

SCESPE

VISÃO DO PRODUTO E PROJETO

Versão 1

Histórico de Revisão

<i>Data</i>	<i>Versão</i>	<i>Descrição</i>	<i>Autor</i>
<i>02/04/2024</i>	<i>0.1</i>	<i>Criação do documento</i>	<i>Henrique Quenino Vinicius Santos André Silva Bruno de Menezes Eduardo Sandes Larissa Vieira Carlos Eduardo Alves</i>
<i>04/04/2024</i>	<i>0.2</i>	<i>Evolução do documento</i>	<i>Henrique Quenino Vinicius Santos André Silva Bruno de Menezes Eduardo Sandes Larissa Vieira Carlos Eduardo Alves</i>
<i>08/04/2024</i>	<i>1</i>	<i>Visão Geral do Produto</i>	<i>Henrique Quenino Vinicius Santos André Silva Bruno de Menezes Eduardo Sandes Larissa Vieira Carlos Eduardo Alves</i>

Sumário

1	<i>VISÃO GERAL DO PRODUTO</i>	6
1.1	Problema	6
1.2	Declaração de Posição do Produto	6
1.3	Objetivos do Produto	7
1.4	Tecnologias a Serem Utilizadas	7
2	<i>VISÃO GERAL DO PROJETO</i>	7
2.1	Organização do Projeto	8
2.2	Planejamento das Fases e/ou Iterações do Projeto	8
2.3	Matriz de Comunicação	9
2.4	Gerenciamento de Riscos	9
2.5	Critérios de Replanejamento	9
3	<i>PROCESSO DE DESENVOLVIMENTO DE SOFTWARE</i>	10
3.1	Metodologia	10
3.1.1	Requisitos.....	10
3.1.2	Equipe de Desenvolvimento	11
3.1.3	Usuários	12
3.1.4	Tipo de Projetos e Riscos Associados	13
3.1.5	Escolha do Processo de Desenvolvimento de Software	15
3.2	Ferramentas	15
3.3	Processos e Procedimentos	16
3.3.1	Escolha do Processo de Engenharia de Requisitos	16
3.3.2	Escolha do Processo de Engenharia de Requisitos	17
3.3.3	Atividades RAD.....	18

4	<i>LIÇÕES APRENDIDAS</i>	21
4.1	Unidade 1	21
4.2	Unidade 2	21
4.3	Unidade 3	22
4.4	Unidade 4	22
5	<i>REFERÊNCIAS BIBLIOGRÁFICAS</i>	22

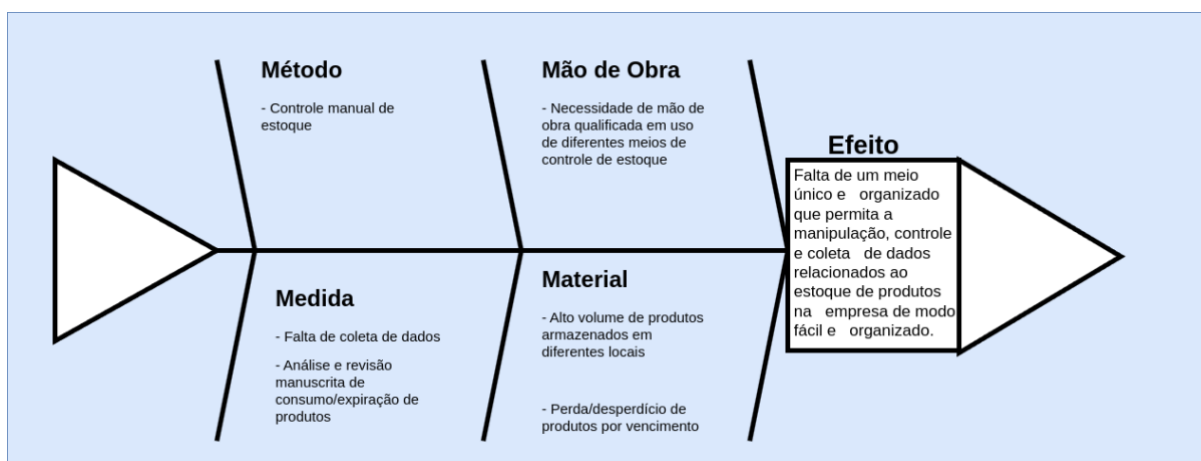
VISÃO DO PRODUTO E PROJETO

1 VISÃO GERAL DO PRODUTO

1.1 Problema

O problema se constitui, na organização de um depósito central de nove unidades da rede de Ensino Biângulo, cuja gestão implica o controle rigoroso da entrada e saída de suprimentos e produtos. É crucial a emissão de etiquetas com códigos de barras individuais para cada item recebido, viabilizando sua identificação, bem como o acompanhamento minucioso das quantidades armazenadas. Além disso, é fundamental a geração de dados estatísticos para uma gestão mais eficaz. Destaca-se que os produtos podem ser distribuídos para todas as unidades conforme suas respectivas necessidades e os pedidos realizados.

Figura 1 - Diagrama de Ishikawa



Fonte: Eduardo Sandes (2024)

1.2 Declaração de Posição do Produto

Para	Rede Colégio Biângulo
Quem	Necessita de gerir eficientemente o depósito das várias unidades do colégio
O (nome do produto)	É um sistema eficiente de controle de estoque
Que	Auxílio na gestão de depósito, de forma eficiente e organizada, integrando o estoque das unidades para uma análise geral e específica
Ao contrário	Planilhas digitais, manuscrita ou afins
Nosso produto	Ofertará estoque controlado e eficiente

1.3 Objetivos do Produto

Tem-se por objetivo desenvolver um software eficiente e intuitivo para a gestão de estoque em um depósito central de uma rede de escolas com nove unidades, permitindo o controle detalhado dos suprimentos, movimentação de itens entre unidades, geração de relatórios e integração com sistemas de compra e fornecedores.

1.4 Tecnologias a Serem Utilizadas

[Forneça o objetivo principal do projeto, e objetivos secundários (caso haja)]

2 VISÃO GERAL DO PROJETO

2.1 Organização do Projeto

[apresentada a divisão de atribuições e responsabilidades entre os membros do projeto, sem qualquer relação de hierarquia ou grau de importância. Todos os integrantes são igualmente importantes e responsáveis pelo sucesso do projeto.]

<i>Papel</i>	<i>Atribuições</i>	<i>Responsável</i>	<i>Participantes</i>
<i>Desenvolvedor</i>	<i>Codificar o produto, codificar testes unitários, realizar refatoração</i>	<i>Maria</i>	<i>José, João, Laura, Sandra</i>
<i>Dono do Produto</i>	<i>Atualizar o escopo do produto, organizar o escopo das sprints, validar as entregas</i>	<i>Sandra</i>	<i>Maria, José, João, Laura</i>
<i>Analista de Qualidade</i>	<i>Garantir a qualidade do produto, garantir o cumprimento do conceito de pronto, realizar inspeções de código</i>	<i>José</i>	<i>Maria, Sandra, Laura, João</i>
<i>Cliente</i>	<i>...</i>	<i>...</i>	<i>...</i>

2.2 Planejamento das Fases e/ou Iterações do Projeto

[Registrar o projeto, as fases de seu ciclo de vida e suas iterações, especificando suas datas de início e de fim, bem como os produtos a serem gerados.

O planejamento do projeto deve ser **atualizado de maneira sucessiva**, a partir da realização de cada ciclo (sprint, iteração) de desenvolvimento]

Deve ser apresentado o planejamento de **TODO o projeto**, ainda que, preliminar. Ao longo da disciplina, tal planejamento deverá ser atualizado.

<i>Sprint</i>	<i>Produto (Entrega)</i>	<i>Data Início</i>	<i>Data Fim</i>
<i>Sprint 1</i>	<i>Definição do Produto</i>	<i>18/01/22</i>	<i>03/02/22</i>
<i>Sprint 2</i>	<i>MVP e Planejamento do Projeto</i>	<i>08/02/22</i>	<i>24/02/22</i>
<i>Sprint 3</i>	<i>Funcionalidades A, B, C, D</i>	<i>01/03/22</i>	<i>17/02/22</i>

<i>Sprint 4</i>	<i>Funcionalidades E, F e G</i>	<i>22/03/22</i>	<i>07/04/22</i>
...

2.3 Matriz de Comunicação

[Esta seção descreve a estratégia de comunicação adotada para monitoramento do progresso do projeto. Identificar a periodicidade de reuniões e o envio dos relatórios exigidos pelo processo e opcionalmente outros relatórios exigidos pelo cliente.]

<i>Descrição</i>	<i>Área/ Envolvidos</i>	<i>Periodicidade</i>	<i>Produtos Gerados</i>
<i>- Acompanhamento das Atividades em Andamento</i> <i>- Acompanhamento dos Riscos, Compromissos, Ações Pendentes, Indicadores</i>	<i>- Equipe do Projeto</i>	<i>- Semanal</i>	<i>- Ata de reunião</i> <i>- Relatório de situação do projeto</i>
		<i>- Quinzenal</i>	
<i>- Comunicar situação do projeto</i>	<i>- Equipe</i> <i>- Prof.</i>	<i>- Semanal</i>	<i>- Ata de reunião e</i> <i>- Relatório de Situação do Projeto</i>

2.4 Gerenciamento de Riscos

[Para o Gerenciamento de Riscos devem ser realizadas tarefas, como:

- Identificar todos os riscos possíveis e detectáveis em cada fase do projeto;*
- Executar as ações para mitigar os riscos que tenham muita exposição ao risco caso este ocorra na Lista de Riscos do Projeto;*
- Fazer uma revisão da lista dos riscos periodicamente, com o propósito de averiguar uma possível incidência de um risco e ver se há outros riscos ainda não relatados;*
- Em caso de confirmação de um risco previsto, agir no sentido de contingenciá-lo conforme programado;*
- Registrar os riscos no Painel de Controle do Projeto e no Plano do Projeto (Riscos iniciais);]*

Os riscos do projeto devem ser acompanhados e atualizados a cada ciclo.

2.5 Critérios de Replanejamento

[Descrever os critérios de replanejamento que serão utilizados, caso seja necessário realizá-lo no projeto.]

Os critérios de replanejamento do projeto devem ser acompanhados e atualizados a cada ciclo. E, aplicados, conforme necessidade.

3 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

3.1 Metodologia

Com base em Gupta (2019), foram respondidas uma série de perguntas, relacionadas a temas específicos para que seja possível definir qual o processo que melhor se encaixa no projeto. Temas:

1. Requisitos
2. Equipe de Desenvolvimento
3. Usuários
4. Tipo de Projetos e Riscos Associados

3.1.1 Requisitos

Figura 2 - Modelo Gupta para requisitos

Table 3.1 : *Selection of a model based on characteristics of requirements*

<i>Requirements</i>	<i>Waterfall</i>	<i>Prototype</i>	<i>Iterative Enhancement</i>	<i>Evolutionary development</i>	<i>Spiral</i>	<i>RAD</i>
Are requirements easily understandable and defined?	Yes	No	No	No	No	Yes
Do we change requirements quite often?	No	Yes	No	No	Yes	No
Can we define requirements early in the cycle?	Yes	No	Yes	Yes	No	Yes
Requirements are indicating a complex system to be built	No	Yes	Yes	Yes	Yes	No

fonte: Gupta (2019)

Pergunta	Nosso contexto
Os requisitos são fáceis de entender e definir?	Sim
Nós mudamos os requisitos com bastante frequência?	Não
Nós podemos definir os requisitos ao início de cada ciclo?	Sim
Os requisitos estão indicando um sistema complexo para se construir?	Sim

Conclusão: Em questão de requisitos, temos o indicativo para a cascata, iterativo incremental, desenvolvimento evolutivo e RAD.

3.1.2 Equipe de Desenvolvimento

Figura 3 - Modelo Gupta para equipe de desenvolvimento

Table 3.2 : Selection based on status of development team

<i>Development team</i>	<i>Waterfall</i>	<i>Prototype</i>	<i>Iterative enhancement</i>	<i>Evolutionary development</i>	<i>Spiral</i>	<i>RAD</i>
Less experience on similar projects	No	Yes	No	No	Yes	No
Less domain knowledge (new to the technology)	Yes	No	Yes	Yes	Yes	No
Less experience on tools to be used	Yes	No	No	No	Yes	No
Availability of training, if required	No	No	Yes	Yes	No	Yes

Fonte: Gupta (2019)

Pergunta	Nosso contexto
Pouca experiência em projetos similares?	Sim
Pouco conhecimento de domínio (novato na tecnologia)?	Sim
Pouca experiência com as ferramentas que serão usadas?	Não
Disponibilidade para treinamento, se necessário	Sim

Conclusão: Em questão de equipe de desenvolvimento, temos o indicativo para o desenvolvimento evolutivo e iterativo incremental.

3.1.3 Usuários

Figura 4 - Modelo Gupta para usuários

Table : 3.3 : Selection based on user's participation

<i>Involvement of Users</i>	<i>Waterfall</i>	<i>Prototype</i>	<i>Iterative enhancement</i>	<i>Evolutionary development</i>	<i>Spiral</i>	<i>RAD</i>
User involvement in all phases	No	Yes	No	No	No	Yes
Limited user participation	Yes	No	Yes	Yes	Yes	No
User have no previous experience of participation in similar projects	No	Yes	Yes	Yes	Yes	No
Users are experts of problem domain	No	Yes	Yes	No	No	Yes

Fonte: Gupta (2019)

Pergunta	Nosso contexto
Usuário está envolvido em todas as fases?	Sim
Participação limitada do usuário?	Não
Usuário não tem experiência anterior em participação em projetos similares?	Não
Usuário são especialistas no domínio do problema?	Sim

Conclusão: Em questão de usuário, temos o indicativo para o RAD.

3.1.4 Tipo de Projetos e Riscos Associados

Figura 5 - Modelo Gupta para tipo de projeto e risco associado

Table 3.4 : Selection based on type of project with associated risk

<i>Project type and risk</i>	<i>Waterfall</i>	<i>Prototype</i>	<i>Iterative enhancement</i>	<i>Evolutionary development</i>	<i>Spiral</i>	<i>RAD</i>
Project is the enhancement of the existing system	No	No	Yes	Yes	No	Yes
Funding is stable for the project	Yes	Yes	No	No	No	Yes
High reliability requirements	No	No	Yes	Yes	Yes	No
Tight project schedule	No	Yes	Yes	Yes	Yes	Yes
Use of reusable components	No	Yes	No	No	Yes	Yes
Are resource (time, money people etc.) scarce ?	No	Yes	No	No	Yes	No

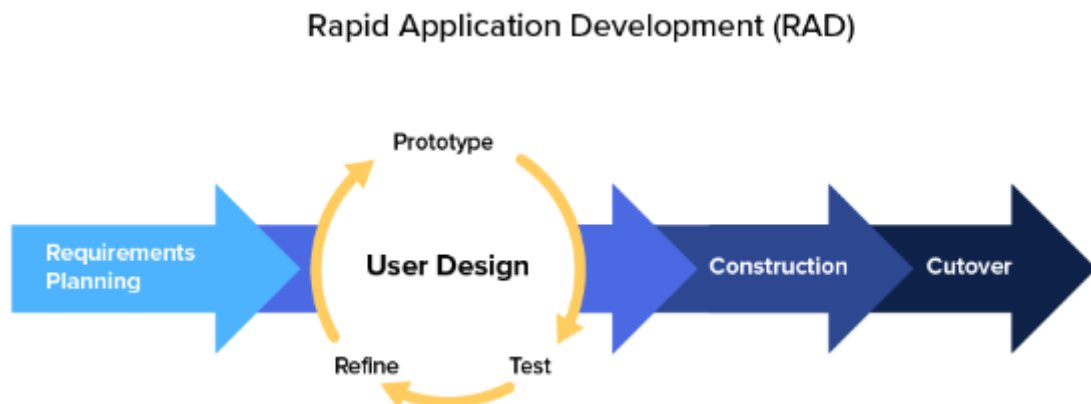
Fonte: Gupta (2019)

Pergunta	Nosso contexto
O projeto é melhoria de um sistema existente?	Não
O financiamento está estável para o projeto?	Sim
Requisitos de alta confiabilidade?	Sim
Cronograma do projeto é apertado?	Sim
Uso de componentes reutilizáveis?	Sim
Os recursos (Tempo, dinheiro, pessoas etc.) estão escassos?	Não

Conclusão: Em termos de Tipo de Projeto e Risco Associado, temos o indicativo para o protótipo, espiral e RAD.

3.1.5 Escolha do Processo de Desenvolvimento de Software

Figura 6 - Procedimento do RAD



Fonte: Marsicano (2019)

A escolha do método para este projeto foi guiada pelo método proposto por Gupta (2019), em que respondemos algumas questões e escolhemos o processo que mais se adequa ao objetivo do projeto.

Considerando o foco no usuário, a necessidade de uma interação eficiente com o sistema, as restrições de tempo impostas pelo limitado prazo estipulado e também a relativa inexperiência da equipe, a escolha foi o RAD (Rapid Application Development). Apesar da natureza variável dos requisitos no processo RAD, optamos por estabelecer a maioria deles no início do projeto, visando garantir uma base sólida para sua evolução, mas com espaço para mudanças controladas nos requisitos.

3.2 Ferramentas

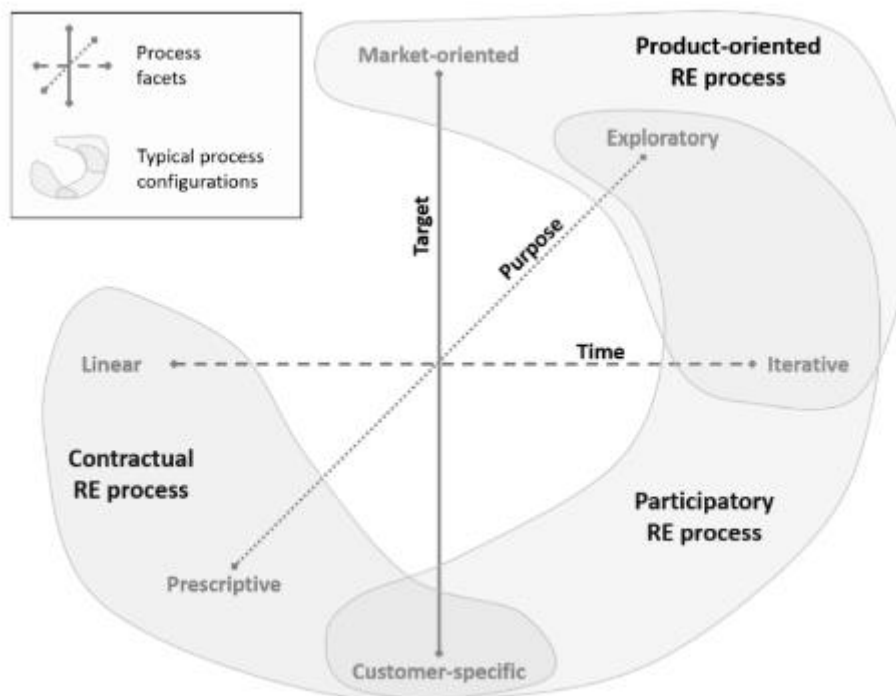
Para que seja possível a execução do processo escolhido, foi escolhida as seguintes ferramentas para desenvolvimento, controle e organização da equipe:

- **Comunicação:** WhatsApp e Discord;
- **Interface de Desenvolvimento:** Visual Studio Code;
- **Gerenciamento de tarefas:** Sharepoint e Miro;
- **Prototipação:** Figma;
- **Versionamento:** Git e Github;
- **Desenvolvimento de ideias:** Miro;

3.3 Processos e Procedimentos

3.3.1 Escolha do Processo de Engenharia de Requisitos

Figura 7 - Faceta da engenharia de requisitos



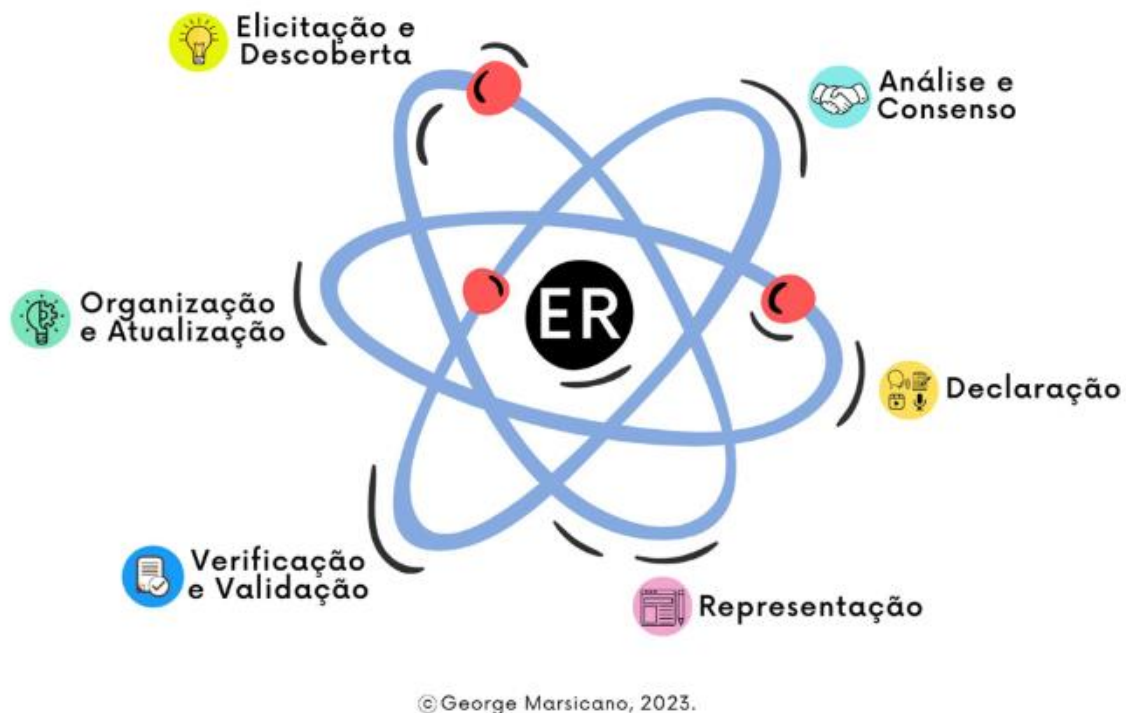
Fonte: Marsicano (2023)

Baseado nas Facetas do Processo de ER (IREB, 2022), foi definido que é um processo de ER Contratual, pois:

- **Alvo:** Cliente específico;
- **Propósito:** Prescritivo;
- **Tempo:** Linear.

3.3.2 Escolha do Processo de Engenharia de Requisitos

Figura 8 - Atividades da Engenharia de Requisitos



Fonte: Marsicano (2023)

Com o que foi definido nos tópicos anteriores, podemos incluir as atividades de ER (MARSICANO, 2023) dentro dos procedimentos do processo de desenvolvimento de software RAD.

3.3.3 Atividades RAD

O RAD é um método para desenvolvimento de softwares que permite uma prototipagem mais rápida e entrega iterativa do produto. Trata-se de um modelo alternativo ao tradicional modelo cascata que, em geral, foca em um processo de desenvolvimento sequencial e pouco flexível. Ele enfatiza um ciclo de desenvolvimento curto com o objetivo de ter um desenvolvimento melhor e mais rápido (Redação Cronapp, 2017).

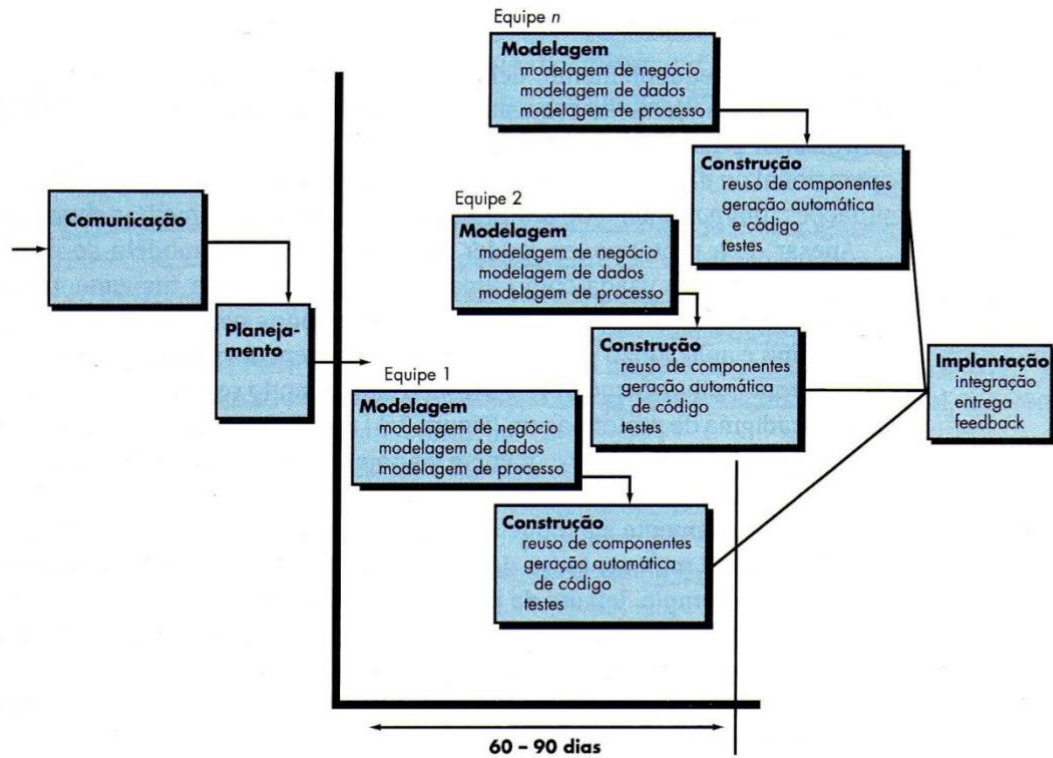
Este processo permite que o software seja desenvolvido em partes menores e mais gerenciáveis, o que facilita a implementação de feedbacks constantes e a realização de alterações.

O RAD é uma metodologia flexível, capaz de suportar mudanças no escopo, novos recursos e alterações facilmente. Além disso, trabalha com a reutilização de componentes do software, o que ajuda a otimizar o processo de desenvolvimento, garantindo uma entrega mais rápida e uma redução no desperdício de tempo e dados. (Noleto, 2020)

Nós, como grupo, estamos comprometidos em utilizar os feedbacks constantes para melhorar e refinar nosso software. Isso significa que estaremos sempre ouvindo e aprendendo com nossos usuários, colegas de equipe e partes interessadas para garantir que nosso software atenda às suas necessidades e expectativas.

Por fim, daremos grande ênfase ao levantamento de requisitos. Entender claramente o que nossos usuários precisam e esperam de nosso software é crucial para o seu sucesso. Portanto, passaremos um tempo considerável garantindo que tenhamos uma compreensão clara e completa dos requisitos antes de começarmos a codificar, além de buscar validações constantes.

Figura 9 - Modelo RAD



Fonte: Univas (2013)

3.3.3.1 Requisitos e planejamento

3.3.3.1.1 Planejamento de requisitos

Envolve o uso de técnicas utilizadas para o levantamento dos requisitos, como brainstorming, análise de tarefas, análise de formulários, cenários de usuário, FAST (Facilitated Application Development Technique), etc. (geeksforgeeks, 2024)

<i>Identificador</i>	<i>Atividade</i>	<i>Método</i>	<i>Ferramenta</i>	<i>Entrega</i>
<i>PR-1</i>	<i>Brainstorming</i>	<i>Reunião</i>	<i>Miro e Discord</i>	<i>Entendimento</i>

				<i>inicial dos requisitos a serem desenvolvidos em um ciclo</i>
<i>PR-2</i>	<i>Análise das funcionalidades</i>	<i>Reunião</i>	<i>Miro e Discord</i>	<i>Análise do custo de cada requisito</i>

3.3.3.1.2 Descrição do usuário

Esta fase consiste em receber feedback do usuário e construir o protótipo usando ferramentas de desenvolvedor. Por outras palavras, inclui o reexame e a validação dos dados recolhidos na primeira fase. Os atributos do conjunto de dados também são identificados e elucidados nesta fase. (geeksforgeeks, 2024)

<i>Identificador</i>	<i>Atividade</i>	<i>Método</i>	<i>Ferramenta</i>	<i>Entrega</i>
<i>DU-1</i>	<i>Elicitação de requisitos</i>	<i>Reunião com cliente</i>	<i>Discord</i>	<i>Definição de requisitos a serem entregues</i>
<i>DU-2</i>	<i>Prototipação</i>	<i>Pareamento</i>	<i>Figma e Discord</i>	<i>Protótipos de alta e baixa fidelidade</i>
<i>DU-3</i>	<i>Verificação e Validação</i>	<i>Reunião com cliente</i>	<i>Figma e Discord</i>	<i>Validação ou reprovação dos protótipos</i>

3.3.3.1.3 Construção

Nesta fase ocorre o refinamento do protótipo e a entrega. Inclui o uso real de ferramentas automatizadas para transformar processos e modelos de dados a fim de desenvolver o produto final. Todas as modificações e melhorias necessárias devem ser feitas nesta fase. (geeksforgeeks, 2024)

<i>Identificador</i>	<i>Atividade</i>	<i>Método</i>	<i>Ferramenta</i>	<i>Entrega</i>
<i>CT-1</i>	<i>Codificação</i>	<i>Pair Programming</i>	<i>Visual Studio Code Live Share</i>	<i>Requisito Funcional</i>
<i>CT-2</i>	<i>Integração</i>	<i>Git merge</i>	<i>GitHub</i>	<i>Todas as funcionalidades operantes e sem conflito</i>

3.3.3.1.4 Cutover (teste e validação)

Todas as interfaces entre os módulos independentes desenvolvidos por equipes separadas devem ser testadas adequadamente. O uso de ferramentas e subpartes altamente automatizadas facilita os testes. Isto é seguido por testes de aceitação pelo usuário. (geeksforgeeks, 2024)

<i>Identificador</i>	<i>Atividade</i>	<i>Método</i>	<i>Ferramenta</i>	<i>Entrega</i>
<i>CO-1</i>	<i>Testes das funcionalidades</i>	<i>Testes unitários</i>	<i>Visual Studio Code e unittest</i>	<i>Garantia efetiva da qualidade das funcionalidades a serem entregues</i>
<i>CO-2</i>	<i>Validação das funcionalidades</i>	<i>Apresentação remota e testes de aceitação</i>	<i>Discord</i>	<i>Validação, reprovação ou reformulação de requisitos entregues.</i>

4 LIÇÕES APRENDIDAS

4.1 Unidade 1

[Liste as lições aprendidas na retrospectiva, com ênfase especial nas ações a serem tomadas para melhorar, por exemplo: o ambiente de desenvolvimento, o processo ou a colaboração da equipe.]

[Aqui, também devem estar registradas as dificuldades que a equipe teve durante a unidade, e como conseguiu (ou não as superar).]

4.2 Unidade 2

[Liste as lições aprendidas na retrospectiva, com ênfase especial nas ações a serem tomadas para melhorar, por exemplo: o ambiente de desenvolvimento, o processo ou a colaboração da equipe.]

[Aqui, também devem estar registradas as dificuldades que a equipe teve durante a unidade, e como conseguiu (ou não as superar).]

4.3 Unidade 3

[Liste as lições aprendidas na retrospectiva, com ênfase especial nas ações a serem tomadas para melhorar, por exemplo: o ambiente de desenvolvimento, o processo ou a colaboração da equipe.]

[Aqui, também devem estar registradas as dificuldades que a equipe teve durante a unidade, e como conseguiu (ou não as superar).]

4.4 Unidade 4

[Liste as lições aprendidas na retrospectiva, com ênfase especial nas ações a serem tomadas para melhorar, por exemplo: o ambiente de desenvolvimento, o processo ou a colaboração da equipe.]

[Aqui, também devem estar registradas as dificuldades que a equipe teve durante a unidade, e como conseguiu (ou não as superar).]

5 REFERÊNCIAS BIBLIOGRÁFICAS

MARSICANO, George. Slides: Escolhas da Engenharia de Software. 2023. Disponível em:

https://aprender3.unb.br/pluginfile.php/2518488/mod_folder/content/0/Unidade%201%20-%20Aula%20-%20Escolhas%20da%20ESW.pdf

Raja Gupta. Fundamentals of Software Engineering. Engineering Handbook. 2019

Redação Cronapp: Como aplicar o RAD no desenvolvimento de softwares. 21 de julho de 2017. Disponível em: <https://blog.cronapp.io/como-aplicar-o-rad-no-desenvolvimento-de-softwares/#:~:text=O%20RAD%20%C3%A9%20um%20m%C3%A9todo,desenvolvimento%20sequencial%20e%20pouco%20flex%C3%A9vel.>

Betrybe, Noletto Cairo. RAD: conheça o desenvolvimento rápido de aplicação! 02 de agosto de 2020. Disponível em: [RAD: conheça o desenvolvimento rápido de aplicação! – Insights para te ajudar na carreira em tecnologia | Blog da Trybe \(betrybe.com\)](https://betrybe.com/insights/para-te-ajudar-na-carreira-em-tecnologia/)

UNIVAS. Metodologias clássicas. 2013. Disponível em:

<https://metodologiasclassicas.blogspot.com/p/incremental.html>

Rapid application development model (RAD) – Software Engineering. GeeksforGeeks,

2024. Disponível em: <https://www.geeksforgeeks.org/software-engineering-rapid-application-development-model-rad/>