

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
FACULDADE DE TECNOLOGIA DE MAUÁ**

**GABRIEL TAVARES LORENA
LEONARDO RODRIGUES SILVA
RENAN COELHO DA SILVA MENEZES**

DESENVOLVIMENTO ÁGIL DE BANCO DE DADOS

MAUÁ/SP

2024

**GABRIEL TAVARES LORENA
LEONARDO RODRIGUES SILVA
RENAN COELHO DA SILVA MENEZES**

DESENVOLVIMENTO ÁGIL DE BANCO DE DADOS

Monografia apresentada à FATEC Mauá,
como parte dos requisitos para obtenção do
Título de Tecnólogo em Informática para
Negócios.

Orientador: Prof. M. Sc. Osmil Aparecido
Morselli.

MAUÁ/SP

2024

Catálogo-na-Publicação – Biblioteca Fatec Mauá

005.74

L869d Lorena, Gabriel Tavares.

Desenvolvimento ágil de banco de dados / Gabriel Tavares
Lorena, Leonardo Rodrigues Silva, Renan Coelho da Silva
Menezes. – 2024.

57 p. : il. ; 30 cm.

Orientador: Prof. Me. Osmil Aparecido Morselli.

Trabalho de conclusão de curso (Curso Superior de Tecnologia
em Informática para Negócios) – Faculdade de Tecnologia de
Mauá.

Referências: p. 54-57.

1. Banco de dados. 2. Cascata. 3. Metodologia ágil. 4.
Metodologia tradicional. 5. *Scrum*. I. Silva, Leonardo Rodrigues. II.
Menezes, Renan Coelho da Silva. III. Morselli, Osmil Aparecido. IV.
Título.

CDD 23. : Banco de dados 005.74

Elaborada por Tatiana Sambinelli CRB-8 SP-011003/O

**GABRIEL TAVARES LORENA
LEONARDO RODRIGUES SILVA
RENAN COELHO DA SILVA MENEZES**

DESENVOLVIMENTO ÁGIL DE BANCO DE DADOS

Monografia apresentada à FATEC Mauá,
como parte dos requisitos para obtenção do
Título de Tecnólogo em Informática para
Negócios.

Aprovação em: 18 jun. 2024.

Prof. M. Sc. Osmil Aparecido Morselli.
FATEC Mauá
Orientador

Prof(a). Renata Rodrigues
FATEC Mauá
Avaliador(a)

Prof(a). Nizi Voltareli Morselli
FATEC Mauá
Avaliador(a)

AGRADECIMENTOS

Gostaríamos de expressar nossa sincera gratidão a todos aqueles que contribuíram para a realização deste Trabalho de Conclusão de Curso (TCC). Esta jornada acadêmica foi desafiadora, mas cheia de aprendizado e não teria sido possível sem o apoio e incentivo de algumas pessoas especiais.

Primeiramente, gostaríamos agradecer às nossas famílias, que sempre estiveram ao nosso lado, apoiando-nos em todas as etapas deste caminho. O amor incondicional e compreensão foram fundamentais para nossa motivação e perseverança.

Aos nossos professores, quero expressar minha profunda gratidão. Suas orientações, conhecimento e *feedback* foram inestimáveis. Cada um de vocês desempenhou um papel crucial no nosso desenvolvimento acadêmico e somos gratos por termos tido a oportunidade de aprender com vocês.

Por fim, gostaríamos de agradecer a todos os que, de alguma forma, contribuíram para este trabalho. Seu apoio, paciência e encorajamento foram fundamentais.

Este TCC representa não apenas nossa conquista acadêmica, mas também a recompensa de todos os esforços coletivos. Muito obrigado por fazerem parte desta realização.

Com gratidão - Gabriel Tavares Lorena, Leonardo Rodrigues Silva e Renan Coelho da Silva Menezes!

RESUMO

Este trabalho aborda o uso da Metodologia *Scrum* no desenvolvimento de banco de dados. Visa demonstrar a eficácia dessa quando comparada à metodologia tradicional. Nele abordou-se a necessidade da agilidade para sobrepor a burocratização, comprovada por meio de uma simulação, onde se demonstrou parte por parte a utilização da Metodologia *Scrum* para o desenvolvimento de um banco de dados e como se consegue obter um alto controle do projeto, por meio da transparência, inspeção e adaptação.

Palavras-chave: banco de dados; cascata; metodologia ágil; metodologia tradicional; scrum.

ABSTRACT

This work addresses the use of the Scrum Methodology in database development. It aims to demonstrate its effectiveness when compared to the traditional methodology. It addressed the need for agility to overcome bureaucratization, proven through a simulation, which demonstrated part by part the use of the Scrum Methodology for the development of a database and how high control of the project can be achieved, through transparency, inspection and adaptation.

Keywords: database; waterfall; agile methodology; traditional methodology; scrum.

SUMÁRIO

1 INTRODUÇÃO	9
1.1 Tema	9
1.2 Justificativa	10
1.3 Problema	10
1.4 Hipótese	11
1.5 Objetivo	12
1.6 Metodologia	12
1.7 Estrutura da monografia	12
2 CONTEXTUALIZAÇÃO	13
2.1 Transformação digital	13
2.2 O problema da burocratização	14
2.3 A necessidade da agilidade.....	15
3 METODOLOGIAS DE PROJETO	17
3.1 Metodologia de projeto preditiva	17
3.2 Metodologia de projeto ágil	17
3.3 Metodologia de projeto híbrida	20
4 METODOLOGIA TRADICIONAL DE DESENVOLVIMENTO DE SOFTWARE	21
5 METODOLOGIA ÁGIL SCRUM DE DESENVOLVIMENTO DE SOFTWARE	24
6 BANCO DE DADOS	32
6.1 Necessidade para desenvolver um banco de dados	32
6.2 O que é um banco de dados?	32
6.3 O que são dados?	35
6.4 Banco de dados na empresa.....	35
6.5 Criação do modelo de banco de dados	36
6.6 Tipos de banco de dados	37
6.7 Vantagens do banco de dados.....	38

7 SIMULAÇÃO.....	40
8 CONCLUSÃO	52
REFERÊNCIAS.....	54

1 INTRODUÇÃO

Este capítulo é composto pelas seções: 1.1 Tema a 1.7 Estrutura da monografia.

1.1 Tema

Este trabalho tem como objetivo discorrer sobre o desenvolvimento ágil de bancos de dados. Para isso, é necessário abordar conceitos fundamentais, como as metodologias ágeis e *Scrum*.

Nos últimos anos, a abordagem ágil estabeleceu-se como uma das perspectivas mais empolgantes e eficazes no campo do desenvolvimento de software, inclusive em bancos de dados. O desenvolvimento ágil permite uma visão flexível e interativa focada na entrega contínua de valor ao cliente.

No desenvolvimento ágil de bancos de dados, as equipes concentram-se na colaboração, comunicação eficaz e entregas incrementais. Em vez de seguir um plano rígido do início ao fim, as equipes adaptam-se às mudanças de requisitos e priorizam as tarefas com maior valor. Isso se traduz em um processo mais eficiente e na capacidade de entregar funcionalidades de maneira mais rápida.

O trabalho propõe uma comparação entre duas abordagens de gestão de projetos: o *Scrum*, uma metodologia ágil conhecida por sua flexibilidade e adaptabilidade e a metodologia tradicional, frequentemente associada a abordagens mais sequenciais e burocráticas.

O desenvolvimento ágil de bancos de dados oferece muitas vantagens quanto a flexibilidade e eficiência, mas também apresenta desafios que precisam ser abordados com atenção. A colaboração, a comunicação eficaz e a adoção de melhores práticas são vitais para o sucesso dessa abordagem no mundo do desenvolvimento de bancos de dados.

1.2 Justificativa

Compreender as razões para estudar o desenvolvimento ágil de banco de dados é importante, uma vez que se trata de uma abordagem que está transformando a forma como as organizações lidam com o gerenciamento de dados.

Aqui estão algumas justificativas para aprofundar o estudo do desenvolvimento ágil de banco de dados: o potencial para otimizar a entrega de sistemas de banco de dados; o impacto em diversos setores e indústrias; a abertura de novas possibilidades de agilidade e eficiência; o efeito na qualidade dos dados e as oportunidades profissionais que surgem com sua adoção.

Investigar o desenvolvimento ágil de banco de dados permite compreender as mudanças em curso na gestão de dados, explorar novas formas de agilidade e eficiência na entrega de sistemas de banco de dados, avaliar a melhoria da qualidade dos dados e aproveitar as crescentes oportunidades profissionais nesse campo. Trata-se de um enfoque dinâmico e promissor que vale a pena explorar em detalhes.

1.3 Problema

O desenvolvimento ágil de banco de dados representa um ponto de vista inovador e altamente relevante no mundo da gestão de dados. Distingue-se por sua ênfase na flexibilidade, colaboração e entrega contínua de funcionalidades, o que o torna um contraponto interessante à metodologia tradicional de desenvolvimento de bancos de dados.

Uma das principais questões é a avaliação da eficácia e eficiência de ambas as metodologias. Enquanto a metodologia tradicional tende a seguir um plano detalhado e rígido, a abordagem ágil busca a entrega incremental e flexível.

A comparação entre esses métodos deve considerar como cada uma lida com mudanças de requisitos, adaptação a imprevistos e a capacidade de atender às necessidades do cliente em constante evolução.

Outro ponto crítico a ser comparado é o controle do projeto. A metodologia tradicional oferece um alto nível de controle desde o início, com planejamento detalhado e requisitos bem definidos, enquanto, a metodologia ágil enfatiza o controle por meio da transparência, inspeção e adaptação ao longo do processo. A análise

deve considerar como esses tratamentos lidam com a gestão de riscos, qualidade do produto e prazos.

A comparação entre as metodologias, também, deve abordar a colaboração e a comunicação. A metodologia ágil promove uma comunicação mais próxima e interativa entre a equipe de desenvolvimento e o cliente, enquanto a metodologia tradicional, muitas vezes, segue um modelo de comunicação mais formal.

A eficácia da colaboração e da comunicação em ambas as abordagens deve ser avaliada para determinar qual delas se adapta melhor às necessidades da equipe e do projeto.

Além disso, a flexibilidade e a capacidade de adaptação são pontos fundamentais a serem comparados. A metodologia ágil é conhecida por sua capacidade de se ajustar rapidamente às mudanças, enquanto a metodologia tradicional pode ser mais rígida nesse aspecto.

A comparação entre as metodologias ágeis e tradicionais no desenvolvimento de banco de dados é necessária para determinar qual delas melhor atende às necessidades e objetivos específicos de um projeto ou organização.

1.4 Hipótese

A implementação de metodologias ágeis no desenvolvimento de bancos de dados pode resultar em processos mais eficientes, maior flexibilidade para adaptação a mudanças nos requisitos e na entrega de sistemas de banco de dados de maior qualidade, quando comparada à abordagem tradicional.

Por meio da análise de simulação é possível demonstrar que o desenvolvimento ágil de banco de dados pode ser uma estratégia altamente benéfica às empresas que buscam melhorar sua capacidade de resposta às necessidades do mercado e de seus clientes.

1.5 Objetivo

Identificar as etapas de aplicação da metodologia *Scrum* para o desenvolvimento de um banco de dados e demonstrar uma comparação entre os benefícios da metodologia ágil sobre a tradicional.

1.6 Metodologia

O tipo de pesquisa de acordo com os objetivos é a descritiva, com abordagem qualitativa e técnicas de investigação fortemente voltadas ao levantamento bibliográfico e a simulação.

Usou-se o elemento textual da monografia para o desenvolvimento do trabalho, buscando fontes de artigos científicos, livros, convenções e palestras de especialistas, professores e escritores, que já estudam e utilizaram a metodologia ágil.

1.7 Estrutura da monografia

Esta monografia é composta por oito capítulos:

- Capítulo 1 Introdução;
- Capítulo 2 Contextualização;
- Capítulo 3 Metodologias de projeto;
- Capítulo 4 Metodologia tradicional de desenvolvimento de software;
- Capítulo 5 Metodologia ágil *Scrum* de desenvolvimento de software;
- Capítulo 6 Banco de dados;
- Capítulo 6 Simulação;
- Capítulo 7 Conclusão;

2 CONTEXTUALIZAÇÃO

Este capítulo é composto pelas seções: 2.1 A transformação digital a 2.3 A necessidade da agilidade.

2.1 Transformação digital

Segundo Llorente (2016), ao se falar de transformação digital é necessário discutir a transformação dos negócios. Os produtos e serviços experimentaram uma revolução e irão surgir serviços de valor agregado que utilizarão as novas tecnologias e criarão impacto direto sobre a demonstração de resultados dos negócios e a experiência com o cliente.

Maximizar canais e processos, encontrar novos modelos e fluxos de receita tendo base as exigências dos clientes, e, naturalmente, mudar a cultura interna da empresa para adicionar o "digital" no centro de tudo. (Id., 2016)

De acordo com Llorente (2016), ao se aderir ao desafio da transformação digital, as organizações devem assumir que a modificação vem de dentro e, necessariamente, apostar em novas metodologias: na liderança; nas áreas de operações; nos modelos de trabalho; nas ferramentas de coleta e análise da informação; e até a condução da experiência do consumidor.

Somente as organizações que assumirem essas várias vertentes da mudança capitalizarão todas as novas chances que têm na era da economia digital. (Id., 2016)

Em cada um desses setores, uma nova startup, movida a energia digital, criou grande valor para os clientes, enquanto enfraquecia e solapava a posição das empresas tradicionais. Embora o desafiante digital esteja comendo o lucro delas, as titulares convencionais se veem incapazes de reagir, competindo diretamente com a mesma oferta. (ROGERS, 2017, p. 152).

Como comentado por Salim Ismail, em seu livro *Exponential Organizations*, no passado projetava-se o crescimento de uma organização de forma linear. Hoje, deve-se entender que as empresas que conseguem crescer de maneira exponencial estão sempre à frente na curva de crescimento comercial. (LLORENTE, 2016)

O empreendedor é submetido a pensar sempre em termos da curva de crescimento comercial, como ele poderia trocar os cenários que enfrenta, que inovações podem incomodar o seu campo de ação, o que a concorrência está

realizando e como usar tudo isso para inovar. A pressão para educar-se, inovar e se transformar de acordo com o que o mundo muda é muito alta. (LLORENTE, 2016)

Para construir um negócio, é necessário reunir diferentes elementos, e a compreensão da tecnologia é apenas um deles. Por isso, as empresas de sucesso são aquelas que conseguem integrar a tecnologia ou potencializá-la sem esquecer outras partes do negócio. Um empreendedor pode ser magnífico, tecnologicamente falando, um ás no desenvolvimento de produtos, mas pode não ter êxito sem um bom conjunto de vendas, marketing, desenvolvimento, administração e atendimento ao cliente. Não é o suficiente ser bom tecnologicamente; para subir, é necessária uma equipe diversificada, no caso de uma empresa digital ou tradicional. (LLORENTE, 2016, p. 57).

Conforme a demanda do mercado mundial e a complexidade das inovações tecnológicas foram crescendo, o cenário oferece maior dificuldade em aplicar métodos tradicionais e arcaicos para resolver problemas novos e complexos em praticamente todos os projetos.

2.2 O problema da burocratização

O presidente do conselho que reúne as instituições de ensino e pesquisa, Fernando Peregrino, afirmou que a burocracia, além de atrasar o trabalho dos pesquisadores, gera o desperdício bilionário de dinheiro público. (PEREGRINO [S. d.] *apud* WESTPHALEN, 2019)

"Num cálculo considerando a perda de tempo do cientista com a burocracia, nós chegamos a um número: 35% do tempo dele é consumido com essas coisas, com a burocracia. Aplicando aos gastos com ciência e tecnologia no Brasil, nós chegamos ao número de R\$ 9 bilhões, em 2016. Só no sistema federal são R\$ 3 bilhões". (PEREGRINO [S. d.] *apud* WESTPHALEN, 2019)

O processo de burocratização ocorre nas sociedades ocidentais por sua complexificação. É uma resposta racional ao aumento da complexidade da vida social. Não é somente uma característica do setor público e estatal como o senso comum tende a entender. Decorre da necessidade de organização de processos de produção complexos. Desse modo, as instituições produtivas também tendem a se burocratizar com a construção de mecanismos de controle e de gestão, procurando, assim, a garantia de eficiência. (VERONESE, 2006, p. 15).

Pode-se verificar que ainda há muita burocracia no âmbito tecnológico, causando perda de tempo e recursos para o desenvolvimento necessário.

De acordo com Semedo (2012), muitas empresas já tiveram problemas relacionados a grandes investimentos em que não conseguiram atingir os objetivos a que se propuseram, visto que, diversas vezes não efetuam a escolha adequada dos métodos, do âmbito e dos critérios que precisam para desenvolverem esses projetos, originando perda de qualidade e diminuição da sua produtividade.

Como reflexo destas escolhas, não conseguem alcançar os objetivos de negócio previamente definidos e propostos. Neste âmbito, é fundamental perceber a importância da utilização de uma metodologia, a qual se pode definir como: um conjunto de técnicas, procedimentos, ferramentas e documentação que sustentam e auxiliam os gestores/responsáveis no desenvolvimento e implementação de um sistema de informação. Dentro de uma metodologia, existe ainda várias fases que orientam os gestores/responsáveis do projeto para a correta definição e escolha de técnicas apropriadas de acordo com as etapas que ocorrem durante o desenvolvimento e implementação de um sistema de informação. (SEMEDO, 2012, p. 4).

2.3 A necessidade da agilidade

Com o passar do tempo, vieram novas metodologias e visões que buscam exigir menor burocratização. Há projetos que demandam anos de desenvolvimento, muitas vezes devido ao excesso de burocracia como protocolos, memorando e procedimentos formais que acabam atrasando ainda mais a sua entrega, já que os projetos podem ter problemas que, muitas vezes, possuem muita complexidade e contratempos.

De acordo com Silva; Souza; Camargo (2013), no início da década de 1990, existiam muitos projetos que tinham retrabalhos e, por consequência, acabavam estourando o prazo de entrega definido no início do planejamento do projeto. Os retrabalhos eram consequências de projetos que se utilizavam na maior parte do tempo para documentar ao invés de desenvolver e testar.

Surge, então, a metodologia ágil voltada aos projetos e que seus eventuais problemas sejam tratados de perto e com maior flexibilidade, cooperação, eficiência, redução de desperdício, foco no cliente, eliminação da burocracia e a melhoria contínua surgiu. (Id., 2013)

Segundo Cintra *et al.* (2018), uma vez que os métodos tradicionais não estavam demonstrando mais eficácia, as empresas buscaram soluções alternativas para diminuir os desperdícios e custos que os outros métodos resultavam, sendo assim surgiu a metodologia ágil.

Por meio dessa nova metodologia é possível desenvolver projetos com menor burocracia, uma vez que não é obrigatória a execução de grandes análises e pesquisas iniciais, visto que é possível fazer modificações ao longo do projeto conforme a necessidade encontrada. (Id., 2018)

Segundo Soares (2004), as metodologias tradicionais reconhecidas por serem pesadas ou orientadas a planejamentos, têm que ser aplicadas apenas em locais em que os requisitos do sistema são estáveis e onde os requisitos futuros são previsíveis.

Porém, em projetos em que é previsto diversas mudanças e os requisitos são passíveis de alterações, onde mudar partes do código não é uma coisa que apresenta muito custo, as equipes são pequenas, as datas de entrega são curtas e o desenvolvimento rápido é fundamental, não pode haver requisitos parados, necessitam-se, então, de metodologias ágeis. Além disso o ambiente das organizações é de constante mudança, não permitindo então que os requisitos sejam estáticos. (Id., 2004)

3 METODOLOGIAS DE PROJETO

"A metodologia de projeto é um conjunto de estratégias e práticas que facilitam o planejamento e a execução de um projeto. De acordo com o dicionário, a palavra metodologia refere-se à relação entre fatores teóricos e práticos. Ou seja, como os métodos, enquanto teoria, podem ser aplicados em determinado contexto". (ESPINHA, 2021)

Segundo Espinha (2021), "Existem inúmeras metodologias de projeto, pois cada uma terá um foco de melhoria. Mas, por consenso, costumamos agrupá-las em três categorias gerais: Preditiva, Ágil e Híbrida".

3.1 Metodologia de projeto preditiva

As metodologias preditivas são as que priorizam o controle e o planejamento do projeto. Isto é, as etapas são mapeadas e estruturadas antes de iniciar a execução - Figura 1. (ESPINHA, 2021)

Figura 1 - O essencial da metodologia preditiva.



Fonte: Veronese (2021).

3.2 Metodologia de projeto ágil

As metodologias ágeis, também chamadas de adaptativas, são aquelas que priorizam uma visão mais ampla do projeto, tendo foco no valor da entrega do que no planejamento em si - Figura 2. (ESPINHA, 2021)

Figura 2 - O essencial da metodologia ágil.



Fonte: Janson (2022).

"A metodologia ágil é considerada um divisor de águas na indústria de software, em razão da sua abordagem revolucionária". (FIA, 2022)

"Em 2001, quando ela foi oficialmente lançada, os desenvolvedores e, principalmente, os clientes, penavam para chegar a soluções satisfatórias. Muito por causa de uma mentalidade defasada, na qual os processos esbarravam na burocracia e no excesso de requisitos formais". (Id., 2022)

"Ela surgiu a partir do manifesto ágil, que ocorreu em na primavera de 2000, líderes da comunidade de extreme *programming* se reuniram para discutir as práticas do XP. Durante essa reunião também foi debatido a relação entre o XP e os até então chamados Método Leves". (BRASILEIRO, 2020)

"Esses tais de Métodos Leves eram os que hoje conhecemos como *SCRUM*, *DSDM* [*Dynamic Software Development Method*], *Adaptive Software Development* [*ASD*], *Crystal*, *Feature-Driven Development*, *Pragmatic Programming* [*FDMD*] entre outros". (Id., 2020, grifo nosso)

"Essa relação foi discutida pelo fato dos métodos leves e do XP estarem na contramão dos até então métodos pesados. Ou seja, eles possuíam uma abordagem mais leve e menos burocrática". (Id., 2020)

"Como consequência dessa discussão foi concluído que o XP era melhor como um método específico, porém com espaço comum entre ele e os métodos leves. Foi dessa conclusão que Robert Cecil Martin, conhecido como Tio Bob, resolveu criar um encontro para as pessoas interessadas em Métodos Leves". (Id., 2020)

Segundo Brasileiro (2020), "Muitas pessoas foram contactadas, porém apenas 17 estavam presente em fevereiro de 2001 em um resort de *ski* nas montanhas nevadas de Utah".

"Durante essa reunião um grande consenso sobre como deveriam ser os métodos de desenvolvimento de software foi criado. E foi batizado como O Manifesto do desenvolvimento de Software Ágil, no qual chamamos de Manifesto Ágil". (Id., 2020)

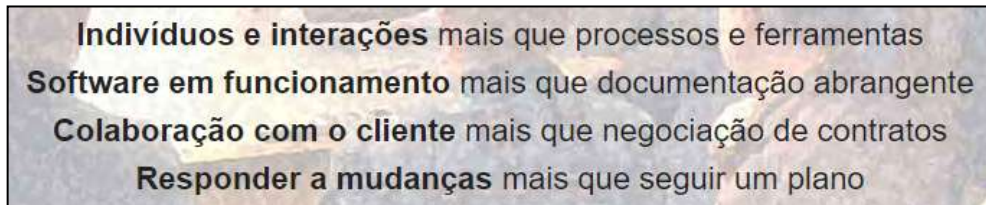
Com a conclusão dessa reunião, chegou-se aos doze princípios do Manifesto Ágil, que são: (AGILEMANIFESTO, 2001)

- Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor;
- Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas;
- Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos;
- Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto;
- Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho;
- O método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara;
- Software funcional é a medida primária de progresso;
- Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes;
- Contínua atenção à excelência técnica e bom design, aumenta a agilidade;
- Simplicidade - a arte de maximizar a quantidade de trabalho que não precisou ser feito - é essencial;
- As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis;

- Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

A Figura 3 apresenta os valores do Manifesto Ágil. (AGILEMANIFESTO, 2001)

Figura 3 - Valores do Manifesto Ágil.



Fonte: Agilemanifesto (2001).

Em suma, metodologia ágil é:

Uma forma de gerir projetos, em que se busca a otimização dos processos. Em vez de seguir um plano rígido - como na gestão tradicional -, na metodologia ágil, podemos fazer ajustes e melhorias ao longo do desenvolvimento de projetos. Isso significa identificar e corrigir falhas rapidamente, garantindo resultados mais efetivos. Portanto, essa forma de gestão permite às pessoas gerentes uma abordagem moderna e eficiente para enfrentar desafios do mercado. (LOSNAK, 2023, p. 1).

3.3 Metodologia de projeto híbrida

A metodologia híbrida combina elementos das metodologias ágeis e preditivas no desenvolvimento de projetos - Figura 4. (ESPINHA, 2021)

Figura 4 - O essencial da metodologia híbrida.



Fonte: Espinha (2021).

4 METODOLOGIA TRADICIONAL DE DESENVOLVIMENTO DE SOFTWARE

O conteúdo, a seguir, apresenta a transcrição da seção "Metodologia Tradicional" contida no artigo intitulado "Metodologias ágeis para o desenvolvimento de softwares", escrito por Thiago Bessa e Daniel Dias Branco Arthaud e publicado, em 14 mar. 2019, na Revista Ciência e Sustentabilidade.

O termo tradicional, nesse contexto, é utilizado para identificar metodologias ou processos que são baseados na metodologia clássica (Modelo em Cascata). O que melhor caracteriza estas metodologias é a separação bem rígida das fases de projeto, que consistem em: Levantamento de Requisitos, Análise, Desenho, Implementação, Testes e Implantação. Cada fase tem suas especificidades e possuem entre si interdependência, isto é, a próxima fase só começa quando a anterior estiver pronta. Elas seguem também algumas premissas, que são, linearidade, determinismo, especialização, foco na execução e crescimento exponencial do custo de alteração.

Nas metodologias tradicionais, o desenvolvimento se divide em duas grandes partes, a concepção e a construção. É na fase de concepção que se tem a maior interação com o cliente e é gasto um grande tempo identificando as necessidades para evitar que apareçam novas necessidades ou durante a fase de construção. Essa fase de concepção é considerada a mais importante do projeto, pois dependendo dela, este terá sucesso ou não.

Se surgirem imprevistos durante a construção acarretará retrabalho e aumento no tempo de projeto e isso causará impactos no preço, prazo e qualidade, o que pode gerar conflitos com o cliente, que além de sofrer com os impactos possui uma grande possibilidade de o sistema não atender suas necessidades.

Durante a construção não há interação com o cliente, impedindo que sejam feitas correções ou modificações no escopo do projeto. As metodologias tradicionais, devem ser utilizadas apenas em situações em que os requisitos são estáveis e previsíveis.

[...]. (Pontes, 2019, p. 178).

De acordo com Borchardt; Campos; Kubota (2022, grifo nosso),

O método cascata (em inglês *Waterfall*) é uma forma de gerenciamento que utiliza fases sequenciais e, de acordo com Keeling e Branco (2018), pode ser considerado um método preditivo, pois o projeto é planejado integralmente desde o seu início e as necessidades de recursos são conhecidas desde o princípio; porém, esse tipo de gerenciamento não acomoda bem possíveis mudanças e deve seguir uma sequência rígida de processos - [Figura 5]. (Kubota, 2022, p. 3).

De acordo com Nascimento (2019), "O método cascata demanda extensiva documentação, incorre em altos custos gerenciais e necessita de bastante dedicação dos envolvidos no projeto para a infraestrutura de armazenamento e gestão da informação".

Figura 5 - Gestão da informação.

Forças	Fraquezas
O projeto é planejado integralmente desde o seu início.	Limitações em acomodar possíveis mudanças.
As necessidades dos recursos são conhecidas desde o começo do projeto.	As mudanças podem gerar custos consideravelmente elevados, tornando o projeto extensamente caro.
Não são necessárias equipes com habilidades diferenciadas.	Leva muito tempo até qualquer entrega ser realizada.
Membros da equipe não precisam estar localizados.	Requer planos completos e detalhados.
	Tem uma sequência rígida de processos.
	O valor para o cliente não é o foco principal.

Fonte: Borchardt; Campos; Kubota (2022).

"[...]. O método é assim chamado porque prevê uma sequência em cascata de uma atividade para outra, abordando os problemas anteriormente encontrados em desenvolvimento de software, adicionando etapas que cobrem verificação, análise de risco e desenvolvimento". (BALLE *et al.*, 2018 *apud* NASCIMENTO, 2019)

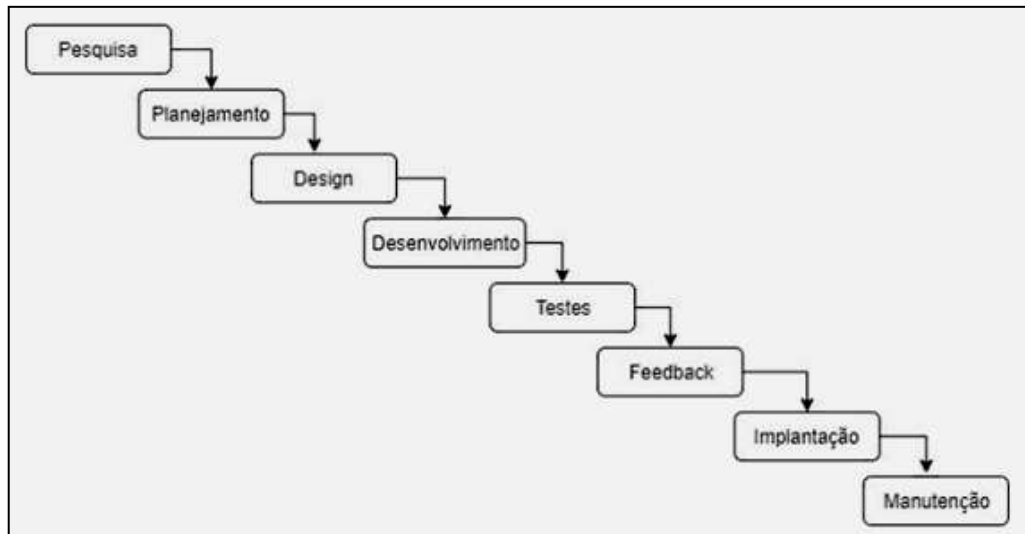
A Figura 6 mostra que "A metodologia enfatiza uma progressão linear estruturada entre fases definidas. Cada fase consiste em um conjunto definido de atividades e entregas que devem ser realizadas antes que a fase seguinte possa começar". (AWAD, 2005 *apud* Id., 2019)

Segundo AWAD (2005) *apud* Nascimento (2019), "As fases são sempre chamadas de forma diferente, mas a ideia básica é que a primeira fase tenta capturar o que o sistema fará, seus requisitos de sistema e software; a segunda fase determina como isso será projetado".

"A terceira fase é onde os desenvolvedores começam a escrever o código, a quarta fase é o teste do sistema e a fase final é focada em tarefas de implementação, como treinamento e documentação". (Id., 2005 *apud* Id., 2019)

"O *feedback* do proprietário do projeto é recebido apenas após o projeto for completamente desenvolvido e testado. O método em cascata é adequado para projetos de desenvolvimento de softwares de baixa complexidade, onde os requisitos são claros e um planejamento detalhado pode ser facilmente elaborado para o todo projeto". (DESPA, 2014 *apud* Id., 2019)

Figura 6 - Metodologia em cascata.



Fonte: Adaptado Despa (2014) *apud* Nascimento (2019).

5 METODOLOGIA ÁGIL DE DESENVOLVIMENTO DE SOFTWARE SCRUM

Este capítulo foi elaborado utilizando o documento "Guia do *Scrum* - O guia definitivo para o *Scrum*: as regras do jogo" escrito por Ken Schwaber e Jeff Sutherland e publicado em nov. 2020.

Com intuito de tornar a leitura do texto desse capítulo mais fluída, "os autores e anos" não foram declarados, tanto nas citações diretas quanto indiretas em cada parágrafo desse capítulo. Caso o texto seja de autoria diferente de Schwaber; Sutherland (2020), o(s) autor(es) será(ão) obrigatoriamente citado(s).

Schwaber; Sutherland (2020) declaram que o "*Scrum* é um *framework* que não é pesado e que ajuda pessoas, times e organizações a criar valor por meio de soluções adaptáveis para problemas complexos. *Scrum* é simples, o *framework Scrum* é conscientemente incompleto, apenas com a definição das partes necessárias para implementar a teoria *Scrum*".

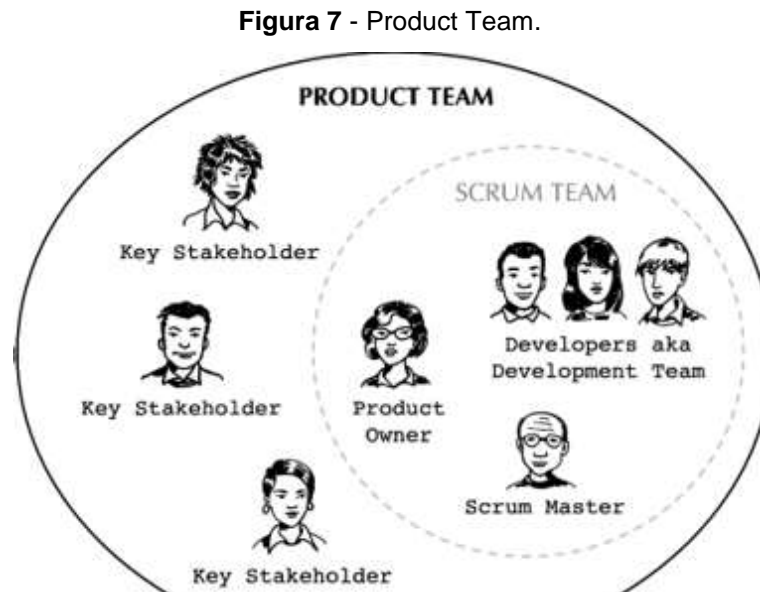
O *Scrum* é construído sobre o cérebro coletivo das pessoas que o utilizam. Em vez de dar às pessoas instruções detalhadas, as ordens do Guia do *Scrum* definem seus relacionamentos e interações. *Scrum* junta quatro eventos formais para inspeção e adaptação, que estão dentro de um evento, a *Sprint*. Esses eventos funcionam porque implementam os pilares empíricos do *Scrum*.

Os pilares empíricos do *Scrum* são:

- **Transparência:** o processo resultante e o trabalho devem poder ser vistos tanto para quem executa o trabalho quanto para quem recebe o trabalho;
- **Inspeção:** os aparatos do *Scrum* e o progresso rumo às metas definidas devem ser verificados com frequência e diligência para detectar mudanças ou problemas potencialmente indesejáveis;
- **Adaptação:** se qualquer aspecto de um processo se desviar para fora dos limites admissíveis ou se o produto resultante for inaceitável, o processo que está sendo feito ou os materiais que estão sendo criados devem ser ajustados. O ajuste tem que ser feito o mais rápido possível para diminuir novos desvios.

A unidade fundamental do Scrum é um breve time de pessoas, um *Scrum Team*. O *Scrum Team* é formado por um *Scrum Master*, um *Product owner* e *Developers*. Dentro de um *Scrum Team*, não há pequenos times ou hierarquias. É uma unidade coesa de profissionais empenhados em um objetivo de cada vez, a Meta do Produto.

A Figura 7 apresenta a composição do *Product Team*.



Fonte: Pichler (2022).

O "*Scrum Team* é pequeno o suficiente para permanecer ágil e grande o suficiente para concluir um trabalho significativo dentro de uma *Sprint*, normalmente 10 ou menos pessoas".

Developers são as pessoas do *Scrum Team* que estão compromissados em fazer a criação de qualquer aspecto de um Incremento utilizável a cada *Sprint*. As habilidades específicas necessárias pelos *Developers* geralmente são múltiplas e variam de acordo com o domínio de trabalho. [...].

Os *Developers* são sempre responsáveis por:

- Criar um plano para a *Sprint* e o *Sprint Backlog*;
- Introduzir gradualmente qualidade aderindo a uma *Definition of Done*;
- Adaptar seu plano a cada dia em direção à meta da *Sprint*.

"O *Product Owner* é responsável por aumentar o máximo o valor do produto resultante do trabalho do *Scrum Team*".

O *Product Owner* também é responsável pelo gerenciamento do *Product Backlog*, que inclui:

- Desenvolver e comunicar claramente a meta do produto;
- Criar e comunicar explicitamente os itens do *Product Backlog*;
- Ordenar os itens do *Product Backlog*;
- Dar garantia que o *Product Backlog* seja transparente, visível e compreensível.

"O *Product Owner* é uma pessoa que não tem um comitê e representa as necessidades de muitos *stakeholders* no *Product Backlog*".

"O *Product Owner* garante que os participantes estejam preparados para discutir os itens mais importantes do *Product Backlog* e como eles são mapeados para a Meta do Produto".

O *Scrum Master* serve ao *Scrum Team* de várias maneiras, incluindo:

- Treinar os membros do time em autogerenciamento e *cross-funcionalidade*;
- Ajudar o *Scrum Team* a se concentrar na criação de incrementos de alto valor que atendem à *Definition of Done*;
- Provocando a remoção de impedimentos ao progresso do *Scrum Team*;
- Garantir que todos os eventos *Scrum* ocorram e sejam positivos, produtivos e mantidos dentro do *Timebox*.

"*Scrum Master* é responsável por estabelecer o *Scrum* conforme definido no Guia do *Scrum*. Eles fazem isso ajudando todos a entender a teoria e a prática do *Scrum*, tanto no *Scrum Team* quanto na organização".

"O *Scrum Master* é responsável pela eficácia do *Scrum Team*. Eles fazem isso permitindo que o *Scrum Team* melhore suas práticas dentro do *framework Scrum*".

A "*Sprint* é um contêiner para todos os outros eventos".

"Cada evento no *Scrum* é uma oportunidade formal para inspecionar e adaptar os artefatos do *Scrum*".

"Esses eventos são projetados especificamente para permitir a transparência necessária. A falha em operar quaisquer eventos conforme prescrito resulta em oportunidades perdidas de inspeção e adaptação".

"Os eventos são usados no Scrum para criar regularidade e minimizar a necessidade de reuniões não definidas no Scrum. O ideal é que todos os eventos sejam realizados no mesmo horário e local para reduzir a complexidade".

"*Sprint* são eventos de duração fixa de um mês ou menos para criar consistência. Uma nova *Sprint* começa imediatamente após a conclusão da *Sprint* anterior".

"Todo o trabalho necessário para atingir a meta do Produto, incluindo *Sprint Planning*, *Daily Scrums*, *Sprint Review* e *Sprint Retrospective*, acontece dentro de *Sprints*".

Durante a *Sprint*:

- Nenhuma mudança é feita que coloque em risco a meta da *Sprint*;
- A qualidade não diminui;
- O *Product Backlog* é refinado conforme necessário;
- O escopo pode ser esclarecido e renegociado com o *Product Owner* conforme mais é aprendido.

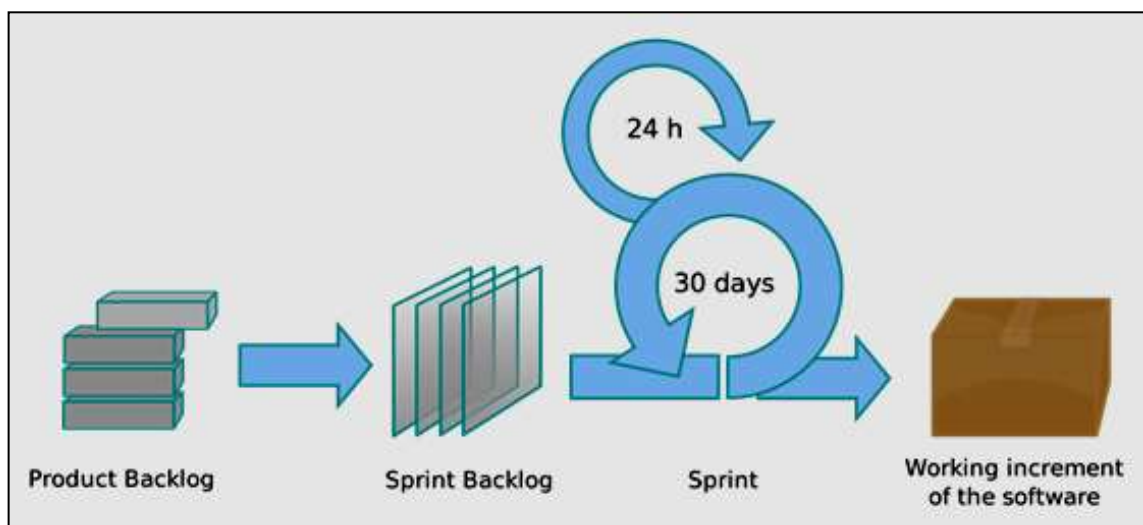
"A *Sprint Planning* inicia a *Sprint* ao definir o trabalho a ser realizado na *Sprint*. Este plano resultante é criado pelo trabalho colaborativo de todo o *Scrum Team*".

A *Sprint Planning* aborda os seguintes tópicos:

- **Por que esta *Sprint* é valiosa?** - é definida uma meta para a *Sprint* visando o aumento de valor e utilidade do produto, essa meta é definida com todo o *Scrum Team*. A meta da *Sprint* deve ser finalizada antes do final da *Sprint Planning*;
- **O que pode ser feito nesta *Sprint*?** - por meio de discussão com o *Product Owner*, os *Developers* selecionam itens do *Product Backlog* para incluir na *Sprint* atual. Selecionar o quanto pode ser concluído em uma *Sprint* pode ser um desafio. No entanto, quanto mais os *Developers* sabem sobre seu desempenho anterior, sua capacidade futura e sua *Definition of Done*, mais confiantes eles estarão em suas previsões quanto a *Sprint*. Um bom método para auxiliar nessa etapa é utilizar o quadro *Kanban*.
- **Como o trabalho escolhido será realizado?** - para cada item do *Product Backlog* selecionado, os *Developers* planejam o trabalho necessário para

criar um incremento que atenda à *Definition of Done*. Isso geralmente é feito decompondo itens do *Product Backlog* em itens de trabalho menores de um dia ou menos. A forma como isso é feito fica a critério exclusivo dos *Developers*. A meta da *Sprint*, os itens do *Product Backlog* selecionados para a *Sprint*, mais o plano para entregá-los são chamados juntos de *Sprint Backlog*. A *Sprint Planning* tem um *Timebox* definido com duração máxima de oito horas para uma *Sprint* de um mês. Para *Sprints* mais curtas, o evento geralmente é mais curto - Figura 8.

Figura 8 - Meta da Sprint.



Fonte: Alura (2016).

"O propósito da *Daily Scrum* é inspecionar o progresso em direção a Meta da *Sprint* e adaptar o *Sprint Backlog* conforme necessário, ajustando o próximo trabalho planejado".

"A *Daily Scrum* é um evento de 15 minutos para os *Developers* do *Scrum Team*".

"Para reduzir a complexidade, é realizado no mesmo horário e local, todos os dias úteis da *Sprint*. Se o *Product Owner* ou o *Scrum Master* estão trabalhando ativamente nos itens do *Sprint Backlog*, eles participam como *Developers*".

"Os *Developers* podem selecionar qualquer estrutura e técnicas que quiserem, desde que seu *Daily Scrum* se concentre no progresso em direção a Meta da *Sprint* e produza um plano de ação para o próximo dia de trabalho. Isso cria foco e melhora o autogerenciamento".

As *Daily Scrums* melhoram as comunicações, identificam os impedimentos, promovem a rápida tomada de decisões e conseqüentemente, eliminam a necessidade de outras reuniões.

"O propósito da *Sprint Review* é inspecionar o resultado da *Sprint* e determinar as adaptações futuras. O *Scrum Team* apresenta os resultados de seu trabalho para os principais *stakeholders* e o progresso em direção a Meta do Produto é discutido".

"Durante o evento, o *Scrum Team* e os *stakeholders* revisam o que foi realizado na *Sprint* e o que mudou em seu ambiente. Com base nessas informações, os participantes colaboram sobre o que fazer a seguir. O *Product Backlog* também pode ser ajustado para atender a novas oportunidades".

"A *Sprint Review* é o penúltimo evento da *Sprint* e tem um *Timebox* com prazo máximo de quatro horas para uma *Sprint* de um mês. Para *Sprints* mais curtas, o evento geralmente é mais curto".

"O *Scrum Team* inspeciona como foi a última *Sprint* em relação a indivíduos, interações, processos, ferramentas e sua *Definition of Done*".

"Os elementos inspecionados geralmente variam com o domínio de trabalho. As suposições que os desviaram são identificadas e suas origens exploradas. O *Scrum Team* discute o que deu certo durante a *Sprint*, quais problemas encontraram e como esses problemas foram (ou não) resolvidos".

"O *Scrum Team* identifica as mudanças mais úteis para melhorar sua eficácia. As melhorias mais impactantes são endereçadas o mais rápido possível".

"A *Sprint Retrospective* conclui a *Sprint*. É limitada pelo *Timebox* de no máximo três horas para uma *Sprint* de um mês. Para *Sprints* mais curtas, o evento geralmente é mais curto".

"Os artefatos do *Scrum* representam trabalho ou valor. Eles são projetados para maximizar a transparência das principais informações. Assim, todos os que os inspeciona têm a mesma base para adaptação".

"Cada artefato contém um compromisso para garantir que ele forneça informações que aumentem a transparência e o foco contra o qual o progresso pode ser medido":

- Para o *Product Backlog* é a Meta do Produto;
- Para o *Sprint Backlog* é a Meta da *Sprint*;
- Para o *increment* é a *Definition of Done*.

"O *Product Backlog* é uma lista ordenada e emergente do que é necessário para melhorar o produto. É a única fonte de trabalho realizado pelo *Scrum Team*".

"Os itens do *Product Backlog* que podem ser realizados pelo *Scrum Team* em uma *Sprint* são considerados preparados para seleção no evento *Sprint Planning*".

Eles geralmente adquirem esse grau de transparência após as atividades de refinamento. O *Product Backlog refinement* é o ato de quebrar e incluir definição adicional aos itens do *Product Backlog* para ter itens menores e mais precisos. Esta é uma atividade contínua para adicionar detalhes, como descrição, ordem e tamanho.

"Os *Developers* que farão o trabalho são responsáveis pelo dimensionamento. O *Product Owner* pode influenciar os *Developers*, ajudando-os a entender e selecionar *trade-offs* (trocas de itens)".

"O *Sprint Backlog* é composto pela Meta da *Sprint* (porque), o conjunto de itens do *Product Backlog* selecionados para a *Sprint* (o que), bem como um plano de ação para entregar o Incremento (como)".

"O *Sprint Backlog* é um plano feito por e para os *Developers*. É uma imagem altamente visível, em tempo real do trabalho que os *Developers* planejam realizar durante a *Sprint* para atingir a Meta da *Sprint*".

"Consequentemente, o *Sprint Backlog* é atualizado ao longo da *Sprint* conforme mais é aprendido. Deve ter detalhes suficientes para que eles possam inspecionar seu progresso na *Daily Scrum*".

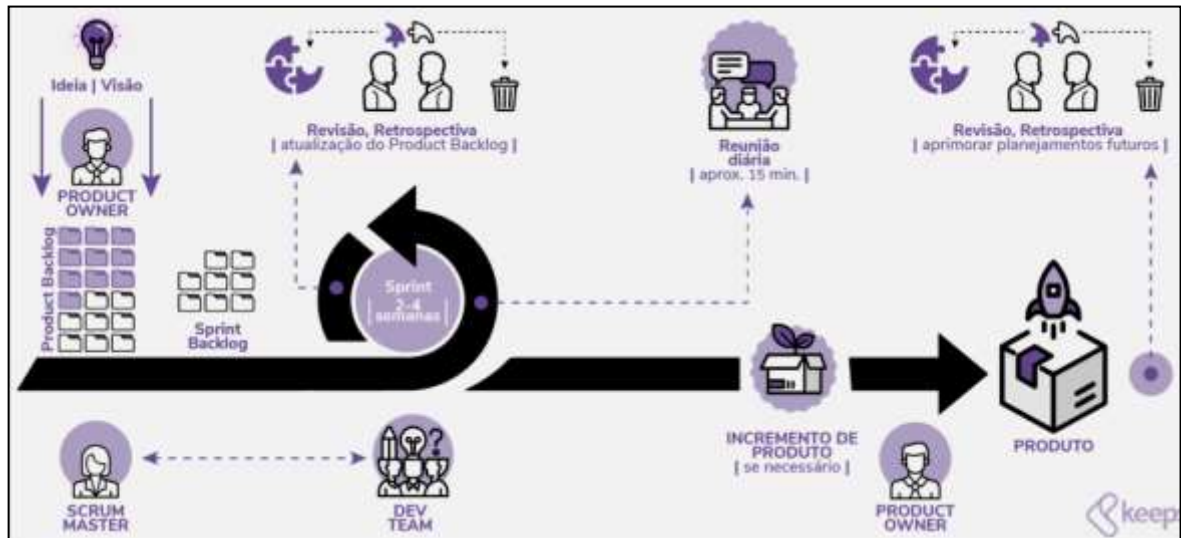
"Um incremento é um trampolim concreto em direção a Meta do Produto. Cada incremento é adicionado a todos os incrementos anteriores e completamente verificados, garantindo que todos os incrementos funcionem juntos. A fim de fornecer valor, o incremento deve ser utilizável".

"Vários incrementos podem ser criados em uma *Sprint*. A soma dos incrementos é apresentada na *Sprint Review*, apoiando assim o empirismo".

"No entanto, um incremento pode ser entregue aos stakeholders antes do final do Sprint. A *Sprint Review* nunca deve ser considerada um marco para liberar valor".

A Figura 9 representa um resumo funcional da Metodologia *Scrum*.

Figura 9 - Esquema de funcionamento da metodologia Scrum.



Fonte: Costa (2022).

6 BANCO DE DADOS

Este capítulo é composto pelas seções: 5.1 Necessidade para desenvolver um banco de dados a 5.7 Vantagens do banco de dados.

6.1 Necessidade para desenvolver um banco de dados

Para desenvolver um banco de dados, deve-se saber as necessidades que devem ser inseridas o banco de dados.

Segundo a Equipe PM3 (2023), "Entender qual é a demanda para a criação do banco de dados é o primeiro passo para criar um sistema eficiente para armazenar e manipular dados".

Saber o porquê se deve desenvolver um banco de dados é bastante importante para organização que irá trabalhar com dados, assim, identificar os motivos irá simplificar os efeitos negativos na criação de sistema eficiente. (Id., 2023)

"Um banco de dados bem estruturado pode otimizar significativamente o dia a dia de uma empresa, principalmente em negócios que lidam com grandes quantidades de dados e precisam acessá-los e manipulá-los de maneira simples e prática". (Id., 2023)

6.2 O que é um banco de dados?

"A expressão Banco de Dados originou-se do termo inglês *datbanks*. Este foi trocado pela palavra *Databases* - Base de Dados - devido possuir significação mais apropriada". (COSTA, 2011)

Date (1985) define "Um Banco de Dados é uma coleção de dados operacionais armazenados, sendo usados pelos sistemas de aplicação de uma determinada organização".

Segundo Costa (2011), "Um banco de dados é uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa".

"Um banco de dados é uma coleção de dados relacionados" (ELMASRI; NAVATHE, 2010)

"Um banco de dados é uma coleção organizada e estruturada de informações, armazenadas de forma a permitir o acesso, a manipulação e a recuperação eficiente dos dados". (ALMEIDA, 2023)

Sendo um dispositivo estruturado e organizado para armazenar dados, o banco de dados pode estar dentro de sistemas computacionais, com intuito de melhorar o gerenciamento de controle dos dados. De certa forma, o banco de dados acaba sendo um recurso importante para coleta e armazenamento de dados, utilizado bastante em questões diversas do dia a dia. (Id., 2023)

"Bancos de dados armazenam dados na forma de uma coleção de tabelas, onde cada tabela armazena dados sobre uma entidade particular". (PRESCOTT, 2015)

Com essa definição demonstra-se que "Nesse modelo, os dados são organizados em linhas e colunas, onde cada tabela representa uma entidade ou um conjunto de entidades relacionadas". (ALMEIDA, 2023)

Por exemplo, quando se está gerenciando as informações de um determinado setor de uma empresa, neste setor há dados dos funcionários, função, salário e metas. Os dados serão armazenados em tabelas, tendo os registros das informações em cada coluna de forma organizada e controlada. (Id., 2023)

De acordo com Costa (2011), um banco de dados traz algumas vantagens:

- **Controle centralizado de dados** - os dados estão concentrados em um único local e isto proporciona um maior controle. Na abordagem de processamento de arquivos os dados estão dispersos, pois cada aplicação mantém arquivos de dados próprios;
- **Controle da redundância, redução do espaço de armazenamento e compartilhamento de dados** - no enfoque de banco de dados o dado é armazenado apenas uma vez e pode ser compartilhado (de forma concorrente ou não) por diversos usuários;
- **Eliminação de inconsistências e garantia de integridade** - no método tradicional baseado em arquivos, dada a repetição de informação armazenada, pode acontecer de um mesmo dado apresentar valores divergentes. Em banco de dados é possível manter a consistência e a integridade dos dados;

- **Estabelecimento de padrões e facilidade de acesso aos dados** - na abordagem de banco de dados, devido à centralização dos dados, torna-se mais propício instituir padrões de nomenclatura e documentação. Devido a essa padronização a recuperação de informação é mais eficiente;
- **Independência de dados** - no sistema de arquivos, a definição da estrutura de armazenamento e do método de acesso aos dados está inclusa no código das aplicações.

Pode se afirmar que um banco de dados é bastante importante para o registro de dados, essencial para o dia a dia, pois com a grande quantidade de dados que se pode coletar um banco de dados se torna fundamental para armazenar estes dados.

Um banco de dados é composto por três componentes principais: (EDU, 2023)

- **Dados:** são as informações que você deseja armazenar no banco de dados. Pode ser qualquer coisa, desde nomes de clientes e números de telefone até registros de vendas ou informações científicas complexas;
- **Sistema de Gerenciamento de Banco de Dados (SGBD) ou (DBMS):** é o software responsável por gerenciar o banco de dados. Ele fornece interfaces para criar, modificar e consultar os dados, além de garantir a segurança e a integridade deles;
- **Hardware:** refere-se aos dispositivos físicos nos quais o banco de dados é armazenado, como servidores de banco de dados e unidades de armazenamento.

Ressalta-se que esses três componentes são fundamentais para se ter a noção de como o banco de dados tem suas principais dinâmicas funcionais, observando a forma como o banco de dados trabalha com os dados, o meio como são registrados, a maneira como é gerenciado pelo próprio sistema de gerenciamento do banco de dados e o modo que são armazenados. (Id., 2023)

6.3 O que são dados?

Elmasri; Navathe (2010) informam "Com dados, queremos dizer fatos conhecidos que podem ser registrados e possuem significado implícito".

"Dados são observações documentadas ou resultados da medição. A disponibilidade dos dados oferece oportunidades para a obtenção de informações". (PINHEIRO, 2023)

Os dados tendem ser tudo aquilo que são registrados, por exemplo, uma faculdade definiu algumas salas por porcentagem de alunos masculinos e femininos, a diferença de idade e a região que cada um mora. Esses dados captados pela faculdade podem ser utilizados para medir, a quantidade de homens ou mulheres, verificar se há alunos mais novos ou mais velhos e por último analisar os alunos por onde moram, se moram em uma distância considerada ou na própria cidade onde a faculdade se localiza. (Id., 2023)

Com isso pode-se analisar como os dados serão tratados em determinadas situação, lembrando sempre que, em geral, dados é tudo que é captado, em mundo físico, comportamental ou sensorial, sendo observados ou medidos para se tornarem dados. (Id., 2023)

6.4 Banco de dados na empresa

Um banco de dados será empregado em empresas em nível baixo ou alto, deverá ser entendida a necessidade que a empresa tem, assim, será possível definir qual o melhor banco de dados para atender suas necessidades.

"Um empreendedor de pequeno porte, com um negócio que gera poucos arquivos e necessita de um nível de processamento mais limitado, pode optar por uma edição mais básica de um Banco de Dados". (MICROSERVICE, 2023)

"Empresas de todos os segmentos precisam se adaptar a esse contexto digital para tomar decisões mais inteligentes, principalmente as que trabalham diretamente com tecnologia e produtos digitais como receita". (PM3, 2023)

"Já uma empresa maior, que precisa de serviços *Big Data*, ou seja, com imenso volume de dados, tem a necessidade de mais velocidade para processamento

de mais informações e versões mais robustas do Banco de Dados". (MICROSERVICE, 2023)

6.5 Criação do modelo de banco de dados

Para aplicar o banco de dados em um SGBD, primeiramente, deve-se analisar a criação de modelo de banco de dados, "deve-se sempre seguir uma sequência de etapas cujo resultado será o desenho correto de um modelo que represente o mais fielmente possível o universo para o qual a aplicação está sendo desenvolvida". (IMPACTA, 2017)

"Esse processo é chamado de modelagem de dados e segue algumas etapas". (PM3, 2023)

- **Análise de requisitos:** é o passo anterior, em que se define as regras de negócio e o objetivo do banco de dados junto aos *stakeholders*;
- **Modelagem conceitual:** criação de um modelo conceitual, com um diagrama simples, descrevendo os elementos que serão incluídos no BD. Essa relação, normalmente, é feita em um Diagrama ER (Entidade-Relacionamento);
- **Modelagem lógica:** o modelo lógico é o passo seguinte no processo de modelagem, no qual utiliza-se do modelo conceitual para criar uma relação lógica no diagrama definido previamente. Aqui serão definidas as chaves (primárias e estrangeiras) e os tipos de dados, por exemplo;
- **Modelagem física:** etapa mais técnica, onde acontece a implementação do modelo lógico criado. A modelagem física prepara o modelo para ser lido em um SGBD.

A manutenção também é essencial nesse processo, uma vez que o banco de dados pode requerer mudanças e atualizações em alguns momentos, além da necessidade de fazer *backups* para manter os dados seguros. (Id., 2023)

6.6 Tipos de banco de dados

Após analisar a necessidade do banco de dados será preciso saber qual tipo de banco de dados escolher, porém há dois tipos de banco de dados:

- **Banco de dados relacional:** "Um banco de dados relacional é um tipo específico de banco de dados que organiza e armazena os dados em tabelas com estrutura pré-definida". (ALMEIDA, 2023)
- **Banco de dados não relacional:** "Os bancos de dados não relacionais são mais adequados para aplicações que exigem alta escalabilidade e flexibilidade no esquema de dados". (CALANCA, 2023)

"Os bancos de dados relacionais quanto os não relacionais têm seus pontos fortes e fracos, e a escolha entre um ou outro dependerá das necessidades específicas de cada aplicação". (CALANCA, 2023)

Com a definição de Calanca (2023), pode se notar que estes tipos de banco de dados levam em conta a necessidade de como a organização usará seu banco de dados, sendo bastante importante observar, em primeiro lugar, a necessidade para depois se analisar com qual tipo de banco de dados será manuseado.

"Para os bancos de dados relacionais, temos também o conceito de SGBDR (Sistema de Gerenciamento de Banco de Dados Relacional), que é um tipo específico de SGBD projetado para armazenar, gerenciar e recuperar dados em um formato relacional". (ALMEIDA, 2023)

Para se trabalhar com o banco de dados relacional é necessário ter-se conhecimento de alguns tipos de linguagem, pois os principais SGBDs relacionais utilizam uma dessas opções linguagem: (PM3, 2023)

- **PostgreSQL:** é *open source*, suporta várias linguagens de programação, tem grande potencial de escalabilidade e é muito estável, sendo uma das opções mais avançadas do mercado;
- **SQL Server:** pertence à Microsoft, possibilitando a integração com vários serviços da empresa, é usado tanto a nível profissional, quanto empresarial;

- **Oracle DB:** amplamente usado no contexto empresarial, possui muita estabilidade e pode-se compartilhar recursos, pertence à Oracle Corporation;
- **MySQL:** um dos mais usados no mercado, é *open source*, comporta múltiplas plataformas e é muito usado em sites (principalmente WordPress) e aplicativos.

Quanto ao banco de dados não relacional, também é conhecido como NoSQL, "Esse tipo de *database* não segue um modelo de tabela estruturado e pré-definido com linhas e colunas como nos bancos de dados relacionais". (PM3, 2023)

6.7 Vantagens do banco de dados

Para o uso do banco de dados nas organizações é fundamental saber suas diversas vantagens que ela pode impor dentro das empresas:

- **Garantem mais segurança:** "As informações que estão alocadas nos Bancos de Dados ficam estão menos sujeitas a ataques e outras possibilidades de perdas". (MICROSERVICE, 2023)
Essa consistência é essencial para as empresas, especialmente se coletam dados de várias maneiras ou empregam várias técnicas de coleta de dados internas e externas. (VIBE, 2023)
- **Agilizam o acesso às informações:** "Um Banco de Dados eficiente permite a troca de informações sem falhas entre as diversas áreas da empresa". (MICROSERVICE, 2023)
"Quando os dados são armazenados em um banco de dados por meio de um sistema de gerenciamento de banco de dados, os usuários têm acesso mais fácil a uma ampla gama de informações em um único local. Como resultado, as pessoas podem reagir rapidamente e fazer melhores julgamentos". (VIBE, 2023)
- **Permitem o alinhamento das estratégias:** "Usando banco de dados e outras ferramentas de computação e inteligência de negócios, as organizações agora podem aproveitar os dados que coletam para executar

com mais eficiência, permitir uma melhor tomada de decisões e se tornar mais ágeis e escaláveis". (VIBE, 2023)

"Com as informações disponíveis e com possibilidade de cruzamento de dados, os usuários do Banco de Dados podem ter uma visão mais completa do seu negócio e traçar estratégias mais alinhadas com a realidade, e não baseadas apenas em suposições ou estimativas". (MICROSERVICE, 2023)

- **Ajudam a aumentar a produtividade:** "Com os números da empresa mais organizados e com fácil acesso, outra consequência positiva é o aumento da produtividade". (MICROSERVICE, 2023)

Banco de dados são essenciais para garantir uma organização e otimização de processos empresariais, principalmente com a necessidade de manejo de grandes quantidades de dados. (PM3, 2023)

7 SIMULAÇÃO

Para melhor explicar a utilização da Metodologia *Scrum* no desenvolvimento de um banco de dados, decidiu-se utilizar o *Kanban Board Game*, que consiste em um jogo que é um Sistema *Kanban* simples onde se simula o desenvolvimento de software, no qual há um acúmulo de cartões que representam histórias para desenvolvimento.

As histórias são puxadas do "*Backlog*" para uma coluna "Pronto", depois para "Análise", "Desenvolvimento", "Teste" e finalmente "Implantado".

Cada história especifica a quantidade de trabalho necessária a processar. Os resultados aleatórios dos dados representam a variabilidade de quanto se é capaz fazer a cada dia.

Todos os dias, um número aleatório entre um e seis é usado para determinar quanto trabalho foi completado.

Existem diferentes cores que representam especialistas em cada área (Analistas, Desenvolvedores, Testadores). Caso o integrante atue em sua área de especialização esse número é duplicado.

Utilizando essa plataforma, juntamente da Metodologia *Scrum*, realizou-se a seguinte simulação: antes de iniciar o projeto, o *Product Owner* do produto realizou uma reunião com o *stakeholder* para ajudá-lo a compreender todas as necessidades e expectativas associadas a este projeto de banco de dados.

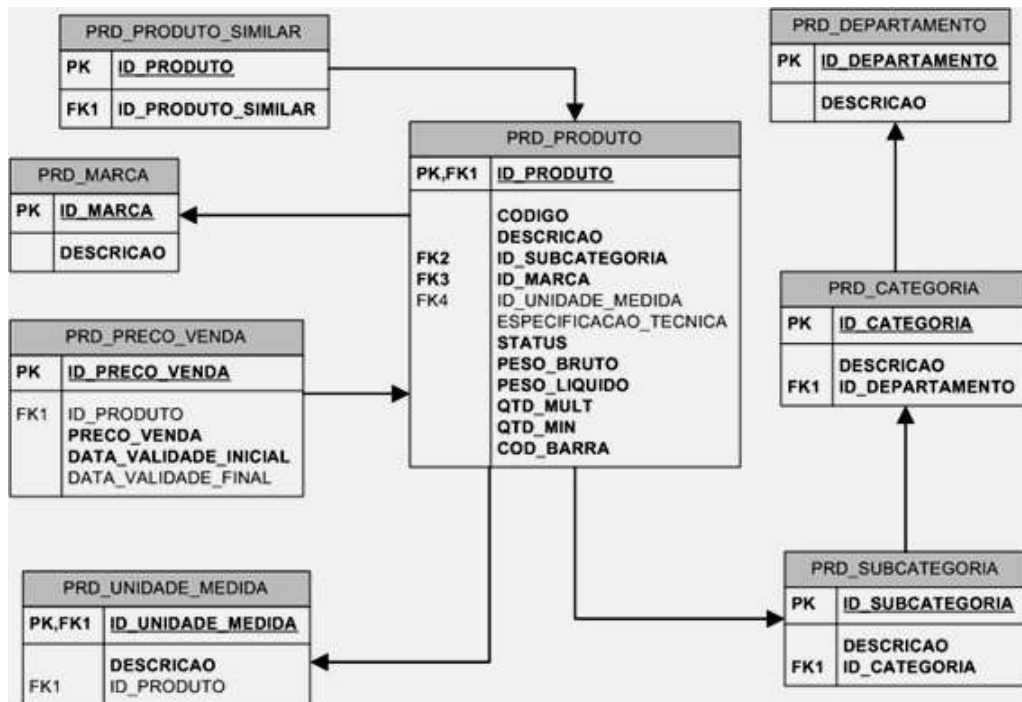
Foi registrada a seguinte demanda:

- Um banco de dados simples que permita aos funcionários controlarem facilmente a entrada e saída de produtos alimentícios.

Após entender as necessidades do interessado, todo o *Scrum Team* se reuniu para elaborar o *Backlog* do Produto.

Após ter descoberto a necessidade do *stakeholder*, todo o *Scrum Team* se reuniu para desenvolver o *Product Backlog* e conseguiram desenvolver o seguinte MER (Modelo de Entidades e Relacionamentos), para uma maior evidência de como funcionaria o projeto - Figura 10.

Figura 10 - Tabelas do projeto do banco de dados.



Fonte: Camargo (2010).

Em seguida, definiram os *Backlogs* da *Sprint* para um *sprint* de cinco dias, com as seguintes prioridades ilustradas nas Tabelas 1e 2.

Tabela 1 - Backlog menor tempo da Sprint.

MENOR TEMPO				
Tarefas	Valor	Tempo	Tipo	Nome da tarefa
I2	0	12	Documentação	CONSULTA FATURAMENTO
F1	150	15	Prazo fixo	CONSULTA VENDAS REEMBOLSADAS
I1	0	16	Documentação	DOCUMENTAÇÃO
S6	110	23	Valor para o cliente	ATUALIZAÇÃO DO LAYOUT
I3	0	24	Documentação	HELP DESK
S14	100	24	Valor para o cliente	TABELA USUARIO
S1	110	26	Valor para o cliente	CONSULTA PRODUTO
S10	110	27	Valor para o cliente	TABELA FORNECEDORES
S4	100	27	Valor para o cliente	TABELA CLIENTE
S8	100	28	Valor para o cliente	TABELA TRANSPORTADOR
S12	100	29	Valor para o cliente	CONSULTA PRODUTO REJEITADO PELA QUALIDADE
S13	100	30	Valor para o cliente	TABELA PRODUTO
S3	110	31	Valor para o cliente	TABELA PRODUTO SIMILAR
S11	130	32	Valor para o cliente	TABELA MARCA
S2	120	32	Valor para o cliente	TABELA PREÇO DE VENDA
F3	100	36	Prazo fixo	TABELA UNIDADE DE MEDIDA
S7	120	36	Valor para o cliente	TABELA DEPARTAMENTO
S9	120	36	Valor para o cliente	TABELA CATEGORIA
S5	130	37	Valor para o cliente	TABELA SUBCATEGORIA

Fonte: Autores do trabalho (2023).

Tabela 2 - Backlog maior valor da Sprint.

MAIOR VALOR				
Tarefas	Valor	Tempo	Tipo	Nome da tarefa
F1	150	15	Prazo fixo	CONSULTA VENDAS REEMBOLSADAS
S11	130	32	Valor para o cliente	TABELA MARCA
S5	130	37	Valor para o cliente	TABELA SUBCATEGORIA
S2	120	32	Valor para o cliente	TABELA PREÇO DE VENDA
S7	120	36	Valor para o cliente	TABELA DEPARTAMENTO
S9	120	36	Valor para o cliente	TABELA CATEGORIA
S6	110	23	Valor para o cliente	ATUALIZAÇÃO DO LAYOUT
S1	110	26	Valor para o cliente	CONSULTA PRODUTO
S10	110	27	Valor para o cliente	TABELA FORNECEDORES
S3	110	31	Valor para o cliente	TABELA PRODUTO SIMILAR
S14	100	24	Valor para o cliente	TABELA USUARIO
S4	100	27	Valor para o cliente	TABELA CLIENTE
S8	100	28	Valor para o cliente	TABELA TRANSPORTADOR
S12	100	29	Valor para o cliente	CONSULTA PRODUTO REJEITADO PELA QUALIDADE
S13	100	30	Valor para o cliente	TABELA PRODUTO
F3	100	36	Prazo fixo	TABELA UNIDADE DE MEDIDA
I2	0	12	Documentação	CONSULTA FATURAMENTO
I1	0	16	Documentação	DOCUMENTAÇÃO
I3	0	24	Documentação	HELP DESK

Fonte: Autores do trabalho (2023).

A seguir, foi elaborado *Sprint Planning*, onde estabeleceram as prioridades das histórias e a estimativa de tempo necessária para cada uma delas.

9º dia: começou-se a simulação a partir do nono dia. Foi realizado um *Daily* com as necessidades do dia - Figuras 11 e 12:

- Para esse dia, decidiu-se concentrar os esforços em avançar o máximo possível nas tarefas de menor complexidade que proporcionam o maior valor;
- Testers: S1, S2 e S3, para que pudessem avançar o máximo possível para enviar para o *Deployed*;
- Desenvolvedores: um para o L1 e outro para S6 com intuito de concluir a tarefa e liberá-la para os *Testers*;

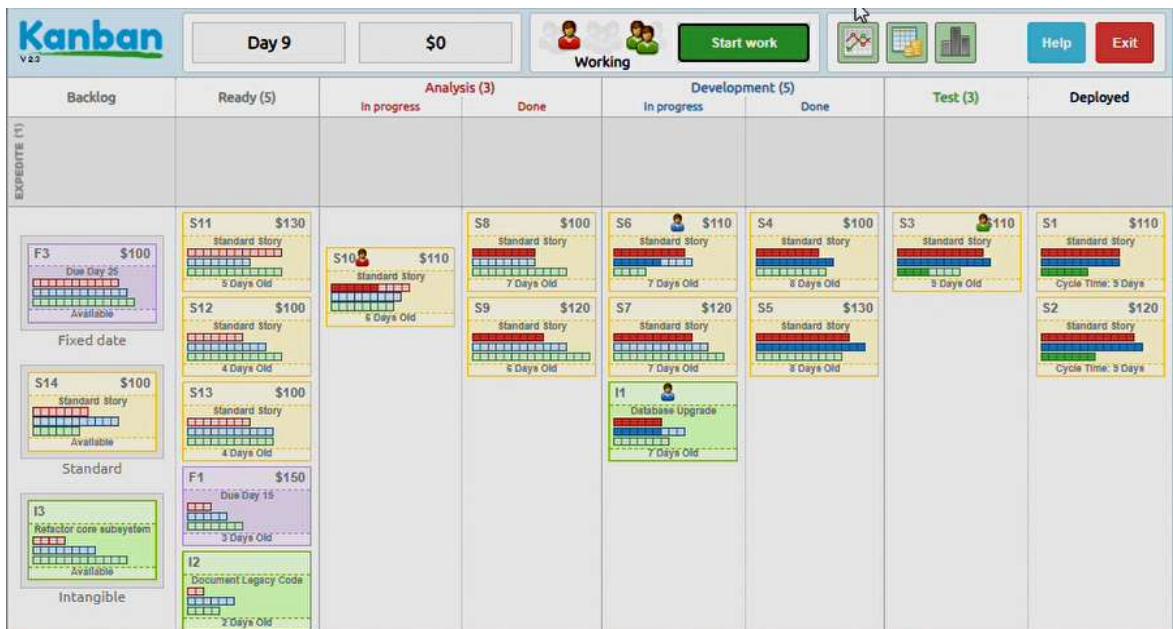
- Analistas: S9 porque era uma tarefa que já tinha sido iniciada no dia anterior e adicionalmente S8 por ser uma tarefa de menor complexidade:

Figura 11 - Início do dia 09.



Fonte: Autores do trabalho (2023).

Figura 12 - Fim do dia 09.



Fonte: Autores do trabalho (2023).

10º dia: ao se realizar o *Daily*, o time informou que as tarefas conseguiram ser concluir.

Foram feitas duas entregas para o *Deployed*.

Definiu-se a mesma estratégia do dia anterior: um membro do *squad* para cada função - Figuras 13 e 14:

- Testers: um para cada etapa visando a conclusão do maior número de atividade com a menor quantidade de *Testers* para cada tarefa;
- Desenvolvedores: focados nas S6 e L1, uma vez que essas tarefas estavam próximas de serem concluídas, com apenas algumas horas restantes para sua finalização, o objetivo era liberá-las no dia seguinte para serem testadas;
- Analistas: deixou-se um analista para concluir a S10 e outro para iniciar a atividade de prazo fixo, com término previsto antes da primeira *Sprint*.

Figura 13 - Início do dia 10.



Fonte: Autores do trabalho (2023).

Figura 14 - Fim do dia 10.



Fonte: Autores do trabalho (2023).

11º dia: na reunião diária, o time informou que nem todas as tarefas conseguiram ser entregues.

Foi realizada apenas uma entrega e quase chegou-se à conclusão de mais duas, faltando apenas dois quadrados da tarefa S3 e um quadrado da tarefa S5 do *tester* - Figuras 15 e 16:

- Contou-se com um *Tester* a menos na equipe, pois quebrou a perna e não conseguiu vir trabalhar;
- Estipulou-se um limite às tarefas de andamento. Decidiu-se colocar um *Tester* para auxiliar a equipe de análise com a tarefa S10, que estava inacabada e um *Tester* para concluir a tarefa L1;
- Os *Testers* não foram deixados nas mesmas tarefas do dia 10, a prioridade foi otimizar a eficiência no menor período, em vez de focar na entrega em larga escala.

Figura 15 - Início do dia 11.



Fonte: Autores do trabalho (2023).

Figura 16 - Fim do dia 11.



Fonte: Autores do trabalho (2023).

12º dia: no *Daily* o time informou que as tarefas conseguiram ser entregues.

Não foi realizada nenhuma entrega, porém desafogou-se o limite dos desenvolvedores.

Desta vez optou-se pelo maior número de entregas. Imaginou-se que o andamento do dia anterior teria sido mais eficaz, mas como não houve entregas e as

atividades já estavam com até onze dias sem conclusão, decidiu-se pela maior entrega e menor eficiência - Figuras 17 e 18:

- Tester: um para cada etapa de teste;
- Desenvolvedor: entrou mais um desenvolvedor para equipe;
- Analista: um para cada etapa de análise.

Figura 17 - Início do dia 12.



Fonte: Autores do trabalho (2023).

Figura 18 - Fim do dia 12.



Fonte: Autores do trabalho (2023).

13º dia: no Daily o time informou que se realizaram duas entregas.

Enviou-se um *Tester* para a tarefa F1, de prazo determinado, para que possa ser concluída o quanto antes e liberar a área de teste, já que houve um problema na tarefa S5 - Figuras 19 e 20:

- Testers: destinou-se um desenvolvedor e um *Tester* para a tarefa S5, pois foi descoberto um erro crítico ao entrar na área de teste. Adicionou-se dois membros da *squad*, na mesma tarefa, para mitigar o quanto antes a situação;
- Desenvolvedor: encaminhou-se um desenvolvedor para concluir a tarefa S7 e outro para iniciar a tarefa S10 devido a sua baixa complexidade para os desenvolvedores;
- Analista: definiu-se um para cada tarefa, sendo a S14 de menor complexidade e a S11 de maior complexidade e valor.

Figura 19 - Início do dia 13.



Fonte: Autores do trabalho (2023).

Figura 20 - Fim do dia 13.



Fonte: Autores do trabalho (2023).

A Tabela 3, a seguir, apresenta a comparação entre a gestão tradicional - PMBOK - e a ágil - Scrum - no desenvolvimento de projetos de software.

Tabela 3 - Comparativo entre PMBOK e Scrum.

CARACTERÍSTICAS	GESTÃO TRADICIONAL - PMBOK	GESTÃO ÁGIL - SCRUM
Ter definido a priori	Escopo.	Tempo (Sprints).
Responsável pela organização para atingir os objetivos do projeto	Gerente de projeto.	Scrum Master.
Frequência de reuniões de status	Dependendo da complexidade/necessidade do projeto, alinhar a frequência.	Diárias.
Escopo	Bem definido nas fases iniciais do projeto e formalização por meio da WBS (Work Breakdown Structure).	Escopo é definido em alto nível e os requisitos são priorizados e definidos de forma iterativa. Necessita de maior controle de planejamento.
Tempo	Cronograma detalhado para a realização de todo o projeto.	Cronograma orientado a produto com entregas incrementais de duas a quatro semanas.
Custo	Monitoração das alterações para que não altere o custo planejado.	Maior controle em função da rapidez na incorporação de alterações.
Qualidade	Processos de verificação, validação e plano de testes.	Programação em pares, testes incrementais e refatoração.
Riscos	Análise de riscos durante todo o ciclo de vida do projeto.	Aplica-se o mesmo conceito da gestão tradicional.
Comunicação	Formal e documentada.	Implícita, interpessoal e colaborativa.
Recursos Humanos	Papéis claros e bem definidos.	Confiança nos membros da equipe e ambiente colaborativo.
Aquisição	Controle por contrato e escopo bem definido e documentado.	Presença do cliente, volatilidade de requisitos e pouca documentação.
Integração	Plano de projeto detalhado e controle total do projeto pelo gerente.	Plano do projeto evolutivo e gerente do projeto atuam com facilitador.

Fonte: Adaptado MundoPM (2021, p. 44).

7 CONCLUSÃO

Verificou-se por meio da simulação a importância da metodologia *Scrum* em um desenvolvimento de um banco de dados. Evidenciou-se na *Sprint* de cinco dias os três pilares do *Scrum*: transparência, inspeção e adaptação.

A transparência foi alcançada por meio das reuniões diárias e do quadro *Kanban*, que proporcionaram uma visão clara do progresso das atividades, destacando os desafios e aprendizados de cada dia.

A inspeção permitiu a avaliação constante do projeto, identificando erros e ajustes necessários ao longo do processo. Na adaptação, a equipe conseguiu gerenciar eficazmente recursos e situações imprevistas, como o acidente de um *Tester* e a resolução conjunta de um erro crítico na fase de testes.

Comparando com a metodologia tradicional em cascata, fica claro que o *Scrum* oferece vantagens significativas. Na abordagem tradicional, a burocracia e a rigidez podem dificultar a flexibilidade da equipe. A solicitação de recursos extras ou a realização de tarefas paralelas, frequentemente, demandam procedimentos demorados, como relatórios e aprovações. Além disso, a metodologia tradicional, muitas vezes, foca a entrega metódica de tarefas, em oposição ao *Scrum* que prioriza a entrega incremental de valor ao cliente.

No método tradicional só há possibilidade, em alguns casos, de desempenhar uma outra tarefa do projeto, caso a tarefa em andamento já estivesse concluída, diferente do *Scrum*, ele por viabilizar entregas de valor para o cliente nem sempre foca na entrega de tarefas de forma metódica, não necessita uma grande quantidade de documentação, pois se fundamentando em seus pilares todo time ágil consegue ser corresponsável e compreender o andamento da tarefa.

O desenvolvimento do banco de dados é feito de forma incremental, priorizando funcionalidades de alto valor. Testes e *feedback* ocorrem após cada *Sprint*, permitindo adaptações e melhorias contínuas. A documentação é criada incrementalmente e a implantação é realizada regularmente.

A metodologia ágil permite que a equipe se adapte às mudanças nos requisitos e nas necessidades dos usuários, garantindo a entrega de valor ao longo do tempo e mantendo o banco de dados eficiente e atualizado.

Conclui-se que a Metodologia *Scrum* pode ser considerada mais eficaz que a metodologia tradicional no desenvolvimento de banco de dados por causa da

flexibilidade e adaptabilidade, a simulação demonstra que a metodologia *Scrum* é altamente benéfica no desenvolvimento de bancos de dados, permitindo maior transparência, inspeção contínua, adaptação ágil e uma entrega de valor mais eficiente ao cliente, em comparação com a metodologia tradicional em cascata.

REFERÊNCIAS

AGILEMANIFESTO. **Princípios por trás do Manifesto Ágil**. Publicado em: 2001. Disponível em: <<https://agilemanifesto.org/iso/ptbr/principles.html>>. Acesso em: 10 set. 2023.

ALMEIDA, Marcus. **Banco de dados relacionais**: conhecendo conceitos, terminologias e ferramentas. Publicado em: 14 jul. 2023. Disponível em: <<https://www.alura.com.br/artigos/banco-dados-relacionais-conceitos-terminologias-ferramentas>>. Acesso em: 27 set. 2023.

ALURA. **Scrum**: A meta da Sprint. Publicado em: 18 abr. 2016. Disponível em: <<https://www.alura.com.br/artigos/scrum-a-meta-da-sprint#:~:text=Seria%20muito%20mais%20interessante%20se,pagamento%20com%20cart%C3%A3o%20de%20cr%C3%A9dito>>. Acesso em: 30 maio 2024.

BRASILEIRO, Roberto. **Manifesto Ágil, o que é e qual a sua história**. Publicado em: 2020. Disponível em: <<https://www.metodoagil.com/manifesto-agil/>>. Acesso em: 06 set. 2023.

BORCHARDT, Guilherme Augusto; CAMPOS, Willian Rafael; KUBOTA, Flávio Issao. **Metodologias de gestão de projetos**: compreendendo as abordagens de gestão de projetos em duas empresas multinacionais. Disponível em: <https://abepro.org.br/biblioteca/TN_ST_387_1916_43080.pdf>. Acesso em: 02 nov. 2023.

CALANCA, Paulo. **SQL e NoSQL**: trabalhando com bancos relacionais e não relacionais. Publicado em: 16 ago. 2023. Disponível em: <<https://www.alura.com.br/artigos/sql-nosql-bancos-relacionais-nao-relacionais>>. Acesso em: 20 set. 2023.

CAMARGO, Welligton Balbo de. **SQL Server 2008**: Conceitos de MER e DER, Regras e Tipos de Relacionamentos. Publicado em: 06 nov. 2010. Disponível em: <<https://programandodotnet.wordpress.com/2010/11/06/sql-server-2008-%E2%80%93-parte-9-%E2%80%93-conceitos-de-mer-e-der-regras-e-tipos-de-relacionamentos/>>. Acesso em: 20 set. 2023.

CINTRA, Paulo Silas Carboni; JARDINI, Camila Canavez; JERÔNIMO, Naiara Prado; COSTA, Flávio Henrique de Oliveira. **Aplicação de Scrum como ferramenta de otimização de processos em uma empresa de assessoria de comércio digital**. Publicado em: 2018. Disponível em:

<<http://periodicos.unifacef.com.br/index.php/creare/article/view/1634>>. Acesso em: 26 out. 2023.

COSTA, Elisângela. **Bancos de dados relacionais**. Publicado em: 2011. Disponível em: <<http://www.fatecsp.br/dti/tcc/tcc0025.pdf>>. Acesso em: 20 set. 2023.

_____, Renia. **Metodologia Scrum**. Publicado em: 24 out. 2022. Disponível em: <<https://pt.linkedin.com/pulse/metodologia-scrum-renia-vieira>>. Acesso em: 30 maio 2024.

DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. São Paulo: Campus, 1985.

EDU, Prof. **O que é um banco de dados e como funciona?** Publicado em: 17 jun. 2023. Disponível em: <<https://federalcubatao.com.br/banco-de-dados/o-que-e-um-banco-de-dados-e-como-funciona/>>. Acesso em: 13 set. 2023.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Bancos de Dados**. 6. ed. São Paulo: Addison-Wesley, 2010.

ESPINHA, Roberto. **Metodologia de Projeto**: saiba quais são e como escolher a melhor. Publicado em: 14 set. 2021. Disponível em: <<https://artia.com/blog/tudo-sobre-metodologia-de-projetos/>>. Acesso em: 02 out. 2023.

FIA. **Metodologias ágeis**: o que são, tipos e principais vantagens. Publicado em: 27 maio 2022. Disponível em: <<https://fia.com.br/blog/metodologias-ageis/>>. Acesso em: 15 set. 2023.

IMPACTA. **Como elaborar um projeto de banco de dados?** Publicado em: 10 fev. 2017. Disponível em: <<https://www.impacta.com.br/blog/veja-como-elaborar-um-projeto-de-banco-de-dados/>>. Acesso em: 21 out. 2023.

JASON, Johanes. **Agile and Scrum Methodology**. Publicado em: 19 mar. 2022. Disponível em: <<https://jaenhosjaon.medium.com/agile-and-scrum-methodology-32a14ba3746f>>. Acesso em: 20 out. 2023.

LOSNAK, Giulia. **Metodologia Ágil – O que é?** Publicado em: 04 jul. 2023. Disponível em: <<https://www.alura.com.br/artigos/o-que-e-metodologia-agil>>

LLORENTE, J. A. **A transformação digital**. Publicação em: 2016. Disponível em: <https://www.revistauno.com.br/wpcontent/uploads/2013/09/160520_UNO24_BR.pdf>. Acesso em: 30 out. 2023.

MICROSERVICE. **Por que minha empresa precisa de um Banco de Dados?** Disponível em: <<https://www.microserviceit.com.br/por-que-minha-empresa-precisa-de-um-banco-de-dados/>>. Acesso em: 20 out. 2023.

MUNDOPM. Revista Mundo Project Management. n. 36, Dez. 2020/Jan. 2021. Disponível em: <<http://www.mundopm.com.br/destaquesEd36.shtml>>. Acesso em: 20 out. 2023.

NASCIMENTO, Bruno Jaime. **Desenvolvimento de software e metodologias: Fatores e dificuldades que influenciam na adoção de metodologias de desenvolvimento**. Trabalho de Conclusão de Curso. Instituto Federal de Santa Catarina. Florianópolis: IFSC, 2019. Disponível em: <<https://repositorio.ifsc.edu.br/bitstream/handle/123456789/1231/TCC-Bruno-Jaime-Nascimento.pdf?sequence=1&isAllowed=y>>. Acesso em: 01 nov. 2023.

PICHLER, Roman. Product teams in scrum. Publicado em: 04 maio 2022. Disponível em: <<https://www.romanpichler.com/blog/product-teams-in-scrum/>>. Acesso em: 30 maio 2024.

PINHEIRO, Hildete Prisco. **O que são dados?** Publicado em: 2023. Disponível em: <<https://www.ime.unicamp.br/~hildete/dados.pdf>>. Acesso em: 13 set. 2023.

PM3, Equipe. **Banco de dados: o que é, importância, tipos e como criar um**. Publicado em: 19 out. 2023. Disponível em: <<https://www.cursospm3.com.br/blog/banco-de-dados/>>. Acesso em: 22 out. 2023.

PONTES, Thiago Bessa; Arthaud, Daniel Dias Branco. **Metodologias ágeis para o desenvolvimento de softwares**. Ciência e Sustentabilidade. v. 4, n. 2, pp. 173-213,. Publicado em: 14 mar. 2019. Disponível em: <<https://doi.org/10.33809/2447-4606.422018173-213>>. Acesso em: 01 nov. 2023.

PRESCOTT, Preston. **SQL para iniciantes**. (S.l.): Babelcube Inc., 2015.

ROGERS, David L. **Transformação digital: repensando o seu negócio para a era digital**. 1. ed. Belo Horizonte: Autêntica Business, 2017.

SCHWABER, Ken. SUTHERLAND, Jeff. **Guia do Scrum - um guia definitivo para o Scrum: as regras do jogo.** Publicado em: nov.2020. Disponível em: <https://www.diariomunicipal.sc.gov.br/arquivosbd/atos/2022/06/1656457532_leiassinado0025690001.pdf>. Acesso em: 01 nov. 2023.

SEMEDO, Maria J. M. **Ganhos de produtividade e de sucesso de Metodologias Ágeis vs Metodologias em Cascata no desenvolvimento de projectos de software.** Publicado em: mar. 2012. Universidade Lusófona. Disponível em: <<https://recil.ensinolusofona.pt/bitstream/10437/6174/1/Disserta%C3%A7%C3%A3o-Maria%20Semedo%5Bentrega%5D.pdf>>. Acesso em: 19 out. 2023.

SILVA, D. E. S.; SOUZA, I. T.; CAMARGO, T. **Metodologias ágeis para o desenvolvimento de software: Aplicação e o uso da metodologia Scrum em contraste ao modelo tradicional de gerenciamento de projetos.** Publicado em: 2013. Disponível em: <<http://revistas.ung.br/index.php/computacaoaplicada/article/view/1408>>. Acesso em: 26 se. 2023.

SOARES, Michel dos Santos. **Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software.** Artigo. Publicado em: 01. nov. 2004. Unipac - Universidade Presidente Antônio Carlos. Faculdade de Tecnologia e Ciências de Conselheiro Lafaiete. Disponível em: <<https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/68/53#:~:text=O%20que%20as%20diferencia%20das,e%20mais%20com%20a%20implementa%C3%A7%C3%A3o>>. Acesso em: 19 out. 2023.

VERONESE, A. **A busca de um novo modelo de gestão para a ciência, tecnologia e inovação na política do MCT.** Publicado em: jan. 2006. Disponível em: <<https://www.scielo.br/j/rap/a/pf3qKpGQjFcs3vhM4xFMzqS/?format=pdf&lang=pt>>. Acesso em: 20 out. 2023.

VIBE, Grupo. **Entenda qual a importância do banco de dados e conheça suas vantagens.** Disponível em: <<https://www.blog.vibetecnologia.com/banco-de-dados#:~:text=Usando%20banco%20de%20dados%20e,tornar%20mais%20%C3%A1geis%20e%20escal%C3%A1veis>>. Acesso em: 20 out. 2023.

WESTPHALEN, Marcelo. Agência Câmara de Notícias. **Especialistas dizem que burocracia dificulta o desenvolvimento de pesquisas no País.** Publicado em: 31 maio 2019. Disponível em: <<https://www.camara.leg.br/noticias/558952-especialistas-dizem-que-burocracia-dificulta-o-desenvolvimento-de-pesquisas-no-pais/>>. Acesso em: 27 set. 2023.