



INSTITUTO POLITÉCNICO DE BEJA

Escola Superior de Tecnologia e Gestão

Licenciatura em Engenharia Informática



EPItogether

André Reis nº21269

Pedro Duarte nº21190

INSTITUTO POLITÉCNICO DE BEJA
Escola Superior de Tecnologia e Gestão
Licenciatura em Engenharia Informática

EPItogether

Elaborado por:

André Reis nº21269

Pedro Duarte nº21190

Orientado por:

Isabel Sofia Sousa Brito

Relatório de projeto da cadeira de Engenharia de Software na
Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Beja

2024

Resumo

O presente relatório fornece uma visão geral do projeto de desenvolvimento de software em andamento, detalhando o modelo de execução, os prazos previstos e as ferramentas a serem utilizadas. Com uma abordagem estruturada e colaborativa, buscamos alcançar nossos objetivos dentro do cronograma estabelecido, contando com a contribuição dos membros da equipe em suas áreas de especialização. Através do uso de ferramentas e plataformas CASE adequadas, visamos superar desafios e entregar um produto final de alta qualidade. Este relatório representa um marco importante em nosso progresso, demonstrando nosso compromisso com a excelência e a satisfação dos stakeholders.

Palavras-chave: desenvolvimento de software, projeto, modelo de execução, prazos, ferramentas CASE, colaboração, equipe, especialização, desafios, qualidade, stakeholders.

Abstract

This report provides an overview of the ongoing software development project, detailing the execution model, estimated deadlines, and tools to be used. With a structured and collaborative approach, we aim to achieve our objectives within the established timeline, leveraging the contributions of team members in their areas of expertise. Through the use of appropriate CASE tools and platforms, we seek to overcome challenges and deliver a high-quality end product. This report represents a significant milestone in our progress, demonstrating our commitment to excellence and stakeholder satisfaction.

Keywords: software development, project, execution model, deadlines, CASE tools, collaboration, team, expertise, challenges, quality, stakeholders.

Índice

1. Introdução	2
2. Modelo de Desenvolvimento de Software:	3
2.1. Descrição do Modelo:	4
3. Controlo de versões	5
4. Prazos previstos para execução das tarefas/atividades.....	5
5. Conclusões.....	6

Lista de Figuras

Figura 1- Modelo Cascata.....	3
-------------------------------	---

1. Introdução

Este relatório apresenta um ponto de situação do projeto de desenvolvimento de software em curso, delineando o plano de execução e as etapas a serem seguidas para sua conclusão bem-sucedida. O objetivo principal deste relatório é fornecer uma visão abrangente do modelo de desenvolvimento de software adotado, juntamente com os prazos estimados para cada atividade e as ferramentas a serem utilizadas ao longo do processo. Além disso, serão destacados os membros da equipe envolvidos em cada fase do projeto e algumas ferramentas Case, visando uma gestão eficiente e colaborativa. Este relatório servirá como um guia estratégico para o desenvolvimento contínuo do software, garantindo a conformidade com os requisitos estabelecidos e a entrega oportuna de resultados de qualidade.

2. Modelo de Desenvolvimento de Software:

Cascata.

O modelo em cascata é um dos modelos de ciclo de vida de desenvolvimento de software mais antigos e amplamente utilizados. Ele segue uma abordagem sequencial, onde o desenvolvimento do software é dividido em fases distintas (Análise, Desenho, Codificação, Teste, Implantação Manutenção), cada uma das quais deve ser concluída antes de avançar para a próxima.

O modelo em cascata é caracterizado pela sua abordagem sequencial e linear, onde cada fase depende do resultado da fase anterior. Isso significa que qualquer mudança nos requisitos ou no design do sistema pode ser difícil de incorporar após o início da implementação, tornando o modelo menos flexível em comparação com abordagens ágeis mais iterativas. No entanto, o modelo em cascata é eficaz para projetos onde os requisitos são bem compreendidos e estáveis desde o início.

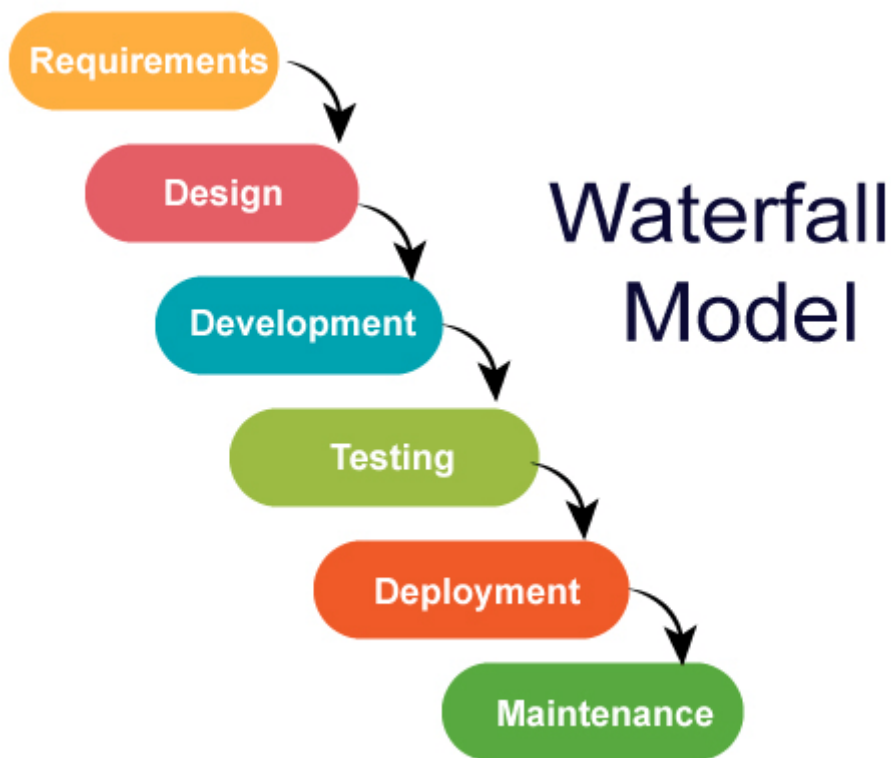


Figura 1- Modelo Cascata

2.1. Descrição do Modelo:

1. Análise (2 semanas):

- Recolha de informação: Realizar entrevistas com potenciais utilizadores, pesquisa online de sistemas semelhantes.
- Identificação de requisitos: Diferenciar entre requisitos funcionais e não funcionais, especificar requisitos usando template.
- Ferramentas CASE: Utilizar VisualParadigm para documentação e modelagem.
- Equipa: Todos os membros da equipa envolvidos.

2. Desenho (3 semanas):

- Elaboração de diagramas UML: Desenvolver diagramas de sequência, diagrama de classes e outros diagramas relevantes.
- Ferramentas CASE: Continuar a utilizar VisualParadigm para elaboração de diagramas.
- Equipa: Especialistas em modelagem e design.

3. Codificação (4 semanas):

- Desenvolvimento da aplicação móvel e da página web: Programação das funcionalidades, criação de interfaces de usuário.
- Plataformas: Utilizar frameworks como React Native para a aplicação móvel e React.js para a página web.
- Ferramentas CASE: IDEs como Visual Studio Code para desenvolvimento.
- Equipa: Programadores responsáveis pela implementação.

4. Testes (2 semanas):

- Testes de unidade, integração e sistema: Verificar se o software atende aos requisitos e está livre de bugs.
- Ferramentas CASE: Utilizar ferramentas de automação de testes como Jest e Selenium.
- Equipa: Testadores dedicados.

5. Implantação (1 semana):

- Preparação para lançamento: Preparar ambientes de produção, fazer testes finais.
- Ferramentas CASE: Ferramentas de integração contínua como Jenkins.
- Equipa: DevOps responsáveis pela implantação.

6. Manutenção (contínua):

- Correção de bugs, atualizações de segurança e melhorias de desempenho.
- Ferramentas CASE: Ferramentas de monitoramento como New Relic.
- Equipa: DevOps e programadores responsáveis pela manutenção.

3. Controlo de versões

Github.

O GitHub foi escolhido como a ferramenta de controlo de versões para este projeto de devido à sua ampla adoção na indústria de tecnologia e às suas características robustas e amigáveis. Como uma plataforma baseada em nuvem, o GitHub permite que a equipe de desenvolvimento colabore de forma eficiente, mantendo um histórico detalhado de todas as alterações no código-fonte. Além disso, oferece recursos poderosos, como ramificações (branches), solicitações de pull (pull requests) e integração contínua, que facilitam a gestão do código e a implementação de boas práticas de desenvolvimento. A interface intuitiva do GitHub torna a navegação e a revisão do código simples para todos os membros da equipe, promovendo uma comunicação transparente e eficaz durante todo o ciclo de desenvolvimento do software.

4. Prazos previstos para execução das tarefas/atividades

- Análise: 2 semanas
- Desenho: 3 semanas
- Codificação: 4 semanas
- Testes: 2 semanas
- Implantação: 1 semana

5. Conclusões

Neste ponto de situação do projeto, delineamos um plano abrangente para o desenvolvimento de software, estabelecendo um modelo claro de execução e definindo os prazos para cada fase. Ao adotar uma abordagem estruturada e colaborativa, estamos confiantes de que seremos capazes de alcançar nossos objetivos dentro do cronograma estabelecido. Com o uso das ferramentas e plataformas CASE adequadas, juntamente com a contribuição dos membros da equipe em suas respectivas áreas de especialização, estamos bem posicionados para superar os desafios e entregar um produto final de alta qualidade. Este relatório serve como um marco importante no nosso caminho em direção ao sucesso do projeto, demonstrando nosso compromisso com a excelência e a entrega de valor aos nossos clientes e usuários finais.