iterações podem ser dispendiosas envolver retrabalho. Por isso, somente é recomendável seu uso quando os requisitos são bem compreendidos e que não venham a ser alterados no futuro (Sommerville, 2011). É baseado nas seguintes etapas:

- Análise e definição dos requisitos, obtido através da consulta aos usuários;

- Projeto de sistema e software, no qual os requisitos são alocados tanto para hardware quanto para software;

- Implementação e teste unitário, no qual o software é desenvolvido em unidades de programa e cada unidade é testada para que os requisitos sejam atendidos;

- Integração e teste de sistema, onde as unidades individuais são integradas e testadas como um sistema completo;

- Operação e manutenção, onde o sistema é instalado colocado em uso e sofre manutenções.

A FIGURA 1 apresenta as principais fases e fluxo do funcionamento do modelo em cascata.

Diagrama

Descrição gerada automaticamente

FIGURA 1 – Modelo em cascata.

Fonte: Sommerville, 2011, pág. 20

**Modelo espiral**

O modelo espiral se baseia em diversas iterações, no qual cada iteração versões mais completas do software são construídas. Esse modelo considera os riscos e a avaliação do cliente em cada iteração. Isso ajuda a mitigar grandes problemas que podem surgir durante a execução do projeto (Pressman, 1995). É definido pelas seguintes quatro atividades:

- Planejamento, onde ocorre a determinação de objetivos, restrições e alternativas;

- Análise dor riscos, onde ocorre a identificação dos riscos ou resolução dos riscos;

- Engenharia, onde ocorre o desenvolvimento do produto no “nível seguinte”;

- Avaliação feita pelo cliente, onde os resultados da engenharia são avaliados.

A FIGURA 2 apresenta as principais fases e fluxo do funcionamento do modelo em espiral.

Diagrama, Desenho técnico

Descrição gerada automaticamente

FIGURA 2 – Modelo em espiral

Fonte: Adaptado de Pressman, 1995, p. 39

Diante dos conceitos apresentados, é possível observar que a codificação é feita somente após uma análise detalhada do escopo do software e requisitos do cliente. Todo o projeto é dividido em fases, e cada fase do projeto possui a sua própria responsabilidade. As fases podem sofrer variações dependendo do modelo a ser adotado.

## 1.3 Requisitos funcionais e não-funcionais