



APLICATIVOS E METODOLOGIAS ÁGEIS

ELDA CARVALHO



CETAM
CENTRO DE EDUCAÇÃO
TECNOLOGICA DO AMAPÁ



AMAZONAS
GOVERNO DO ESTADO

APLICATIVOS E METODOLOGIAS ÂGEIS

EIXO TECNOLÓGICO: INFORMAÇÃO E COMUNICAÇÃO

ELDA CARVALHO

WILSON MIRANDA LIMA
GOVERNADOR DO ESTADO

FÁBIO HENRIQUE DOS SANTOS ALBUQUERQUE
DIRETOR PRESIDENTE

LUANA SILVA DE SOUZA
DIRETORA ACADÊMICA

AMILTON BEZERRA GADELHA
DIRETOR DE RELAÇÕES EMPRESARIAIS E INSTITUCIONAIS

ORLANDO MUNIZ DA SILVA
DIRETOR ADMINISTRATIVO-FINANCEIRO E DE INTERIORIZAÇÃO

MARIA LUCIMAR JACINTO DE SOUSA
DIREÇÃO DA ESCOLA

WANDERLANE GURGEL DO AMARAL
COORDENAÇÃO PEDAGÓGICA

LUCAS CASAS
COORDENAÇÃO DE MÍDIA

ELDA NUNES DE CARVALHO
DOCENTE CONTEUDISTA

©2024 BY CETAM
TODOS OS DIREITOS
RESERVADOS



PALAVRA DO PROFESSOR

Olá. Meu nome é Elda Nunes de Carvalho. Sou formada em Tecnologia em Mecatrônica Industrial, mestre em Informática e especialização em Engenharia de Manutenção Industrial, com experiência técnico-profissional como laboratorista atuando a 01 ano, com atuação em grandes projetos no desenvolvimento de programas e elaboração de solução, e nove anos de experiência como professor do Ensino Técnico e Superior (graduação e pós-graduação) em grandes universidades. Como sou apaixonada pelo que faço e adoro compartilhar minha experiência de vida àqueles que estão iniciando em suas profissões e estudos, estou muito feliz em poder ajudá-lo nessa fase de muito estudo e trabalho. Conte comigo!



ICONOGRAFIA

Esses ícones irão aparecer em sua trilha de aprendizagem. Eles significam:



Indica o ponto de maior relevância no texto.



Quando curiosidades e indagações lúdicas, sobre o tema em estudo, forem necessárias.



Orienta ao estudante que desenvolva atividades de pesquisa, que complementem seus estudos em diferentes mídias: vídeos, filmes, jornais, livros e outras.



Indica a definição de um termo, palavra ou expressão utilizada no texto.



Quando houver necessidade de chamar a atenção sobre algo a ser refletido ou discutido.



Quando for necessário acessar um ou mais sites para fazer download, assistir a um vídeo, ler um texto, ouvir um podcast, etc.



Apresenta dicas de vídeos que enriquecem o assunto tratado.



Apresenta atividades em diferentes níveis de aprendizagem para que o estudante possa realizá-las e conferir o seu domínio do tema estudado.

Com esses ícones, ficará mais fácil nos comunicarmos com você. Basta olhar para um deles e você saberá exatamente o que virá logo em seguida, com uma linguagem direta, lúdica e dialógica. Boa leitura!



SUMÁRIO

| | |
|-----------------------------------------------------------|----|
| Desenvolvimento de Aplicativos e Metodologias Ágeis | 7 |
| Objetivo | 7 |
| Competências | 7 |
| Introdução | 8 |
| Desenvolvimento de Aplicativos e Metodologias Ágeis | 11 |
| Ciclo de Vida de Projetos de Software | 13 |
| Introdução às Metodologias Ágeis: Scrum e Kanban | 15 |
| Scrum | 15 |
| Kanban | 15 |
| Práticas Ágeis em Programação | 16 |
| Desenvolvimento Incremental | 16 |
| Refatoração | 17 |
| Pratique Atividade de Aprendizagem | 18 |
| Referências | 20 |
| Currículo do professor | 21 |

DESENVOLVIMENTO DE APLICATIVOS E METODOLOGIAS ÁGEIS

Objetivo:

Olá! Seja bem-vindo(a) à nossa segunda aula. Nela você irá conhecer o desenvolvimento de aplicativos e metodologias ágeis, especificamente Ciclo de Vida de Projetos de *Software*, Introdução às Metodologias Ágeis: Scrum e Kanban, Práticas Ágeis em Programação e Refatoração. Esses conhecimentos lhe ajudarão a ter sucesso quando você desenvolver aplicativos digitais.

Competências:

Ao final desta unidade você terá desenvolvidos as seguintes competências:

1. Aplicar Metodologias Ágeis no Ciclo de Vida do Desenvolvimento de Aplicativos. Entender e implementar *frameworks* ágeis, como *Scrum* ou *Kanban*, no processo de desenvolvimento de aplicativos, gerenciando *sprints*, tarefas e interações de maneira eficaz para atender prazos e requisitos de qualidade
2. Colaborar com equipes multidisciplinares participando ativamente de reuniões de feedback e revisão de *sprints*, para a promoção de comunicação eficaz.
3. Adaptar-se às mudanças de acordo com as novas demandas dos usuários e mudanças tecnológicas, para aplicar ciclos iterativos de melhoria contínua e garantir a entrega de produtos alinhados com as necessidades do mercado.

Preparado, preparada para uma viagem rumo ao conhecimento? Então vamos lá!



Introdução

Você já parou para pensar como os aplicativos que usamos diariamente são desenvolvidos? Seja um app de mensagens, de transporte ou de redes sociais, todos eles passam por um processo complexo de criação, e é aí que o desenvolvimento de aplicativos entra em cena. Mas, para que esses aplicativos sejam lançados de forma eficiente e com alta qualidade, muitas equipes utilizam as Metodologias Ágeis.

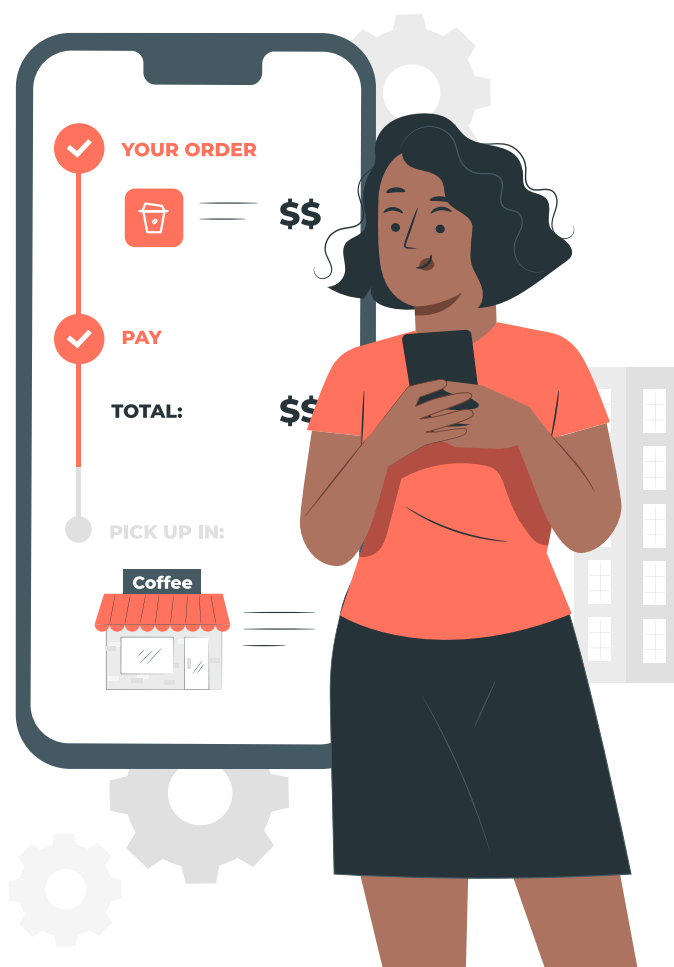
Vamos começar com o básico: o desenvolvimento de aplicativos é o processo de criar *softwares* que podem ser executados em dispositivos móveis, como *smartphones* e *tablets*. Esse processo envolve várias etapas, desde o planejamento até o *design*, codificação, testes e finalmente a publicação. E não é só isso – um bom aplicativo precisa ser fácil de usar, rápido e seguro. Agora, como garantir que todas essas etapas sejam concluídas de forma eficiente e sem muitos percalços? É aí que entram as metodologias ágeis. Ao contrário dos métodos tradicionais de desenvolvimento, onde todas as etapas são feitas de forma linear e rígida, as metodologias ágeis permitem que as equipes trabalhem de forma mais flexível e adaptável. Isso significa que o desenvolvimento acontece em ciclos curtos, chamados de *sprints* (utilizado pela equipe *scrum* cujo produto é criado em uma série de interações. Essas interações são chamadas de sprints, que divide projetos complexos em partes menores), onde o produto é constantemente testado e melhorado com base no feedback dos usuários e nas mudanças de mercado. Mas, quais são os principais objetivos que queremos alcançar com essa combinação de desenvolvimento de aplicativos e Metodologias Ágeis?

1. Entrega Rápida e Eficiente: Queremos que os aplicativos cheguem ao mercado o mais rápido possível, sem comprometer a qualidade. As Metodologias Ágeis permitem que os desenvolvedores entreguem versões funcionais do app em ciclos curtos, o que acelera o tempo de lançamento.





2. **Flexibilidade e Adaptação:** No mundo da tecnologia, as coisas mudam rápido. As Metodologias Ágeis oferecem a flexibilidade necessária para ajustar o projeto conforme surgem novas necessidades ou *feedback* dos usuários.
3. **Qualidade e Satisfação do Usuário:** Um dos maiores objetivos é garantir que o aplicativo final seja de alta qualidade e atenda às expectativas dos usuários. Testes constantes e a integração de *feedback* ao longo do processo de desenvolvimento ajudam a alcançar esse objetivo.
4. **Melhoria Contínua:** As Metodologias Ágeis promovem a ideia de melhoria contínua. A cada *sprint*, a equipe analisa o que funcionou bem e o que pode ser melhorado, garantindo que o produto evolua de forma consistente.



Ao combinar o desenvolvimento de aplicativos com as Metodologias Ágeis, as equipes não só criam produtos melhores e mais rapidamente, mas também conseguem se adaptar às mudanças, inovar constantemente e, o mais importante, entregar valor real aos usuários.

O desenvolvimento de aplicativos tornou-se uma das áreas mais dinâmicas e essenciais na era digital atualmente. Com o crescimento de dispositivos móveis e a demanda crescente por soluções que facilitem a vida das pessoas, as empresas e desenvolvedores enfrentam o desafio constante de criar aplicativos inovadores, eficientes e que atendam às expectativas dos usuários.

Para responder a essa demanda, as Metodologias Ágeis surgiram como um conjunto de práticas que mudaram a forma como o *software* é desenvolvido. Diferente dos métodos tradicionais, que seguiam um caminho linear e muitas vezes inflexível, as Metodologias Ágeis oferecem uma abordagem iterativa e incremental. Elas se concentram na colaboração contínua entre as equipes, no *feedback* constante dos usuários e na capacidade de adaptação rápida às mudanças.





No contexto do desenvolvimento de aplicativos, essas metodologias têm se mostrado muito eficazes. Em um cenário onde a rapidez e a qualidade são fundamentais, as Metodologias Ágeis permitem que os desenvolvedores entreguem versões funcionais de seus aplicativos em ciclos curtos, chamados de *sprints*. Isso não só acelera o tempo de lançamento, mas também assegura que o produto final esteja sempre alinhado com as necessidades do mercado e dos usuários.

Assim, estudar Desenvolvimento de Aplicativos e Metodologias Ágeis significa explorar não apenas as técnicas e ferramentas usadas para criar *softwares* de qualidade, mas também compreender uma abordagem de trabalho que prioriza a eficiência, a flexibilidade e a satisfação do usuário. Esse conhecimento é essencial para qualquer profissional que queira se destacar no competitivo mercado de tecnologia atual, onde a capacidade de inovar e se adaptar rapidamente é fundamental para o sucesso.





DESENVOLVIMENTO DE APLICATIVOS E METODOLOGIAS ÁGEIS

O desenvolvimento de aplicativos tem se tornado uma atividade central no campo da tecnologia da informação, em resposta à crescente demanda por soluções digitais que atendam às necessidades cotidianas. Diante disso, as Metodologias Ágeis trazem uma abordagem eficiente para garantir a entrega de produtos de qualidade com prazos mais curtos. As metodologias ágeis oferecem tempos curtos de desenvolvimento, conhecidos como *sprints*, permitindo ajustes contínuos baseados no feedback dos usuários (CAMPOS, 2018).

A utilização das Metodologias Ágeis no desenvolvimento de aplicativos representa uma mudança significativa no modo como a indústria de *software* opera. Segundo Valente (2019), o foco principal dessas metodologias é a adaptabilidade e a resposta rápida às mudanças, ao contrário de um planejamento rígido e inflexível. Isso é especialmente relevante no desenvolvimento de aplicativos, onde a tecnologia avança rapidamente e as expectativas dos usuários mudam constantemente. Aplicativos móveis, por exemplo, exigem atualizações frequentes e melhorias contínuas para se manterem competitivos no mercado (SOUZA; PEREIRA, 2020).



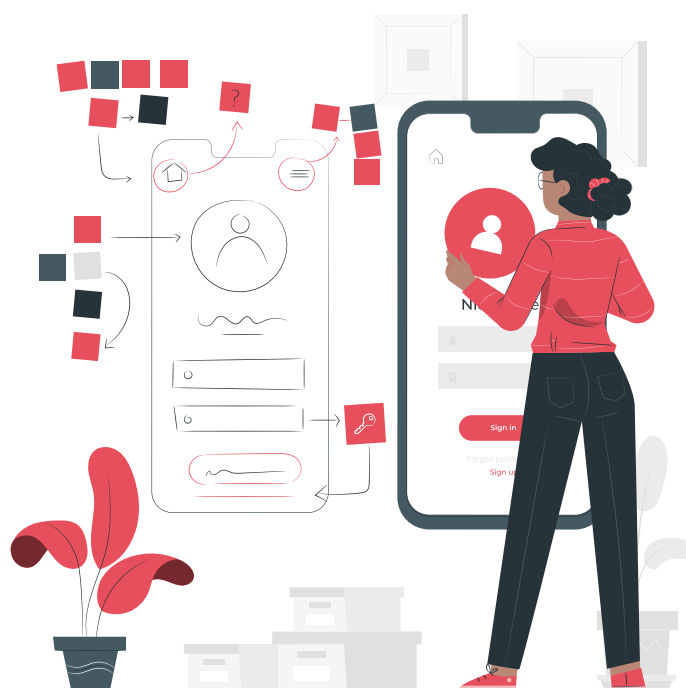


Essas metodologias não apenas melhoram a eficiência do processo de desenvolvimento, mas também promovem uma colaboração mais estreita entre as equipes. Em vez de trabalhar de forma isolada, desenvolvedores, *designers* e outros profissionais participam de reuniões diárias e revisões de *sprint*, garantindo que o projeto avance de acordo com as necessidades reais dos usuários (LIMA, 2017). Esse enfoque colaborativo é fundamental para a criação de aplicativos que ofereçam uma experiência de usuário excepcional.

Além disso, as Metodologias Ágeis facilitam a identificação precoce de problemas e permitem sua rápida resolução, o que reduz o risco de falhas no produto final. Isso é crucial em projetos de desenvolvimento de aplicativos, onde a complexidade técnica e as expectativas dos usuários podem mudar ao longo do tempo. Ao permitir ajustes contínuos e entregas incrementais, essas metodologias garantem que o produto final seja robusto e alinhado com as demandas do mercado (MACHADO; MAIA, 2018).

Portanto, o uso de Metodologias Ágeis no desenvolvimento de aplicativos é uma prática indispensável para qualquer organização que queira se destacar no mercado tecnológico. Essas metodologias não apenas aceleram a entrega de produtos, mas também asseguram que o software desenvolvido seja de alta qualidade, atendendo com precisão às expectativas dos usuários e às necessidades do mercado. Como afirmam Pimentel e Santos (2020), a capacidade de adaptação e a inovação contínua são os pilares do sucesso em um ambiente tecnológico cada vez mais dinâmico e exigente.

Vamos explorar juntos como se dá o desenvolvimento de *software*, desde a concepção de um projeto até a entrega final. Neste e-book, abordaremos o ciclo de vida dos projetos de *software* e as metodologias ágeis, com foco no *Scrum* e no *Kanban*, além de práticas ágeis que podem ser aplicadas no dia a dia da programação. Para isso, usaremos uma linguagem simples e exemplos práticos que facilitarão sua compreensão. Vamos começar?





As metodologias ágeis surgiram em 2001, quando dezessete desenvolvedores se reuniram em uma estação de esqui em Utah, nos EUA, para criar o **Manifesto Ágil**. Esse documento definiu os princípios das metodologias ágeis, enfatizando a flexibilidade, a colaboração, e a entrega contínua de *software*. Essa reunião marcou o início das metodologias ágeis, que foram desenvolvidas como uma resposta à insatisfação com os métodos tradicionais, considerados lentos e inflexíveis.

Ciclo de Vida de Projetos de Software



O ciclo de vida de um projeto de software pode ser comparado ao caminho que percorremos ao construirmos uma casa. Primeiro, é necessário planejar, definir as etapas e, finalmente, executar a construção. No desenvolvimento de *software*, esse processo é dividido em fases, cada uma com sua importância. Segundo Pressman e Maxim (2016), o ciclo de vida do *software* tradicionalmente inclui as seguintes etapas:





1. **Planejamento:** aqui, definimos os objetivos do *software*, requisitos e o escopo do projeto. É como desenhar a planta de uma casa.
 2. **Análise de Requisitos:** nessa fase, detalhamos o que o *software* precisa fazer. Isso envolve conversas com os futuros usuários para entender suas necessidades.
 3. **Projeto:** com os requisitos em mãos, passamos a desenhar como o *software* será estruturado. Pensamos em como as peças se encaixam, como se fosse a estrutura dessa casa.
 4. **Implementação:** é o momento de colocar a mão na massa e começar a programar. Cada linha de código é como um tijolo na construção da casa.
 5. **Testes:** aqui, verificamos se tudo está funcionando como deveria. Testamos se as portas abrem, se as janelas fecham, ou seja, se o *software* faz o que foi planejado.
 6. **Manutenção:** Após a entrega, o *software* precisa ser atualizado e corrigido, assim como uma casa precisa de reparos e melhorias ao longo do tempo.
- Essas etapas são essenciais para garantir que o *software* seja entregue conforme o planejado e atenda às necessidades dos usuários.
- O ciclo de vida de projetos de *software* é fundamental para o sucesso de qualquer desenvolvimento. Ele garante que todas as etapas, desde o planejamento até a manutenção, sejam cuidadosamente planejadas e executadas, resultando em um *software* que atenda às necessidades do usuário, além de ter a possibilidade de ser mantido e de ser realizado **upgrade** ao longo do tempo.



Upgrade - atualização, avanço, melhoramento, melhoria, modernização, progressão, progresso.



Introdução às Metodologias Ágeis: Scrum e Kanban



As metodologias ágeis vieram para mudar a forma como o *software* é desenvolvido, trazendo mais flexibilidade e eficiência ao processo. Ao invés de seguir um caminho linear, como vimos no ciclo de vida tradicional, as metodologias ágeis permitem que as etapas sejam revisadas e ajustadas continuamente.

Scrum

O *Scrum* é uma das metodologias ágeis mais utilizadas e é dividido em ciclos curtos chamados *sprints*, que geralmente duram de uma a quatro semanas. Cada sprint resulta em uma entrega parcial do *software*, que é então revisada e melhorada no próximo ciclo. Conforme vimos na introdução deste e-book e de acordo com o que afirma Schwaber (2004), “Scrum é uma estrutura que permite às equipes resolverem problemas complexos enquanto produzem produtos de maior valor”.

Exemplo prático: imagine que você está desenvolvendo um aplicativo de lista de tarefas. No primeiro *sprint*, você pode focar em criar a funcionalidade básica de adicionar e remover tarefas. No próximo, pode incluir a opção de categorizar as tarefas por prioridade.

Kanban

O Kanban, por outro lado, é uma metodologia visual que usa cartões para representar as tarefas. Os cartões são movidos em um quadro à medida que as tarefas avançam de uma etapa para outra, como “A Fazer”, “Em Progresso” e “Concluído”.





De acordo com Anderson (2010), “Kanban é uma maneira simples e eficaz de visualizar o fluxo de trabalho e identificar gargalos no processo”.

Exemplo prático: Em um projeto de *software*, você pode usar um quadro Kanban para acompanhar o progresso de diferentes partes do sistema, como a interface do usuário, o **backend**, e os testes. À medida que cada tarefa é concluída, o cartão correspondente é movido para a coluna “Concluído”. Recursos oficiais, guias e treinamentos sobre Scrum.



Backend - é a parte do desenvolvimento de software que lida com a lógica do servidor, banco de dados e integração de APIs, processando dados e garantindo o funcionamento da aplicação. Ele é responsável por atender às requisições do frontend e realizar operações em segundo plano.

As metodologias ágeis, como Scrum e Kanban, oferecem uma abordagem mais flexível e eficiente para o desenvolvimento de software. Elas permitem que as equipes se adaptem rapidamente às mudanças e entreguem valor contínuo aos usuários, tornando o processo de desenvolvimento mais dinâmico e centrado no cliente.



Para saber mais sobre Scrum, acesse o site Scrum.org que oferece recursos oficiais, guias e treinamentos sobre Scrum. Disponível em: <https://www.scrum.org/>

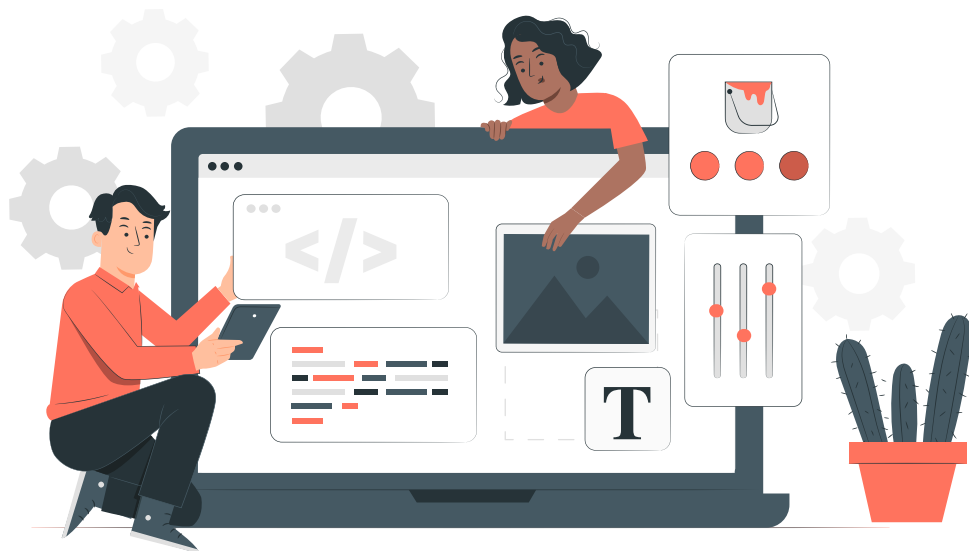
Práticas Ágeis em Programação

Agora que já conhecemos as metodologias Scrum e Kanban, vamos ver como elas podem ser aplicadas diretamente na programação. As práticas ágeis em programação envolvem escrever código de maneira incremental, sempre revisando e melhorando o que já foi feito.

Desenvolvimento Incremental

Uma prática ágil comum é o desenvolvimento incremental, onde o *software* é construído e melhorado em pequenos pedaços. Segundo Beck (2023), “ao desenvolver de forma incremental, podemos entregar valor ao usuário mais rapidamente e adaptar o *software* com base no feedback recebido”.





Exemplo prático: se você pretende programar um sistema de gerenciamento de vendas, pode começar com a funcionalidade de cadastro de produtos. Depois, em incrementos subsequentes, adicione o controle de estoque, relatórios de vendas, e assim por diante.

Refatoração

Outra prática ágil importante é a refatoração, que consiste em melhorar o código existente sem alterar seu comportamento externo. A refatoração ajuda a manter o código limpo e fácil de entender, o que é essencial para a manutenção e evolução do *software*.

Exemplo prático: Se você percebe que um trecho de código está duplicado em várias partes do sistema, pode refatorar o código para movê-lo para uma função única e reutilizável.

A adoção de práticas ágeis em programação, como o desenvolvimento incremental e a refatoração, é essencial para a criação de *software* de alta qualidade. Essas práticas garantem que o código seja continuamente melhorado, facilitando sua manutenção e evolução, ao mesmo tempo em que permite que as equipes entreguem funcionalidades de forma rápida e eficaz.

Conclusão do assunto

Ao longo deste e-book, exploramos o ciclo de vida dos projetos de *software* e as metodologias ágeis Scrum e Kanban. Também vimos como essas práticas podem ser aplicadas na programação para criar *softwares* mais eficientes, flexíveis e alinhados às necessidades dos usuários. O uso dessas metodologias e práticas não só melhora a qualidade do software, como também facilita o trabalho em equipe e a adaptação às mudanças, fatores essenciais para o sucesso no desenvolvimento de *software*.





Resumo

Nesta unidade abordamos o Desenvolvimento de *Software* com foco em metodologias ágeis, explorando o ciclo de vida de projetos, as práticas ágeis, e as metodologias Scrum e Kanban. No primeiro capítulo, discute-se o ciclo de vida tradicional dos projetos de *software*, destacando etapas, como: planejamento, análise de requisitos, projeto, implementação, testes e manutenção. Em seguida, foi introduzido o conceito de metodologias ágeis, explicando como o Scrum e o Kanban oferecem maior flexibilidade e eficiência ao processo de desenvolvimento. Em seguida focamos nas práticas ágeis em programação, incluindo o desenvolvimento incremental e a refatoração, mostrando como essas técnicas podem ser aplicadas para criar *software* de alta qualidade, adaptável e fácil de manter. O e-book enfatiza a importância dessas metodologias e práticas para garantir que o *software* atenda às necessidades dos usuários e seja robusto o suficiente para evoluir com as mudanças tecnológicas.



De acordo com os conceitos abordados no e-book sobre Ciclo de Vida de Projetos de Software, Metodologias Ágeis (Scrum e Kanban), e Práticas Ágeis em Programação, elabore uma síntese com, no mínimo, 20 linhas. A atividade deve abordar os seguintes pontos:

1. Importância do Ciclo de Vida de Projetos de Software: Explique por que é crucial seguir um ciclo de vida bem definido no desenvolvimento de software. Quais são as principais fases e como elas contribuem para o sucesso do projeto?
2. Aplicação das Metodologias Ágeis: Discuta como as metodologias ágeis, como Scrum e Kanban, podem ser aplicadas em projetos de software. Quais são as vantagens dessas metodologias em comparação com o modelo tradicional?
3. Práticas Ágeis em Programação: Descreva a importância de práticas ágeis como o desenvolvimento incremental e a refatoração no processo de programação. Como essas práticas contribuem para a qualidade do código e a satisfação do cliente?
4. Integração dos Conceitos: Mostre como o ciclo de vida do software, as metodologias ágeis e as práticas ágeis em programação podem ser integradas para criar um processo de desenvolvimento eficiente e adaptável.



Resposta esperada:

A importância do Ciclo de Vida de Projetos de Software reside na organização e na sistematização do processo de desenvolvimento, que assegura que todas as etapas, desde o planejamento até a manutenção, sejam cumpridas de forma estruturada. Fases como análise de requisitos, design, implementação e testes são cruciais para garantir que o software final atenda às expectativas dos usuários e seja sustentável a longo prazo.

As metodologias ágeis como Scrum e Kanban proporcionam uma abordagem flexível e centrada no cliente, permitindo que o desenvolvimento ocorra em ciclos curtos com entregas contínuas. Essas metodologias permitem que as equipes respondam rapidamente a mudanças nos requisitos e nas condições do mercado, o que é uma vantagem significativa em relação ao modelo de desenvolvimento tradicional, que é mais rígido.

Práticas ágeis em programação, como o desenvolvimento incremental e a refatoração, são essenciais para manter a qualidade do código e garantir sua evolução ao longo do tempo. O desenvolvimento incremental permite que o software seja construído e aprimorado em pequenos pedaços, enquanto a refatoração assegura que o código permaneça limpo e de fácil manutenção. Integrando o ciclo de vida tradicional, as metodologias ágeis e as práticas ágeis em programação, as equipes de desenvolvimento conseguem criar um processo adaptável e eficiente, que não só garante a entrega de software de alta qualidade, mas também a satisfação contínua do cliente.



REFERÊNCIAS

ANDERSON, D. Kanban: Mudança Evolucionária de Sucesso para seu Negócio de Tecnologia. 1. ed. São Paulo: Casa do Código, 2010.

BECK, K. Desenvolvimento incremental e a entrega de valor. Revista de Engenharia de Software, São Paulo, v. 10, n. 2, p. 45-55, 2023.

CAMPOS, João. **Metodologias Ágeis no desenvolvimento de software: uma abordagem prática**. São Paulo: Editora Inovação, 2018.

LIMA, Carlos. **A colaboração no desenvolvimento de software ágil: impacto e desafios**. Rio de Janeiro: TechPress, 2017.

MACHADO, Ana; MAIA, Ricardo. **Desenvolvimento de software ágil: princípios e práticas**. 2. ed. Porto Alegre: SoftwareHouse, 2018.

PIMENTEL, Fernanda; SANTOS, Bruno. **Inovação e adaptabilidade no desenvolvimento de software ágil**. Recife: Editora Atualiza, 2020.

PRESSMAN, R. S.; MAXIM, B. R. Engenharia de Software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

SCHWABER, K. Agile Project Management with Scrum. 1. ed. Redmond: Microsoft Press, 2004.

SOUZA, André; PEREIRA, Juliana. **Desenvolvimento de aplicativos móveis: desafios e oportunidades**. Belo Horizonte: Editora Digital, 2020.

VALENTE, Lucas. **A evolução das metodologias ágeis no desenvolvimento de software**. Curitiba: TecBooks, 2019.



CURRÍCULO DO PROFESSOR

Elda Nunes de Carvalho

Mestre em Informática pelo Universidade Federal do Amazonas - UFAM, graduada em Tecnologia em Mecatrônica Industrial pelo Instituto Federal do Amazonas - IFAM. especialização em Engenharia de Manutenção Industrial pelo Centro Universitário do Norte - UNINORTE, atualmente sou professora de ensino superior nos cursos de Engenharia da Computação, Ciência da Computação e Análise e Desenvolvimento de Sistemas. Tenho muito interesse pela área de desenvolvimento.





CETAM

CENTRO DE EDUCAÇÃO
TECNOLÓGICA DO AMAZONAS



AMAZONAS

GOVERNO DO ESTADO



CETAM EaD

ESCOLA DE EDUCAÇÃO PROFISSIONAL A DISTÂNCIA