

PPGES

Modelagem de processo para software de sistemas IoT utilizando a Abordagem Essence

Luís Fernando Alves da Silva & Pedro Ramires da Silva Amalfi Costa



PPGES

Modelagem de processo para software de
sistemas IoT utilizando a Abordagem
Essence

by

Luís Fernando Alves da Silva & Pedro Ramires
da Silva Amalfi Costa

advising by

Maicon Bernardino



Sumário

1	Resumo Executivo	1
2	Introdução	2
3	Background	3
3.1	Linguagem de modelagem de processos	3
3.2	Essence	3
4	Estudo de Caso	5
4.1	Objetivo do Projeto	5
4.2	Implementação do Framework Essence	5
4.2.1	Clientes	5
4.2.2	Solução	6
4.2.3	Esforço	6
4.3	Resultados Esperados	6
4.4	Conclusão do Estudo de Caso	6
5	Modelagem de Processos	7
5.1	Áreas de Interesse	7
5.1.1	Clientes	7
5.1.2	Soluções	7
5.1.3	Esforços	8
5.2	Alfas	8
5.3	Estados de Alfa	9
5.3.1	Clientes	9
5.3.2	Solução	10
5.3.3	Esforços	11
5.4	Espaço de Atividades	11
5.4.1	Área de Interesse: Cliente	11
5.4.2	Área de Interesse: Solução	16
5.4.3	Área de Interesse: Esforço	16
5.5	Competências	17
5.5.1	Área de Interesse: Cliente	18
5.5.2	Área de Interesse: Solução	18
5.5.3	Área de Interesse: Esforço	19
6	Vantagens e desvantagens	20
6.1	Vantagens da Modelagem Essence	20
6.2	Desvantagens da Modelagem Essence	20
7	Conclusão	21
7.1	Resumo das Principais Considerações	21
7.2	Considerações Finais	21
	Referências	22

Resumo Executivo

A modelagem Essence é uma abordagem poderosa para estruturar e gerenciar o desenvolvimento de software de maneira eficiente e eficaz. Baseada no Kernel Essence, esta metodologia organiza o processo de desenvolvimento em torno de Áreas de Interesse, Alfas, Estados de Alfa e Espaços de Atividades. Cada um desses componentes desempenha um papel crucial na definição, monitoramento e otimização das atividades do projeto.

A modelagem é dividida em três principais Áreas de Interesse: Clientes, Solução e Esforço. Cada uma dessas áreas cobre aspectos essenciais do desenvolvimento de software, assegurando que todas as fases do ciclo de vida do projeto sejam tratadas de forma integrada.

No contexto da Área de Interesse Clientes, há foco no Engajamento de Usuários e Sensores, abordando a identificação e manutenção das necessidades dos usuários e sensores. Para a Área de Interesse Solução, destaca-se o Ciclo de Vida da Aplicação e a Gestão de Dados, abrangendo desde a concepção até a manutenção da aplicação, além da gestão segura e eficiente dos dados. Finalmente, na Área de Interesse Esforço, concentra-se na Gestão de Trabalho e Métodos, assim como na Gestão de Equipes de Programadores, garantindo a preparação e eficiência da equipe de desenvolvimento.

Cada Alfa passa por diferentes Estados de Alfa que representam o progresso e a maturidade em cada área de interesse, como Usuário (Identificado, Entendido, Engajado) e Aplicação (Concebida, Desenvolvida, Testada, Implantada, Mantida). Os Espaços de Atividades foram definidos para agrupar atividades relacionadas de forma a facilitar a gestão e a execução das tarefas.

Para executar eficazmente a modelagem Essence, é necessário um conjunto de competências que varia de habilidades de comunicação básica até a otimização e melhoria contínua. Este artigo explora detalhadamente cada uma dessas áreas, proporcionando uma visão abrangente e prática da aplicação da modelagem Essence no desenvolvimento de software.

2

Introdução

No desenvolvimento de software, a estruturação e a gestão eficiente dos processos são essenciais para o sucesso dos projetos. Com a crescente complexidade dos sistemas de software e a demanda por entregas rápidas e de alta qualidade, as metodologias de desenvolvimento tradicionais frequentemente se mostram insuficientes. Nesse contexto, a modelagem de processos de software emerge como uma solução para enfrentar esses desafios, proporcionando uma abordagem sistemática e disciplinada para a criação de software.

A abordagem Essence, desenvolvida pela SEMAT (Software Engineering Method and Theory), propõe uma maneira inovadora de modelar e gerenciar processos de desenvolvimento de software. Baseada em um Kernel unificado e flexível, a modelagem Essence organiza o processo em torno de componentes fundamentais, chamados de Alfas, que representam os principais elementos do desenvolvimento de software, como Usuários, Requisitos, Aplicação e Equipe de Desenvolvimento. Cada Alfa passa por diferentes Estados de Alfa, que indicam o progresso e a maturidade ao longo do ciclo de vida do projeto.

Este artigo explora a aplicação da modelagem Essence no contexto do desenvolvimento de software, apresentando uma visão detalhada das Áreas de Interesse, Alfas, Estados de Alfa e Espaços de Atividades. Dividido em três principais Áreas de Interesse (Clientes, Solução e Esforço), o artigo fornece um framework claro e estruturado para guiar o progresso do projeto de software, desde a identificação das necessidades dos usuários até a entrega e manutenção da solução.

O objetivo deste trabalho é fornecer uma compreensão abrangente e prática da modelagem Essence, destacando suas vantagens, desafios e as competências necessárias para sua implementação eficaz. Através de exemplos práticos e discussões teóricas, será demonstrado como a modelagem Essence pode transformar a maneira como os projetos de software são planejados, executados e gerenciados.

3

Background

A modelagem de processos e a engenharia de software são disciplinas que convergem significativamente, especialmente quando se busca garantir a eficiência, a eficácia e a segurança dos sistemas desenvolvidos. Com a crescente complexidade dos sistemas de software e a necessidade de atender a requisitos rigorosos de segurança, tornou-se essencial adotar abordagens integradas desde as fases iniciais do desenvolvimento.

Neste contexto, a utilização de linguagens de modelagem de processos bem estabelecidas, como BPMN (Business Process Model and Notation) [3] e UML (Unified Modeling Language) [1], em conjunto com frameworks modernos como o Essence, oferece uma base sólida para a identificação e mitigação de ameaças, além de melhorar a qualidade e a segurança dos sistemas desenvolvidos. Este capítulo explora como essas ferramentas e abordagens são aplicadas no contexto da modelagem de processos de software, abordando suas funcionalidades, vantagens e desafios.

3.1. Linguagem de modelagem de processos

A linguagem de modelagem de processos é uma ferramenta essencial na engenharia de software, permitindo a representação visual e textual dos processos que compõem o ciclo de vida do desenvolvimento de software. Essas linguagens ajudam a definir, analisar e comunicar processos de forma clara e precisa, facilitando a compreensão e a colaboração entre os membros da equipe e outras partes interessadas.

Existem várias linguagens de modelagem de processos amplamente utilizadas, como UML (Unified Modeling Language), BPMN (Business Process Model and Notation) e SPEM (Software and Systems Process Engineering Metamodel) [4]. Cada uma dessas linguagens tem suas próprias características e é adequada para diferentes tipos de projetos e necessidades.

A modelagem de processos desempenha um papel crucial na identificação de ineficiências, na padronização de práticas e na garantia de qualidade do software. Ao mapear visualmente os processos, é possível identificar pontos de melhoria e otimizar o fluxo de trabalho, resultando em um desenvolvimento de software mais eficiente e eficaz.

3.2. Essence

ESSENCE é uma estrutura desenvolvida pela SEMAT (Software Engineering Method and Theory) para modelagem e gestão de processos de desenvolvimento de software. Essa abordagem inovadora se diferencia das metodologias tradicionais ao oferecer um Kernel flexível e adaptável, projetado para ser aplicável em uma ampla gama de contextos e tipos de projetos de software [2].

O Kernel ESSENCE é fundamentado em Alfas, que representam elementos essenciais do desenvolvimento de software, como Usuários, Requisitos, Solução e Equipe. Cada Alfa passa por Estados de Alfa, indicando seu progresso e maturidade ao longo do ciclo de vida do projeto. Além disso, ESSENCE define Espaços de Atividades, agrupando atividades relacionadas para facilitar a estruturação e gestão dos processos.

Ao adotar ESSENCE, as equipes de desenvolvimento podem selecionar e adaptar práticas de acordo com as necessidades específicas do projeto, promovendo a flexibilidade, a melhoria contínua e a inovação. Isso não apenas melhora a eficiência e eficácia do desenvolvimento de software, mas também facilita a comunicação, alinhamento e colaboração entre os membros da equipe e stakeholders.

Este artigo explora a aplicação prática do ESSENCE no contexto do desenvolvimento de software, demonstrando como essa abordagem pode ser utilizada para otimizar desde a captura de requisitos até a entrega e manutenção de soluções de software, garantindo qualidade e alinhamento com as expectativas dos clientes e usuários finais.

4

Estudo de Caso

Este estudo de caso aborda o desenvolvimento de um sistema de monitoramento de ambiente para coleta e transmissão de informações de temperatura e umidade. O projeto visa criar uma solução que permita monitorar condições ambientais em diferentes locais, utilizando sensores distribuídos e um servidor local para processamento e armazenamento dos dados coletados. O sistema será acessado por um único usuário para visualização e análise das informações coletadas. Um Kernel denominado KISS (Kernel for IoT Systems Software) foi produzido a partir deste estudo de caso e se encontra disponível em (KISS Wiki).

4.1. Objetivo do Projeto

O objetivo principal deste projeto é desenvolver e implementar um sistema de monitoramento que:

- Coleta dados de temperatura e umidade de múltiplos sensores distribuídos em diversos pontos do ambiente monitorado.
- Transmite os dados coletados para um servidor local para processamento e armazenamento centralizado.
- Permite que um único usuário acesse e visualize os dados de forma intuitiva e eficiente.
- Demonstra as etapas e processos de modelagem utilizando o framework Essence, proporcionando uma compreensão clara.
- Serve como base e referência para outros projetos de IoT na agroindústria interessados em implementar soluções similares.

4.2. Implementação do Framework Essence

O framework Essence será aplicado no desenvolvimento deste projeto, focando nas três principais Áreas de Interesse: Clientes, Solução e Esforço.

4.2.1. Clientes

Na área de interesse dos Clientes, as atividades incluirão:

- Identificação dos usuários finais do sistema de monitoramento.
- Entendimento das necessidades e requisitos específicos dos usuários em relação à visualização e análise dos dados de temperatura e umidade.
- Engajamento dos usuários no processo de desenvolvimento para garantir que suas expectativas sejam atendidas ao longo do projeto.

4.2.2. Solução

Na área de interesse da Solução, serão realizadas as seguintes atividades:

- Definição dos requisitos funcionais e não funcionais do sistema de monitoramento, principalmente requisitos que agregem segurança ao projeto.
- Projeto da arquitetura do sistema, incluindo a integração dos sensores, protocolos de comunicação e o servidor local.
- Implementação do sistema conforme os requisitos definidos, utilizando tecnologias adequadas para coleta, transmissão e armazenamento de dados.
- Testes para verificar a funcionalidade e a integridade dos dados coletados pelo sistema.

4.2.3. Esforço

Na área de interesse do Esforço, serão consideradas as seguintes atividades:

- Planejamento das etapas de desenvolvimento, estabelecendo metas claras e cronogramas realistas.
- Execução das atividades conforme planejado, monitorando o progresso e identificando possíveis desvios.
- Revisão e otimização contínua do sistema para melhorar a eficiência e a precisão na coleta e transmissão de dados.

4.3. Resultados Esperados

Espera-se que a implementação do sistema de monitoramento de ambiente proporcione os seguintes resultados:

- Coleta precisa e confiável de dados de temperatura e umidade em tempo real.
- Transmissão eficiente dos dados para o servidor local, garantindo integridade e segurança.
- Interface intuitiva e amigável para que o usuário visualize e interprete os dados coletados de forma eficaz.
- Demonstração clara das etapas e processos de modelagem utilizando o framework Essence, facilitando a replicação e compreensão por parte de outros interessados na área.

4.4. Conclusão do Estudo de Caso

Este estudo de caso ilustra como a modelagem com o framework Essence pode ser aplicada com sucesso em projetos de IoT na agroindústria, especificamente para monitoramento ambiental. Através da definição clara de requisitos, arquitetura robusta e gerenciamento eficaz de processos, o projeto busca não apenas implementar uma solução funcional, mas também servir como um guia para outros profissionais interessados em desenvolver soluções similares. A modelagem com Essence demonstra seu valor ao proporcionar uma abordagem estruturada e escalável para o desenvolvimento de sistemas complexos de IoT.

5

Modelagem de Processos

A modelagem de processos é fundamental para a definição estruturada e eficiente dos elementos essenciais de um sistema de software. Neste capítulo, será explorado detalhadamente as principais áreas de interesse, alfas, estados de alfa, espaços de atividades e competências necessárias para modelar um sistema usando a abordagem Essence.

5.1. Áreas de Interesse

A modelagem Essence organiza um sistema de software em três áreas principais de interesse: Clientes, Soluções e Esforços. Cada área aborda aspectos cruciais para o desenvolvimento e operação de um sistema, garantindo que todas as partes interessadas e requisitos sejam adequadamente considerados desde o início do processo de desenvolvimento como pode ser visto na Figura 5.1, em seguida será falado com mais detalhamento cada uma dessas áreas.

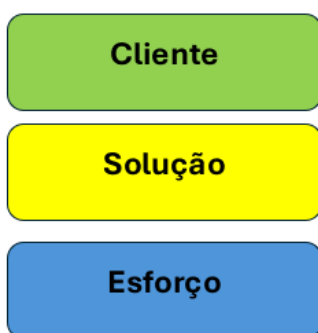


Figure 5.1: Ilustração das Áreas de Interesse na modelagem Essence

5.1.1. Clientes

Descrição: Esta área de interesse representa tudo o que está relacionado com o uso e exploração reais do sistema de software a ser produzido.

- **Usuário Final:** O único usuário que acessa os dados do servidor local. Este usuário pode ser, por exemplo, um técnico de manutenção, um gerente de instalações, ou qualquer pessoa responsável por monitorar e analisar os dados de temperatura e umidade coletados pelos sensores.
- **Interações:** Como o usuário interage com o sistema, que tipo de interface é necessária, e quais são as expectativas em termos de desempenho e segurança dos dados acessados.

5.1.2. Soluções

Descrição: Esta área de interesse compreende tudo necessário para realizar a especificação e desenvolvimento do sistema de software.

- **Sensores de Temperatura e Umidade:** Dispositivos que coletam dados ambientais.
- **Servidor Local:** Computador ou servidor onde os dados dos sensores são coletados, armazenados e processados.
- **Software de Coleta e Transmissão:** Programas que controlam os sensores, coletam os dados e os transmitem ao servidor.
- **Interface de Usuário:** Aplicações ou dashboards que permitem ao usuário final visualizar e interagir com os dados.
- **Protocolos de Comunicação:** Tecnologias e protocolos usados para transmitir dados dos sensores ao servidor, como Wi-Fi, Bluetooth, Zigbee, etc.
- **Mecanismos de Segurança:** Medidas de segurança implementadas para proteger os dados coletados, incluindo criptografia, autenticação, controle de acesso, etc.

5.1.3. Esforços

Descrição: Esta área de preocupação representa tudo relativo à equipe e à maneira como eles realizam seu trabalho.

- **Equipe de Desenvolvimento:** Engenheiros de software, desenvolvedores de firmware, e outros profissionais responsáveis pelo design e implementação do sistema de monitoramento.
- **Equipe de Manutenção:** Técnicos responsáveis pela instalação, calibração, e manutenção dos sensores e do servidor.
- **Gestão de Projeto:** Gerentes de projeto e coordenadores que supervisionam o progresso, prazos e qualidade do trabalho realizado.
- **Práticas e Processos:** Métodos e práticas adotadas pela equipe para garantir um desenvolvimento eficiente e de alta qualidade, incluindo práticas de DevSecOps, revisões de código, testes de segurança, e compliance com normas e regulamentos de segurança.
- **Capacitação e Treinamento:** Programas de treinamento para assegurar que a equipe está atualizada com as melhores práticas de segurança e desenvolvimento de software.

5.2. Alfas

Os Alfas no framework Essence representam os principais elementos do desenvolvimento de software, distribuídos entre as áreas de interesse do Cliente, Solução e Esforço. Na área de interesse Clientes, os Alfas incluem Usuário e Sensores, refletindo a interação direta com o sistema desenvolvido. Na área de interesse Solução, os Alfas são compostos por Dados, Aplicação e Requisitos, abrangendo aspectos fundamentais para a construção e funcionamento da solução de software. Por fim, na área de interesse Esforço, os Alfas envolvem Programadores, Trabalho e Método de Trabalho, essenciais para a implementação e gestão eficaz do processo de desenvolvimento.

Cada Alfa passa por diferentes estados ao longo do ciclo de vida do projeto, refletindo seu progresso e maturidade. Compreender os Alfas e seus estados é fundamental para gerenciar e monitorar eficientemente o desenvolvimento de software como pode ser visto a figura 5.2.

Os relacionamentos entre os Alfas são cruciais para o funcionamento integrado do sistema. No contexto do desenvolvimento de um sistema de monitoramento de ambiente para coleta e transmissão de informações de temperatura e umidade, os principais relacionamentos incluem:

- O **Usuário** utiliza a **Aplicação** para visualizar eventos e dados coletados pelos **Sensores**.
- O **Usuário** monitora os **Sensores** para auxiliar na tomada de decisões informadas.
- Os **Sensores** enviam dados para a **Aplicação**, que os armazena e processa.
- Os **Requisitos Funcionais e Não Funcionais** orientam o trabalho dos **Programadores** durante o desenvolvimento da **Aplicação**.

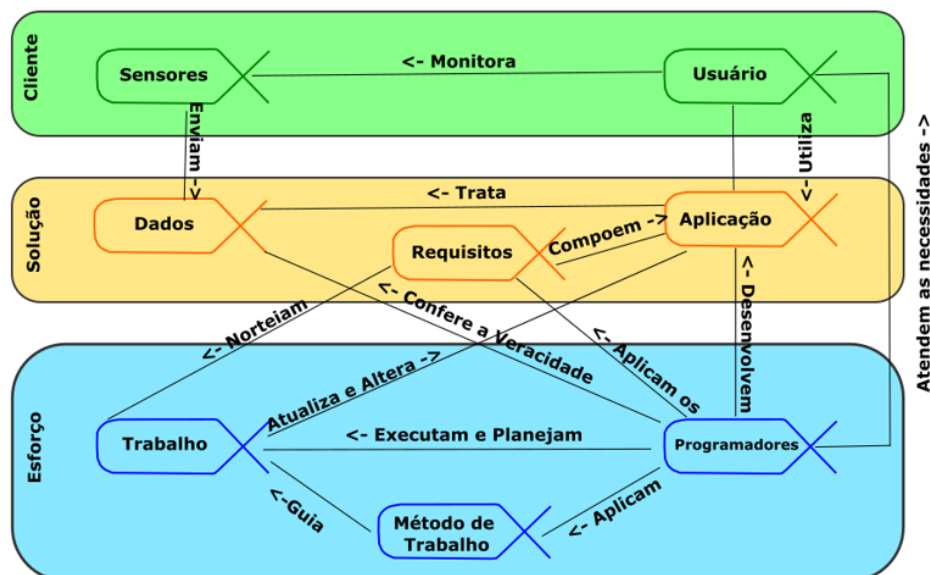


Figure 5.2: Relacionamento entre os ALFAS

- Os **Programadores** aplicam métodos de trabalho específicos para executar e planejar suas tarefas.
- Os **Programadores** verificam a veracidade dos dados provenientes dos sensores e garantem sua integridade dentro da **Aplicação**.
- Durante o desenvolvimento da **Aplicação**, os **Programadores** implementam os requisitos, incluindo medidas de segurança necessárias.
- O **Método de Trabalho** fornece diretrizes essenciais que orientam todo o processo de desenvolvimento.

Esses relacionamentos destacam a interdependência entre os elementos essenciais do sistema, promovendo uma integração eficaz e um desenvolvimento coordenado.

5.3. Estados de Alfa

Os Estados de Alfa representam as condições e características de um Alfa em um determinado momento do tempo. Cada Alfa pode estar em um dos vários estados, como Iniciado, Executado, Concluído, Revisado e Otimizado, dependendo do seu progresso e das atividades realizadas. Acompanhar os Estados de Alfa é fundamental para garantir que o projeto avance conforme o planejado e atenda aos requisitos ao longo do ciclo de vida do software.

5.3.1. Clientes

Os Estados de Alfa relacionados aos Clientes são fundamentais para entender como os usuários finais e os sensores interagem com o sistema.

5.3.1.1. Usuários

Os estados de Alfa do Usuário (Figura 5.3) são:

1. Reconhecido: Identificação dos usuários finais que acessarão e utilizarão os dados.
2. Envolvido: Usuários estão cientes do sistema e foram consultados sobre suas necessidades.
3. Participante: Usuários participam ativamente do desenvolvimento, fornecendo feedback.
4. Satisfeito: Usuários estão satisfeitos com a solução desenvolvida e ela atende às suas necessidades.
5. Defensor: Usuários promovem e apoiam a utilização do sistema.

5.3.1.2. Sensores

Os estados de Alfa do Sensor (Figura 5.4) incluem:

1. Identificado: Todos os sensores necessários foram identificados e especificados.
2. Instalado: Sensores foram instalados no ambiente alvo.
3. Operacional: Sensores estão operando e coletando dados corretamente.
4. Monitorado: Dados dos sensores são monitorados continuamente para garantir precisão.
5. Manutenido: Sensores recebem manutenção regular para assegurar funcionalidade a longo prazo.

5.3.2. Solução

Os Estados de Alfa relacionados à Solução (Figuras 5.5, 5.6, 5.7) são essenciais para o desenvolvimento e operação do sistema.

5.3.2.1. Dados

Os estados de Alfa de Dados são:

1. Coletado: Dados estão sendo coletados dos sensores.
2. Armazenado: Dados coletados estão sendo armazenados no servidor local.
3. Processado: Dados armazenados estão sendo processados para análises.
4. Analisado: Dados processados foram analisados e utilizados para tomadas de decisão.
5. Segurado: Dados estão seguros contra acessos não autorizados e violações.

5.3.2.2. Aplicação

Os estados de Alfa da Aplicação são:

1. Concebida: Aplicação foi concebida e seu design foi finalizado.
2. Desenvolvida: Aplicação foi desenvolvida e está funcional.
3. Testada: Aplicação foi testada para garantir que atende aos requisitos.
4. Implantada: Aplicação foi implantada no servidor local.
5. Mantida: Aplicação está sendo mantida e atualizada conforme necessário.

5.3.2.3. Requisitos

Os estados de Alfa de Requisitos são:

1. Elicitado: Requisitos funcionais e não funcionais foram coletados através de entrevistas, workshops e análise de documentos.
2. Documentado: Requisitos foram documentados de forma clara e precisa, incluindo detalhes técnicos e de segurança.
3. Validado: Requisitos foram validados com os clientes para garantir que atendam às necessidades e expectativas.
4. Implementado: Requisitos foram implementados no sistema, assegurando que todas as especificações sejam atendidas, especialmente os requisitos de segurança.
5. Monitorado: Requisitos estão sendo monitorados continuamente para ajustes e atualizações conforme necessário, incluindo medidas adicionais de segurança.

5.3.3. Esforços

Os Estados de Alfa relacionados aos Esforços (Figuras 5.8, 5.9, 5.10) destacam a importância da equipe e dos processos de trabalho.

5.3.3.1. Programadores

Os estados de Alfa dos Programadores são:

1. Formado: Equipe de programadores foi formada.
2. Treinado: Equipe foi treinada nas ferramentas e métodos de trabalho.
3. Engajado: Equipe está engajada no desenvolvimento ativo do sistema.
4. Coordenado: Equipe trabalha de forma coordenada e eficiente.
5. Avaliado: Desempenho da equipe é avaliado regularmente para melhorias contínuas.

5.3.3.2. Trabalho

Os estados de Alfa do Trabalho são:

1. Iniciado: Atividades de trabalho foram iniciadas.
2. Executado: Atividades de trabalho estão sendo executadas conforme planejado.
3. Concluído: Atividades de trabalho foram concluídas com sucesso.
4. Revisado: Atividades de trabalho são revisadas para identificar melhorias.
5. Otimizado: Práticas de trabalho são otimizadas para melhor desempenho.

5.3.3.3. Método de Trabalho

Os estados de Alfa do Método de Trabalho são:

1. Definido: Método de trabalho foi definido e documentado.
2. Implementado: Método de trabalho foi implementado na prática.
3. Melhorado: Método de trabalho é revisado e melhorado continuamente.
4. Padronizado: Processos são padronizados e documentados.
5. Avaliado: Desempenho do método de trabalho é avaliado e monitorado.

Os cards sobre cada uma das evoluções em cima de cada Estado Alfa apresentado pode ser visto com mais detalhes no link da wiki do projeto (LINK).

5.4. Espaço de Atividades

O Espaço de Atividades define um conjunto de atividades específicas realizadas dentro de cada Alfa e Estado de Alfa. Cada atividade tem um objetivo claro e contribui para alcançar os resultados desejados em cada fase do desenvolvimento do software. A estruturação e o gerenciamento eficaz do Espaço de Atividades são essenciais para manter o projeto no caminho certo e garantir a entrega de um produto de alta qualidade como pode ser visto na figura 5.11, a seguir será explicado com mais detalhes cada parte que consiste a imagem.

5.4.1. Área de Interesse: Cliente

5.4.1.1. Engajamento de Usuários e Sensores

1. Identificar e entender necessidades do usuário:
 - Definir e documentar quem são os usuários finais;
 - Coletar e analisar necessidades e expectativas dos usuários;
 - Envolver os usuários no processo de desenvolvimento, garantindo seu feedback contínuo.
2. Configurar e manter sensores:
 - Instalar, calibrar, monitorar e realizar manutenção nos sensores para garantir seu funcionamento adequado.

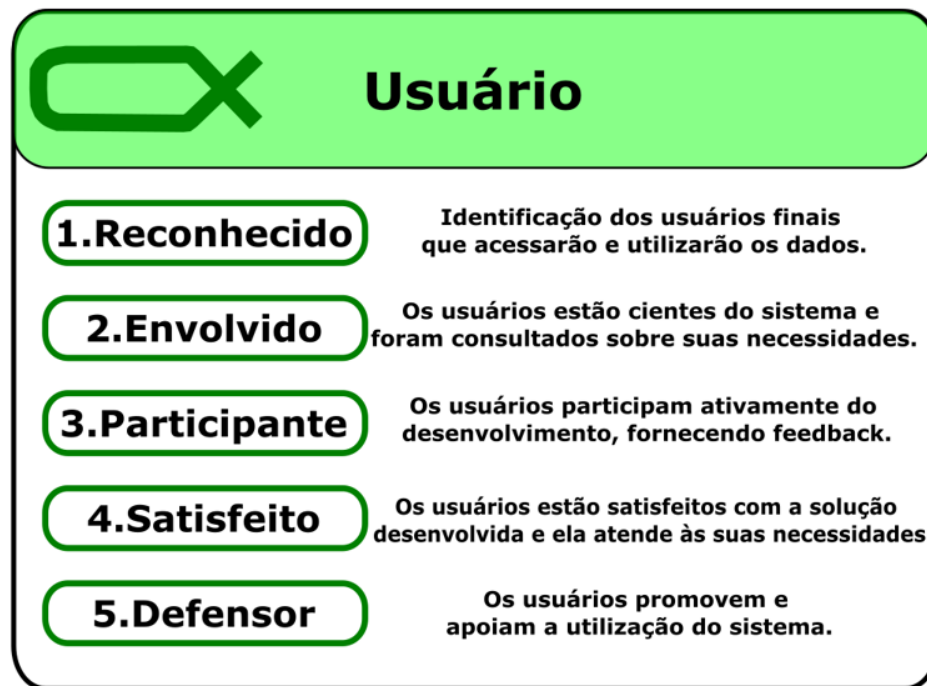


Figure 5.3: Estado Alfa Clientes:Usuário

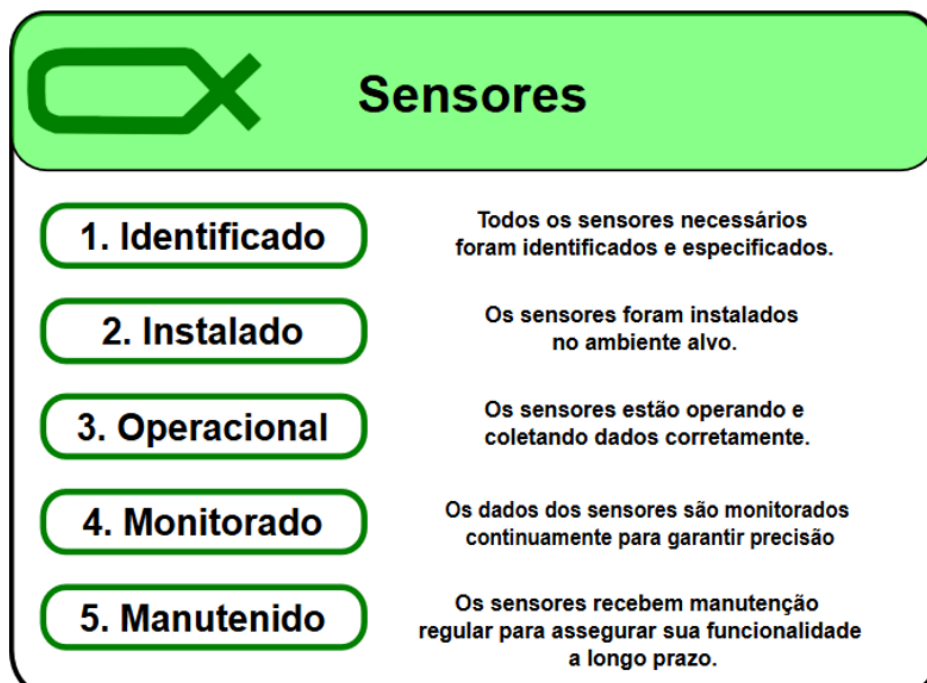


Figure 5.4: Estado Alfa Clientes:Sensores

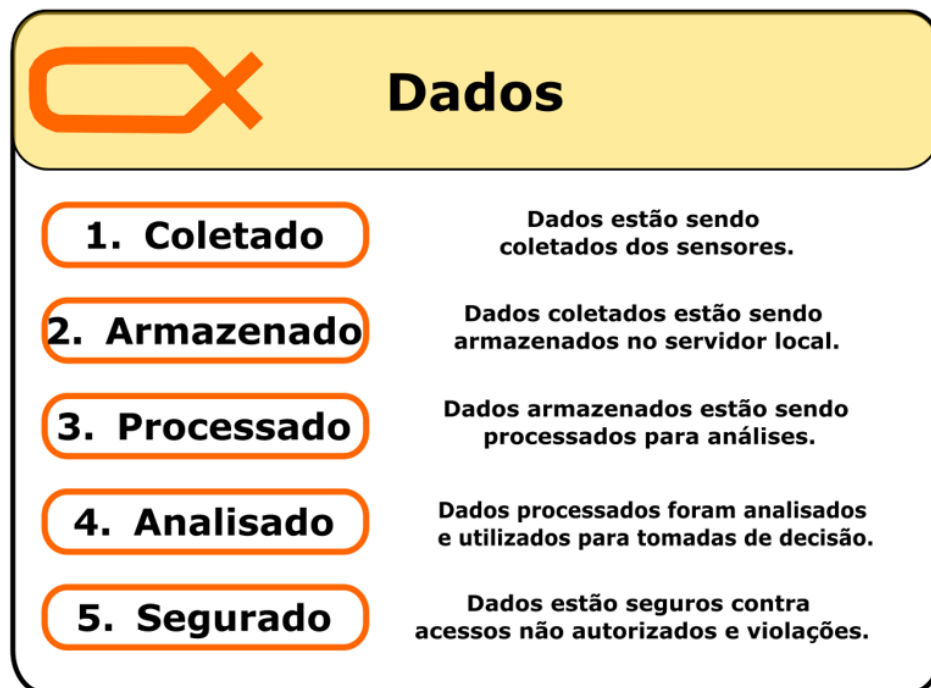


Figure 5.5: Estado Alfa Solução:Dados



Figure 5.6: Estado Alfa Solução:Aplicação

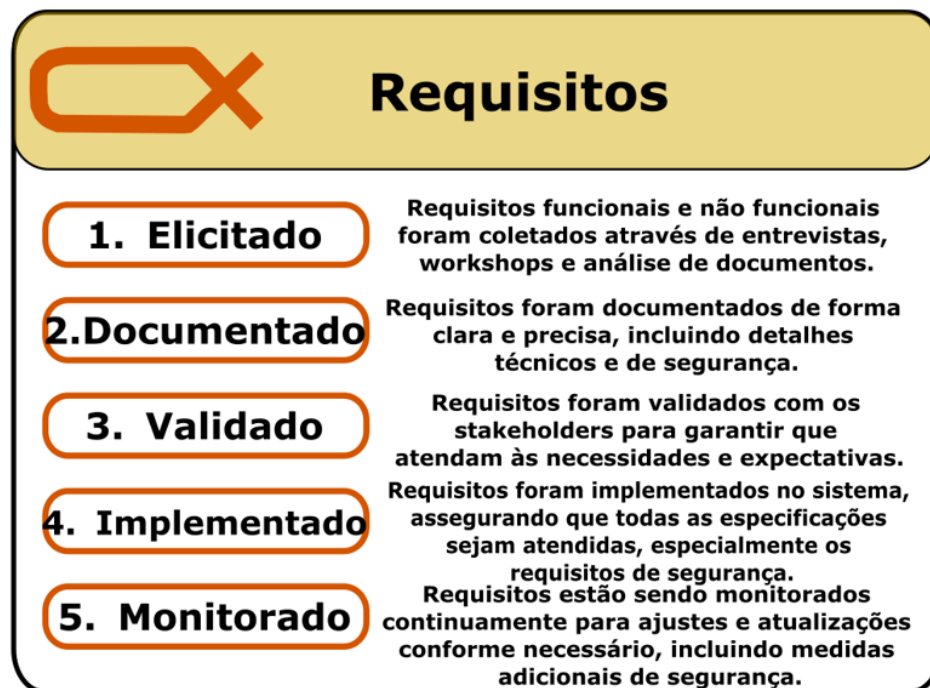


Figure 5.7: Estado Alfa Solução:Requisitos

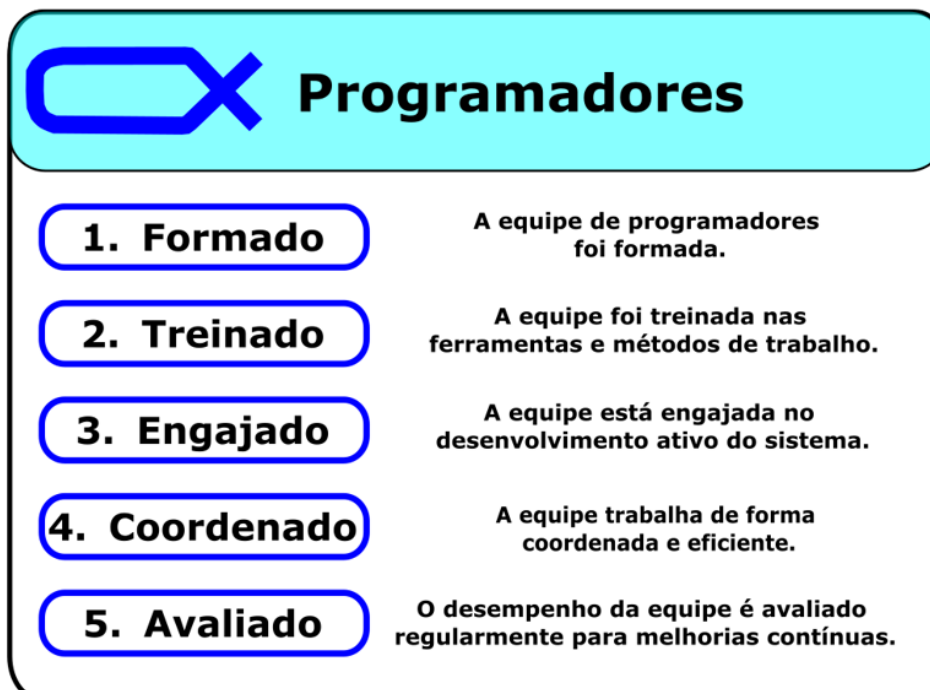


Figure 5.8: Estado Alfa Esforço:Programadores

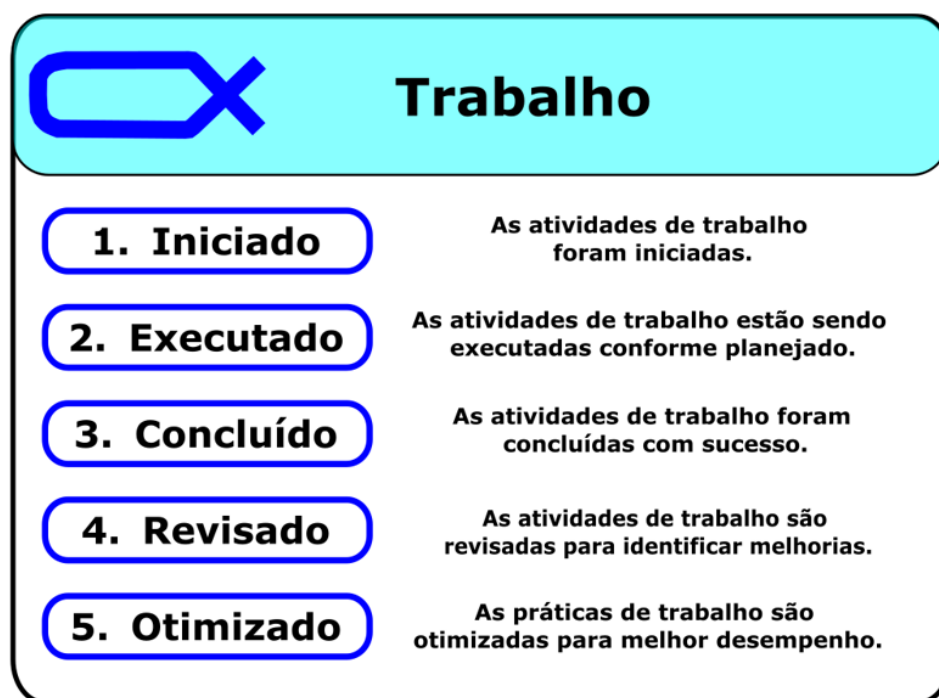


Figure 5.9: Estado Alfa Esforço:Trabalho



Figure 5.10: Estado Alfa Esforço:Método de Trabalho

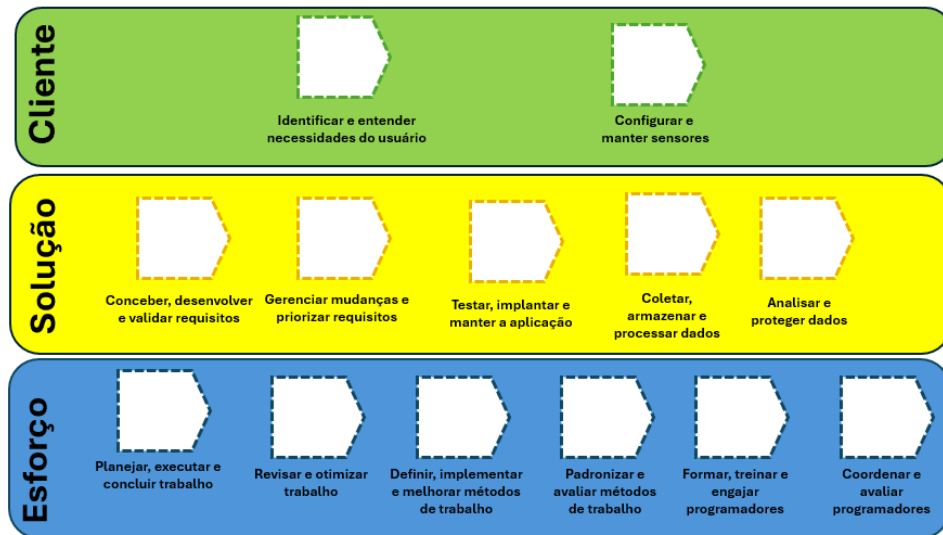


Figure 5.11: Espaço de Atividades

5.4.2. Área de Interesse: Solução

5.4.2.1. Ciclo de Vida da Aplicação e Requisitos

1. Conceber, desenvolver e validar os requisitos:
 - Definir requisitos funcionais e não funcionais da aplicação;
 - Coletar, documentar e aprovar os requisitos com os clientes.
2. Gerenciar mudanças e priorizar os requisitos:
 - Monitorar e ajustar os requisitos conforme necessário;
 - Definir a importância e a ordem de implementação dos requisitos.
3. Testar, implantar e manter a aplicação:
 - Realizar testes para garantir qualidade e funcionalidade;
 - Preparar e realizar a implantação da aplicação em produção;
 - Monitorar e realizar manutenção contínua da aplicação.

5.4.2.2. Gestão de Dados

1. Coletar, armazenar e processar dados:
 - Definir métodos, coletar dados relevantes, implementar mecanismos de armazenamento seguro e eficiente;
 - Analisar e transformar dados para extrair informações valiosas.
2. Analisar e proteger dados:
 - Identificar padrões e gerar insights a partir dos dados;
 - Implementar medidas de segurança para proteger os dados.

5.4.3. Área de Interesse: Esforço

5.4.3.1. Gestão de Trabalho e Métodos

1. Planejar, executar e concluir trabalho:
 - Estabelecer metas e objetivos claros;
 - Realizar as atividades conforme planejado;

- Revisar e validar os resultados.
2. Revisar e otimizar trabalho:
 - Analisar e identificar melhorias;
 - Implementar melhorias para aumentar eficiência.
 3. Definir, implementar e melhorar métodos de trabalho:
 - Documentar processos e procedimentos;
 - Treinar e monitorar a aplicação dos métodos;
 - Coletar feedback e realizar ajustes.
 4. Padronizar e avaliar métodos de trabalho:
 - Documentar e garantir a conformidade com os procedimentos;
 - Medir e revisar a eficácia dos métodos implementados.

5.4.3.2. Gestão de Equipes de Programadores

1. Formar, treinar e engajar programadores:
 - Recrutar programadores com as habilidades necessárias;
 - Fornecer treinamento nas ferramentas e metodologias;
 - Definir papéis e responsabilidades.
2. Coordenar e avaliar programadores:
 - Implementar práticas ágeis e monitorar o progresso;
 - Realizar avaliações de desempenho e coletar feedback.

5.5. Competências

As Competências necessárias abrangem habilidades técnicas, gerenciais e interpessoais. Cada membro da equipe deve possuir competências específicas relacionadas à sua função no projeto, garantindo que todas as atividades sejam realizadas com eficiência e qualidade. A identificação e o desenvolvimento de competências adequadas são fundamentais para o sucesso do projeto e para a aplicação eficaz do framework Essence como pode ser visto na figura 5.12, a seguir será explicado com mais detalhes cada parte que consiste a imagem.



Figure 5.12: Competências

5.5.1. Área de Interesse: Cliente

5.5.1.1. Usuário

Para garantir que o sistema desenvolvido atenda às necessidades dos usuários finais, é crucial compreender profundamente quem são os usuários, quais são suas necessidades e como interagem com o sistema. O envolvimento contínuo dos usuários ao longo do processo de desenvolvimento é essencial para coletar feedback e fazer ajustes conforme necessário.

Competências Necessárias:

1. Pesquisa de Usuários: Habilidade para coletar e analisar as necessidades e expectativas dos usuários.
2. Design Centrado no Usuário: Competência em criar soluções que atendam às necessidades dos usuários.
3. Engajamento de Usuários: Capacidade de envolver os usuários no processo de desenvolvimento e coletar feedback contínuo.

5.5.1.2. Sensores

Os sensores são uma parte integral de muitos sistemas, coletando dados essenciais para o funcionamento do sistema. É importante garantir que os sensores estejam corretamente instalados, mantidos e que os dados coletados sejam precisos e úteis.

Competências Necessárias:

1. Configuração de Hardware: Habilidade para instalar e calibrar sensores.
2. Manutenção de Hardware: Capacidade de monitorar e realizar manutenção em sensores.
3. Análise de Dados de Sensores: Competência para interpretar dados coletados de sensores.

5.5.2. Área de Interesse: Solução

5.5.2.1. Aplicação

O desenvolvimento de uma aplicação robusta envolve várias etapas, desde a concepção até a manutenção contínua. É importante garantir que cada etapa seja realizada com qualidade para atender aos requisitos dos usuários e do negócio.

Competências Necessárias:

1. Análise de Requisitos: Habilidade para definir e documentar requisitos e design da aplicação.
2. Desenvolvimento de Software: Competência em implementar a aplicação conforme os requisitos.
3. Testes de Software: Capacidade de realizar testes unitários, de integração e de sistema.
4. Implantação de Software: Habilidade para preparar e realizar a implantação da aplicação em produção.
5. Manutenção de Software: Competência para monitorar e realizar manutenção contínua da aplicação.

5.5.2.2. Requisitos

Os requisitos são a base de qualquer projeto de software. É essencial que sejam bem definidos, documentados, validados e gerenciados ao longo do ciclo de vida do projeto para garantir que o produto final atenda às expectativas dos usuários e dos clientes.

Competências Necessárias:

1. Elicitação de Requisitos: Habilidade para coletar e documentar os requisitos.
2. Validação de Requisitos: Competência para verificar e aprovar os requisitos com os clientes.
3. Priorização de Requisitos: Habilidade para definir a importância e a ordem de implementação dos requisitos.
4. Gerenciamento de Mudanças: Capacidade de monitorar e ajustar os requisitos conforme necessário.

5.5.2.3. Dados

Os dados são fundamentais para a operação de muitos sistemas. É necessário garantir que os dados sejam coletados, armazenados, processados e analisados de maneira eficaz e segura.

Competências Necessárias:

1. Coleta de Dados: Habilidade para definir métodos e coletar dados relevantes.
2. Armazenamento de Dados: Competência para implementar mecanismos de armazenamento seguro e eficiente.
3. Processamento de Dados: Capacidade de analisar e transformar dados para extrair informações valiosas.
4. Análise de Dados: Habilidade para identificar padrões e gerar insights a partir dos dados processados.
5. Segurança de Dados: Competência para implementar medidas de segurança para proteger os dados.

5.5.3. Área de Interesse: Esforço**5.5.3.1. Trabalho**

A gestão do trabalho envolve desde o planejamento inicial até a execução e revisão das atividades. É importante que o trabalho seja bem organizado e executado de maneira eficiente para alcançar os objetivos do projeto.

Competências Necessárias:

1. Planejamento de Projetos: Habilidade para estabelecer metas e objetivos claros.
2. Execução de Projetos: Competência para realizar as atividades conforme planejado.
3. Revisão de Projetos: Capacidade de revisar e validar os resultados.
4. Análise Pós-Projeto: Habilidade para identificar melhorias.
5. Otimização de Processos: Competência para implementar melhorias para aumentar eficiência.

5.5.3.2. Método de Trabalho

Um método de trabalho bem definido e continuamente melhorado é essencial para garantir que a equipe funcione de maneira eficiente e eficaz. A padronização dos processos ajuda a manter a consistência e a qualidade.

Competências Necessárias:

1. Definição de Processos: Habilidade para documentar processos e procedimentos.
2. Implementação de Processos: Competência para treinar e monitorar a aplicação dos métodos.
3. Melhoria Contínua: Capacidade de coletar feedback e realizar ajustes.
4. Padronização de Processos: Habilidade para garantir a conformidade com os procedimentos.
5. Avaliação de Processos: Competência para medir e revisar a eficácia dos métodos implementados.

5.5.3.3. Programadores

A equipe de programadores é essencial para o desenvolvimento e manutenção do sistema. É importante que a equipe esteja bem formada, treinada, engajada e coordenada para garantir um desenvolvimento eficiente e de alta qualidade.

Competências Necessárias:

1. Formação de Equipes: Habilidade para recrutar programadores com as habilidades necessárias.
2. Treinamento de Equipes: Competência para fornecer treinamento nas ferramentas e metodologias.
3. Engajamento de Equipes: Capacidade de definir papéis e responsabilidades e incentivar a participação ativa.
4. Coordenação de Equipes: Habilidade para implementar práticas ágeis e monitorar o progresso.
5. Avaliação de Desempenho: Competência para realizar avaliações de desempenho e coletar feedback.

Os cards sobre cada uma das Competências abordada e seus respectivos Níveis e requisitos pode ser visto com mais detalhes no link da wiki do projeto (LINK).

6

Vantagens e desvantagens

A modelagem com Essence oferece uma série de vantagens e desvantagens que devem ser consideradas ao adotar este framework em projetos de desenvolvimento de software. Com base na análise das práticas e metodologias empregadas, destaca-se as principais características que impactam diretamente a eficiência e a qualidade dos processos de desenvolvimento.

6.1. Vantagens da Modelagem Essence

1. **Foco na Essência:** A abordagem centrada em essência permite um entendimento claro e conciso dos aspectos fundamentais do projeto desde o início, facilitando uma melhor gestão de requisitos e minimizando retrabalhos.
2. **Flexibilidade:** A estrutura modular do Essence permite adaptações conforme as necessidades específicas do projeto, promovendo a agilidade e a capacidade de resposta a mudanças durante todo o ciclo de vida do software.
3. **Avaliação Contínua:** A capacidade de avaliar continuamente a saúde do projeto através dos Kernel e Alphas oferece uma visibilidade clara do progresso e dos riscos potenciais, permitindo ajustes proativos.
4. **Comunicação Eficaz:** A linguagem comum fornecida pelo Essence facilita a comunicação entre todos os membros da equipe, incluindo stakeholders não técnicos, promovendo uma compreensão mais clara e alinhada dos objetivos do projeto.

6.2. Desvantagens da Modelagem Essence

1. **Curva de Aprendizado:** A adoção inicial do framework pode exigir um período de aprendizado significativo para a equipe, especialmente aqueles não familiarizados com metodologias centradas em essência.
2. **Complexidade Percebida:** A estruturação detalhada do Essence pode parecer excessivamente complexa para projetos menores ou equipes menos experientes, exigindo adaptações para garantir uma implementação eficaz.
3. **Customização Necessária:** Embora flexível, o Essence pode exigir customizações extensivas para se adequar perfeitamente aos contextos específicos de projeto, adicionando complexidade ao processo inicial de implementação.

7

Conclusão

Ao longo deste estudo sobre a modelagem com Essence no cenário de desenvolvimento de software, foi explorado os princípios, práticas e impactos dessa abordagem. A análise detalhada das competências necessárias, vantagens e desvantagens revelou insights significativos sobre como o Essence pode ser aplicado eficazmente para melhorar a qualidade e a eficiência dos projetos.

7.1. Resumo das Principais Considerações

1. **Eficiência e Qualidade:** A modelagem com Essence demonstrou ser uma metodologia robusta para garantir a eficiência operacional e a qualidade final dos produtos de software, fornecendo ferramentas e estruturas claras para guiar o desenvolvimento desde a concepção até a entrega.
2. **Adaptação e Flexibilidade:** A capacidade de adaptar o framework às necessidades específicas do projeto oferece uma vantagem significativa, permitindo uma resposta ágil às mudanças nos requisitos e no ambiente operacional.
3. **Desafios e Oportunidades:** Embora apresente desafios iniciais, como a curva de aprendizado e a necessidade de customização, o Essence abre oportunidades para melhorar a colaboração interdisciplinar, a transparência do processo e a qualidade do produto final.

7.2. Considerações Finais

Em conclusão, a modelagem com Essence representa não apenas uma metodologia estruturada para o desenvolvimento de software, mas também um conjunto de princípios que promovem uma abordagem sistêmica e colaborativa para alcançar resultados superiores. Ao implementar o Essence, equipes podem não apenas melhorar a gestão de projetos, mas também elevar o padrão de entrega de software, alinhando-se de forma mais eficaz às expectativas dos clientes e usuários finais.

Este estudo oferece uma base sólida para futuras pesquisas e implementações práticas do Essence, incentivando uma reflexão contínua sobre as melhores práticas no desenvolvimento de software moderno.

Referências

- [1] OM Group et al. Omg unified modeling language (omg uml), superstructure. *Open Management Group*, 2009.
- [2] Ivar Jacobson, Pan-Wei Ng, Paul E McMahon, Ian Spence, and Svante Lidman. The essence of software engineering: the semat kernel. *Communications of the ACM*, 55(12):42–49, 2012.
- [3] Business Process Model. Notation (bpmn) version 2.0. *OMG Specification, Object Management Group*, 19:52–60, 2011.
- [4] SPEM OMG and O Notation. Software & systems process engineering meta-model specification. *OMG Std., Rev*, 2:18–71, 2008.