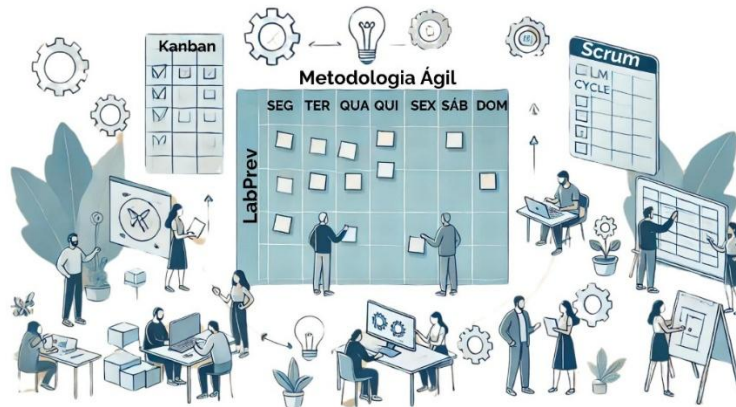


Metodologias ágeis



Fonte: <https://www.gov.br/inss/pt-br/assuntos/laboratorio-de-inovacao-do-inss-labprev/conteudos/metodologias-ageis-como-fazer-mais-sem-desperdicar-tempo>

É fato que a metodologia tradicional descrita no guia de gestão de projetos PMBOK não é a única existente. Com efeito, é uma metodologia que pode conferir ao gestor uma certificação internacional na área.

Contudo, cumpre também apresentar outras metodologias que certamente contribuem para uma eficiente gestão de projetos, sobretudo para aqueles que já possuem algum conhecimento na área de gestão e necessitam de maior agilidade e consequente obtenção de resultados. Nesse contexto, é ainda importante salientar que essas metodologias conhecidas como ágeis, são mais adequadas para projetos de pequeno e médio porte que precisam de maior rapidez em sua conclusão e não necessitam de grandes formalizações documentais.

Evidentemente esse material não tem a pretensão de esgotar o assunto a respeito de metodologias ágeis, mas busca apresentar algumas características e técnicas utilizadas, sobretudo aquelas que são mais conhecidas.

Método SCRUM

Scrum é um framework.

A palavra *framework* sempre foi melhor empregada para indicar estruturas prontas ou conjunto de códigos prontos utilizados para o desenvolvimento de *softwares*, *sites* e aplicativos.

No que se refere especificamente à gestão de projetos, essas estruturas podem ser compreendidas como estruturas baseadas em um conjunto de funções e

componentes pré-definidos com o objetivo de poupar tempo, evitar retrabalhos e garantir maior produtividade e qualidade no desenvolvimento de um projeto.

Essa metodologia foi desenvolvida por *Jeff Sutherland*, que, como piloto de avião, sempre comparou o processo de gestão e finalização de um projeto com a missão de pousar um avião.

Segundo ele, o grande desafio de pousar uma aeronave não pode ser descrito como uma fórmula fixa, pois o piloto precisa fazer ajustes o tempo todo para conseguir manter a aeronave na rota de pouso. Fazendo uma analogia com a gestão de um complexo projeto, a coordenação de um grande número de pessoas e atividades também necessita constantemente da realização de ajustes.

A metodologia SCRUM propõe a fragmentação do projeto em pequenos ciclos de atividades, com frequentes reuniões buscando um rápido alinhamento da equipe e uma maior agilidade nos processos. Nessa metodologia, todas as atividades são acompanhadas de perto e as mudanças de planejamento são implementadas durante todo o tempo.

No Scrum, os problemas mais complexos são decompostos e traduzidos em pequenas tarefas individuais e priorizadas.

Vamos compreender como funciona essa metodologia?

DONO DO PRODUTO E O BACKLOG

Todo projeto começa com o Dono do Produto. Isso não é novidade, não é mesmo?

O dono do produto (*Product Owner*), é a pessoa que melhor representa o usuário final e evidentemente tem autoridade para dizer o que deve e o que não deve ser feito. É ele quem sempre dá a palavra final sobre as características do produto resultante do projeto.

O dono do produto é o encarregado de fazer aquilo que chamamos de *Backlog*. O *Backlog* é uma lista de tarefas a serem executadas, das suas exigências e do que é necessário conter no produto. O *Backlog* deve ser descrito em ordem de prioridade.

É importante compreender que o *Backlog* nunca está completo, pois na medida em que o projeto avança, novas necessidades se apresentarão e, portanto, deverão ser a ele incorporadas. Isso quem faz é sempre o dono do produto. Aqui se apresenta

uma grande diferença em relação ao PMBOK, no qual, uma vez definido o escopo, dificilmente ocorrerão mudanças, enquanto no Scrum isso é comum.

SCRUM MASTER

Nesse contexto, faz-se necessária a apresentação de um outro personagem. Trata-se do Scrum *master*. Esse é o facilitador e potencializador do *Team* (equipe). É ele o responsável por preservar os valores e princípios do *framework* Scrum ajudando a todos os membros da equipe. Nessa metodologia, não há espaço para egos. A liderança é quem serve ao grupo.

O Scrum *Master*, além de buscar potencializar as habilidades de todos os integrantes da equipe, agindo como um coach, ou ainda, como um líder, precisa orientar o dono do produto de maneira que este consiga de fato montar o *Backlog* e mantê-lo atualizado de forma a priorizar os requisitos.

SPRINT

Um outro conceito importante é o *Sprint*. Trata-se do período dentro no qual a equipe executa um conjunto de atividades do *Backlog*. Esse período é definido de acordo com as necessidades da equipe. Um *Sprint* bastante comum nessa metodologia é de duas semanas.

Ao planejar o *Sprint*, deve-se escolher as tarefas do *Backlog* que devem ser concluídas. O Sprint tem um tempo limite e deve ser sempre inferior a um mês. A equipe deverá escolher quais tarefas serão incluídas e quem será o responsável por elas. Evidentemente que para se escolher o conjunto de tarefas que comporão o Sprint é necessário obedecer a um raciocínio lógico no que se refere à sequência e interdependência das atividades.

No que se refere ao Sprint temos:

Sprint Backlog: Conjunto de itens do Product Backlog que a equipe se compromete a entregar durante uma Sprint (período de trabalho).

Sprint Planning: Reunião inicial em cada Sprint para planejar, o que e como será desenvolvido a demanda solicitada.

Sprint Review: Reunião no final da Sprint para revisar o trabalho e avaliar o resultado.

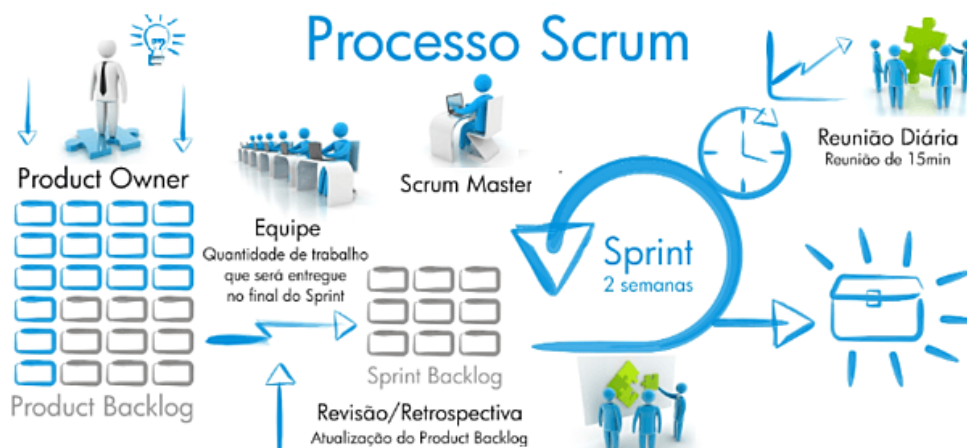
Sprint Retrospective: Reunião para correção, onde deverá ser avaliado o processo e identificar melhorias.

Principais pontos relacionados à metodologia SCRUM:

Daily Scrum

Na metodologia Scrum, as reuniões ocorrem diariamente. Trata-se do Scrum diário (*Daily Scrum*). É imprescindível que essas reuniões não durem mais do que 15 minutos. Elas servem para posicionar todos os integrantes da equipe sobre o andamento das atividades, em que pé estão, se estão encontrando algum tipo de dificuldade e realizar os ajustes que se fizerem necessários. O *Daily Scrum* funciona como uma retrospectiva, onde o trabalho é analisado e servirá para o planejamento do próximo *Sprint*.

Cumprе acrescentar que essa característica do Scrum está sendo largamente utilizada por equipes que trabalham remotamente (online). Está permitindo também a contratação de profissionais residentes em locais distantes do local de implementação do projeto, agregando maior qualificação e competência à equipe.



Fonte: <https://zappts.com.br/scrum-case-ganhos-na/>

Visualização do trabalho utilizando o quadro *Kanban Scrum*

Para que o acompanhamento diário das atividades seja realizado de forma eficaz, uma das ferramentas utilizadas é o *Kanban Scrum*. Trata-se de uma simples tabela na qual pode-se visualizar o status de cada uma das tarefas.

A executar	Em execução	Teste	Enviado	Executado
Tarefa 8	Tarefa 6	Tarefa 5	Tarefa 4	Tarefa 1
Tarefa 9	Tarefa 7			Tarefa 2
Tarefa 10				Tarefa 3

Equipes enxutas

Segundo *Jeff Bezos*, uma equipe tem o tamanho ideal quando pode ser alimentada com duas pizzas (*two pizzas team*).

De acordo com estudos realizados comprovou-se que uma equipe pequena trabalha melhor do que grandes times. A dica é que a equipe não tenha mais do que 10 pessoas.

Sem títulos

A equipe trabalha melhor quando não há títulos para cada função, de maneira que o entendimento de cada um sobre o seu papel não fique limitado dentro da equipe.

Prioridades

As atividades devem ser priorizadas de maneira que se faça uma coisa de cada vez.

Sem heroísmo

Uma equipe não deve ser sacrificada para atingir os objetivos do projeto. Quando o time está sobrecarregado é porque há algo errado, mesmo que esse excesso de esforço seja algo pontual.

Por fim, a metodologia Scrum pressupõe acompanhar a equipe de perto, motivando-a constantemente e mantendo um registro visual de todas as atividades.

KANBAN

Embora seja listada por muitos como uma metodologia ágil, podemos afirmar, sem sombra de dúvidas, que ela está muito mais para uma ferramenta de controle, acompanhamento e priorização de atividades, que pode ser utilizada na gestão de projetos e que já é, há muito tempo, experimentada nos processos de produção.

A palavra Kanban é de origem japonesa e seu significado é “cartão” ou “sinalização”.

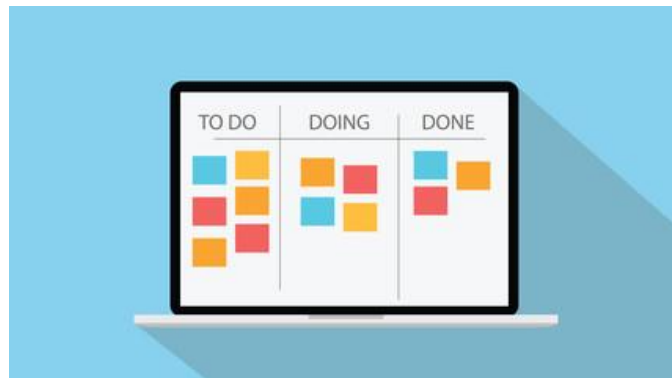
Em princípio, o método Kanban é um método visual. Trata-se de uma maneira de se ter um controle visual de todas as tarefas.

É um sistema bastante simples surgido na Toyota em meados da década de 1950.

Por meio de um quadro exposto em local visível a todos, todas as tarefas são sinalizadas por meio de cartões. Esses cartões podem ser de papel, post-it ou de plástico e obedecem basicamente às cores vermelho, verde e amarelo. Por meio delas é possível identificar rapidamente qual é o seu *status*.

VERMELHO	-	(a executar)
AMARELO	-	(em andamento)
VERDE	-	(Finalizada)

Evidentemente cada empresa e ou projeto, poderá acrescentar outras cores às suas representações, de acordo com sua necessidade.



Fonte:

https://stock.adobe.com/pt/search?as_audience=srp&as_campaign=Freepik&get_facets=1&order=relevance&safe_search=1&search_page=1&as_content=api&k=kanban&tduid=c0200c8c300bbffc341c6da5e733c1e4&as_channel=affiliate&as_campaign=redirect&as_source=arvato&asset_id=215326332

Cabe ao gestor do projeto definir a quantidade de cartões que serão trabalhados por meio de uma criteriosa análise de recursos disponíveis e tempo de execução previstos para cada atividade.

Vantagens:

- Permite analisar os dados com maior rapidez e eficiência.

- Agiliza O processamento de informações porque se baseia simplesmente na observação de cores em um quadro.

- Adapta-se às rotinas e estruturas pré-existentes.

- Facilita a interpretação, a tomada de decisão e a análise dos prazos para cada tarefa, permitindo à equipe discutir sobre os problemas e dificuldades encontrados e, conseqüentemente, apresentar possíveis soluções para tomada decisão.

Cumprir dizer que a integração dos diversos setores de um projeto e ou empresa, depende de que todos os dados sejam acessíveis e facilmente visualizados por todos. Embora seja de fácil entendimento e aplicação, o Kanban exige que todos os integrantes da equipe sejam comprometidos e possuam uma contínua participação. Os cartões precisam estar devidamente organizados e atualizados para que não percam a sua funcionalidade. Portanto, pode-se perceber que o Kanban é muito mais do que um simples quadro com cartões, é, antes de tudo, uma filosofia de trabalho e que, como tal, depende do engajamento de todos.

Assim como o Scrum e como outras metodologias ágeis, a preocupação é com uma tarefa e ou funcionalidade de cada vez. Por essa razão, fica fácil implementar alterações e acrescentar novas tarefas sempre que estas se fizerem necessárias.

Efetivamente faz-se essencial a criação de equipes pequenas que estejam focadas em seus pacotes de trabalho. Cada pacote de trabalho deve ser planejado, possuir um tempo estimado para sua conclusão e definir as responsabilidades de cada integrante da equipe com relação às atividades previstas. Esse será o cartão que circulará pelo quadro até o momento de sua conclusão.

No próprio cartão, deverão ser registradas a data e a hora da sua conclusão. Assim, será possível realizar uma comparação entre o tempo previsto e o realizado. Isso certamente contribuirá para compor os dados históricos do projeto e conseqüente busca por melhorias em projetos futuros.

Metodologia LEAN

A metodologia Ágil Lean é uma filosofia de trabalho que combina os princípios da metodologia ágil como o Scrum ou o Kanban, com os conceitos do Lean Manufacturing. Seu principal foco é a eliminação de desperdícios, otimização de processos e entrega contínua de valor ao cliente.

A metodologia ágil Lean busca maximizar a eficiência e a produtividade das equipes, entregando um produto de qualidade sem sobrecarregar os recursos, com o mínimo de desperdício possível.

Nessa filosofia, assim como no Scrum, o dia de trabalho sempre começa com uma reunião rápida para revisar as tarefas do dia, visualizar o progresso em um quadro (Kanban) e garantir que todos estejam alinhados, buscando realizar atividades que agregam valor, evitam tarefas desnecessárias, e, ao final de cada ciclo de trabalho, fazem uma retrospectiva para identificar pontos de melhoria.

Seguem alguns pontos que destacam a filosofia Lean:

1 - Foco na Entrega de Valor

A equipe deve revisar suas atividades, priorizar as que agregam mais valor e eliminar as que não agregam. Isso implica em desenvolver soluções para necessidades reais do cliente, evitando desperdícios de tempo e recursos.

2 - Eliminação de Desperdícios

Conforme já citado, um dos principais pontos do Lean é a eliminação de desperdícios. Isso faz com que seja prática diária a identificação e remoção de tudo o que não agrega valor ao processo ou ao produto.

3 - Fluxo Contínuo

O objetivo é sempre garantir que o fluxo de trabalho seja contínuo e sem interrupções. Havendo desvios e contratempos, o líder do projeto deve analisar e, se necessário, encaminhar para outros integrantes da equipe. Na prática, isso pode ser implementado através do Kanban, onde o trabalho é visualizado em um quadro.

4 - Melhoria Contínua (Kaizen)

O conceito de Kaizen significa "melhoria contínua" e deve ser exercitado diariamente. Toda a equipe deve comprometer-se e compartilhar as boas práticas, buscando sempre maneiras de incentivar a melhoria de todas as atividades, reduzindo desperdícios e melhorando a qualidade dos resultados.

5 - Tomada de Decisão Baseada em Dados

O Lean orienta que as decisões devem ser baseadas em dados e informações precisas e que não haja dúvidas. Isso significa que a equipe deve cuidar da veracidade das informações para tomar decisões. Havendo dúvidas, deve se reunir e buscar novos dados.

6. Empoderamento das Equipes

No Lean, as equipes são estimuladas a tomar decisões rápidas e a resolver problemas. Isso tende a aumentar a responsabilidade e o engajamento, acelerando a entrega dos resultados.

7. Planejamento Visual

O planejamento visual é um recurso extremamente produtivo, onde todas as atividades e seus respectivos status devem estar disponíveis a todos da equipe. Isso pode ser feito com o uso de ferramentas como quadros Kanban.

8. Feedback e Adaptação Rápida

A Lean estimula e valoriza o feedback rápido e contínuo, buscando que os processos possam ser ajustados conforme necessário. O trabalho não deve ser engessado.

- ✓ Crystal Clear;
- ✓ Método SMART
- ✓ Feature-Driven Development;
- ✓ Extreme Programming Management;
- ✓ Adaptive Software Development;
- ✓ Metodologia LEAN
- ✓ Design Thinking

O que essas metodologias têm em comum? Todas elas são utilizadas na construção e desenvolvimento de softwares.

Portanto, em razão de todas servirem basicamente ao mesmo propósito, o que de certa forma restringe a aplicabilidade a uma única área de atuação, iremos abordá-las de uma forma mais genérica. Vamos lá?

Crystal Clear

Na verdade, não há uma metodologia denominada *Crystal Clear*. Cumpre esclarecer que esse termo consiste em uma família de diferentes tipos de métodos utilizados na gestão ágil de projetos. Trata-se da família *Crystal*. Alguns elementos são centrais e comum a todos os métodos.

Segundo o seu criador Alistair Cockburn, a família *Crystal* se baseia na gestão de pessoas, pois os integrantes de uma equipe possuem habilidades e talentos diferentes que são colocados em prática durante o desenvolvimento e implementação de um projeto.

A ideia é utilizar diferentes tipos de metodologia para diferentes tipos de projetos.

O que os diferencia na verdade são algumas práticas específicas. Aqui o que se busca é uma adequação ao tamanho do projeto e aos riscos envolvidos na sua execução.

A família (*Crystal*) é dividida em cores de acordo com o tamanho da equipe. Sendo:

Crystal Clear: equipes pequenas de até 6 pessoas, com um foco na simplicidade e menor grau de formalidade.

Crystal Yellow: equipes de até 20 pessoas, mantém a flexibilidade, mas com uma maior estrutura, em condições de gerenciar o aumento da complexidade.

Crystal Orange: equipes de até 50 pessoas, incluem um maior número de processos e ferramentas de gestão para garantir a colaboração e a comunicação eficaz.

Crystal Red: equipes de até 200 pessoas, o que leva a uma necessidade de maior formalização e controle dos processos.

Crystal Maroon: equipes muito grandes, onde o controle e a documentação são mais rigorosos. Em razão disso, o aumento da complexidade conduz à uma necessidade de coordenação mais detalhada.

Cada método *Crystal* é caracterizado por uma cor, de acordo com o número de envolvidos.

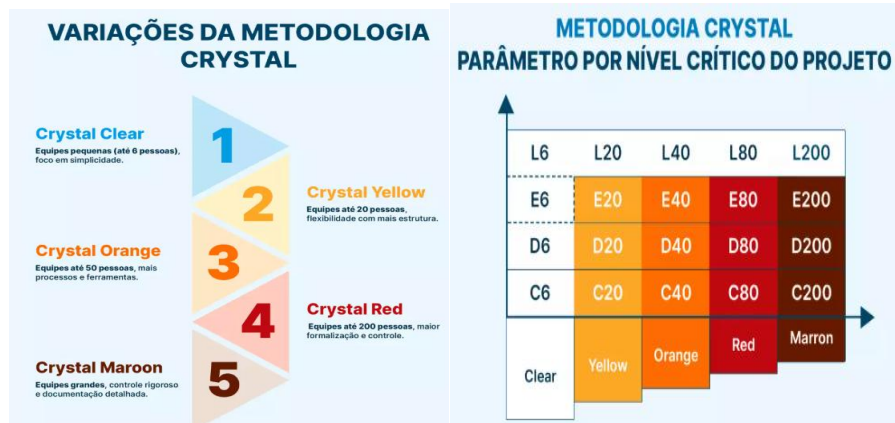
Além das cores, são utilizadas também as letras para indicar a criticidade do projeto.

C → Confort (Conforto) → Nível de criticidade que pode ocasionar desconforto ao cliente.

D → Discretionary Money (Dinheiro disponível) → Nível de criticidade que pode ocasionar pequenas perdas monetárias.

E → Essencial Money (Dinheiro essencial) → Nível de criticidade que pode ocasionar grandes perdas monetárias.

L → Life (vida) → Nível de criticidade que pode ocasionar perda de vidas humanas.



Fonte: <https://blog.runrun.it/metodologia-crystal/>

Existem algumas características que são comuns entre as práticas, as quais podemos destacar:

- ✓ Entregas frequentes: Os donos dos projetos esperam a apresentação de resultados a cada dois meses;
- ✓ Inspeções realizadas a cada incremento;
- ✓ Comunicação constante: a ideia é que todos os integrantes da equipe realizem suas atividades no mesmo ambiente de trabalho. Nos projetos maiores também se busca essa prática na medida do possível;
- ✓ Segurança: aqui temos duas formas de segurança sendo a primeira relacionada à segurança dos membros da equipe que precisam se sentir à vontade, sem medo de represálias, para desenvolverem bem suas atividades. A outra forma de segurança se refere aos usuários do produto resultante do projeto. Nesse caso, poderíamos comparar um projeto de uma nova aeronave e um projeto de um novo modelo de máquina de lavar roupas. Evidentemente que a criticidade relativa à segurança do usuário é totalmente desproporcional;
- ✓ Foco: cada integrante da equipe deve focar em dois ou três itens com prioridade;
- ✓ Acesso para usuários: A equipe do projeto precisa ter acesso aos usuários do sistema a ser desenvolvido;
- ✓ Testes automáticos e Integração: permissão para a realização de testes das funcionalidades do produto resultante integrando seus diversos componentes;
- ✓ Feedback constante: todos os integrantes da equipe se reúnem com bastante regularidade. Isso ocorre também com todas as partes interessadas.

Método SMART

O conceito SMART foi introduzido em 1981 por George T. Doran, um gerente e consultor. Tal metodologia busca concentrar esforços no atingimento de metas focando em entregas rápidas e incrementais.

Em um projeto, as equipes geralmente trabalham com sprints por atividades, que são ciclos curtos de desenvolvimento (geralmente entre 1 e 4 semanas). Durante o planejamento do sprint, as metas SMART podem ser usadas para garantir que as entregas sejam claras e objetivas, mensuráveis e relevantes para o progresso do projeto. Isso ajuda a garantir que o time esteja focado em objetivos específicos, mensuráveis e dentro do prazo.

A metodologia ágil de metas SMART baseia-se em 5 princípios conforme as siglas de seu nome, sendo:

1 - S - Específica (Specific) - A meta deve ser clara e bem definida. No contexto ágil, isso significa que a equipe deve entender o que está sendo esperado e qual é o objetivo final.

2 - M - Mensurável (Measurable) - A meta deve ser quantificável, ou seja, deve ser possível criar um indicador e medir o progresso.

3 - A - Atingível (Achievable) - A meta deve ser realista e alcançável, levando em consideração os recursos, tempo e habilidades da equipe.

4 - R - Relevante (Relevant) - A meta deve ser relevante para o projeto ou para os objetivos do time e da organização. No contexto ágil, isso significa que a meta deve ter impacto no progresso do projeto ou atender a uma necessidade importante do seu desenvolvimento.

5. T - Temporal (Time-bound) - A meta deve ter um prazo bem definido para ser alcançada. No ágil, os prazos são geralmente definidos em ciclos curtos, como sprints. Em resumo, é uma metodologia simples, que permite que se tenha benefícios como:

Clareza: Onde a equipe sabe exatamente o que precisa ser feito.

Foco: Facilitando o foco nas prioridades.

Medição: Ajuda a desenvolver indicadores e medir o sucesso das etapas do projeto.

Alinhamento: Mantém todos alinhados com os objetivos do time e da organização.



Fonte: <https://blog.aevo.com.br/metodologia-agil/>

Dynamic Systems Development Method (DSDM):

O Dynamic Systems Development Method (DSDM) é uma metodologia ágil que foca na entrega de soluções de software de forma rápida e eficiente, utilizando um processo iterativo e incremental. Sua principal característica é a ênfase no envolvimento do cliente e no controle de custos e prazos, sem comprometer a qualidade.

Uma outra característica do DSDM é a entrega frequente de software funcional e a colaboração constante entre todas as partes envolvidas. Ele é estruturado em cinco fases principais: **Pré-projeto, Refinamento e Design, Construção, Entrega e Pós-projeto.**

Durante essas fases, o desenvolvimento é dividido em pequenos testes, com entregas parciais, permitindo ajustes contínuos. O DSDM também destaca a importância de requisitos bem definidos, priorização de funcionalidades essenciais e flexibilidade para mudanças.

Além disso, ele se preocupa com a qualidade contínua, garantindo que o software esteja sempre em conformidade com as expectativas do cliente e com os padrões técnicos.

Agile Unified Process (AUP):

Antes de entendermos a metodologia Agile Unified Process (AUP), é preciso saber que ela é uma versão simplificada do tradicional Rational Unified Process (RUP), que também é uma metodologia de desenvolvimento de software orientada a fases, estruturada em quatro etapas principais: Iniciação, Elaboração, Construção e Transição. Ele adota uma abordagem iterativa e incremental, permitindo entregas frequentes de software.

O **Agile Unified Process (AUP)** busca equilibrar a estrutura do RUP com os princípios ágeis, como entregas rápidas, flexibilidade e colaboração constante com o cliente.

Assim, o AUP foi estruturado para ser aplicado nas mesmas quatro fases descritas na RUP. Cada fase possui um conjunto de atividades específicas e é composta por múltiplas iterações, permitindo ajustes conforme o desenvolvimento avança.

O AUP é mais leve e foca na entrega de projetos de forma funcional e em ciclos curtos. Ele também promove a integração contínua, com ênfase na qualidade e testes regulares, além de priorizar a comunicação eficiente entre as equipes.

O AUP se adapta bem a projetos de pequeno a médio porte, buscando a flexibilidade das metodologias ágeis, sem perder a disciplina de processos bem definidos.

Feature-Driven Development (FDD):

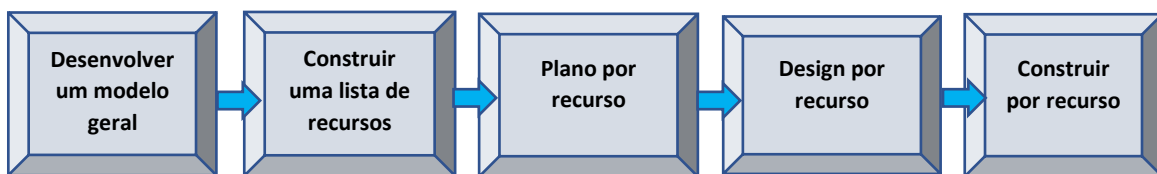
A metodologia Feature-Driven Development (FDD) é uma abordagem ágil focada no desenvolvimento de funcionalidades específicas de um sistema. Ela se baseia na construção do software a partir de features (funcionalidades) que são pequenas e bem definidas, entregues de forma incremental.

O processo começa com a definição do modelo global do sistema, que é uma visão geral do que o software fará. Em seguida, o sistema é dividido em diversas features, que são priorizadas e descritas de forma clara e objetiva. Cada feature geralmente leva de 2 a 10 dias para ser desenvolvida.

Esse modelo promove uma entrega rápida e focada em resultados tangíveis, permitindo o acompanhamento contínuo e o ajuste de prioridades conforme o progresso do projeto.

Ela surgiu em Singapura entre 1997 e 1999. Foi criada por *Jeff de Luca* que se inspirou em um outro método conhecido como Coad (criado por *Peter Coad*). Sua tradução é “Desenvolvimento baseado em Recursos”. É comum sua utilização combinada com o SCRUM e outras metodologias ágeis, pois enquanto outras focam no gerenciamento das tarefas, o FDD busca o desenvolvimento das funções. Aliás, cumpre dizer que ela é bem semelhante ao SCRUM, sobretudo no que se refere à estrutura da equipe.

O FDD possui cinco processos a saber:



- 1- Desenvolve-se um modelo abrangente, normalmente com pouco conteúdo, mas que exprima de modo geral todas as características que o produto do projeto deve possuir;
- 2- Em seguida, com base nas informações levantadas, faz-se uma lista de funcionalidades (recursos) agrupados em conjunto e áreas temáticas;
- 3- Um plano de desenvolvimento para cada pequena funcionalidade (features), que possui valor para o cliente;
- 4- Cada recurso deve ser desenvolvido por uma pequena equipe, onde cada um de seus membros contribui para o seu design e desenvolvimento adicionando mais conteúdo ao modelo;
- 5- O desenvolvimento do recurso propriamente dito deve ser concluído num prazo máximo de duas semanas e avaliado pelo cliente.

Extreme Programming Management

O *Extreme Programming Management*, comumente conhecido como XP, é uma metodologia ágil amplamente utilizada, como já dissemos, nos projetos referentes ao desenvolvimento de *softwares* e tem como ponto de partida a satisfação do cliente. Assim como o SCRUM, trata-se de um *framework* que busca responder às solicitações

dos clientes de forma cada vez mais rápida. Aliás, é baseada em três pilares: agilidade no desenvolvimento, economia de recursos e qualidade do produto.

As equipes são compostas por duplas, que realizam testes automatizados, permitindo que se tenha uma integração contínua e constante.

Essa metodologia valoriza muito a colaboração entre desenvolvedores e clientes, além de promover ciclos curtos de feedback.

Na prática, o método busca proporcionar uma maneira de lidar com as mudanças frequentes nos requisitos e de garantir que a qualidade do software seja alta desde o início.

A metodologia XP se baseia em alguns pilares que auxiliam em sua prática:

1 - Desenvolvimento Iterativo e Incremental

O XP adota um ciclo de desenvolvimento ágil, em que o trabalho é dividido em pequenas iterações (geralmente de uma a três semanas). Durante cada iteração, o time entrega funcionalidades que agregam valor imediato ao cliente.

2 - Programação em Par (Pair Programming)

Um dos pilares do XP é o trabalho em duplas, onde dois desenvolvedores trabalham juntos no mesmo computador. Um escreve o código (chamado de "driver") enquanto o outro revisa e oferece sugestões (chamado de "observer" ou "navigator"). Isso melhora a qualidade do código, reduz erros e acelera o processo de desenvolvimento.

3 - Desenvolvimento Orientado a Testes (TDD - Test-Driven Development)

O XP utiliza o TDD, onde os testes e verificações são escritos e pensados antes mesmo do código, esta é uma forma de se pensar previamente em controles.

4 - Integração Contínua

No XP os testes e integrações são contínuos, ou seja, o código que está sendo desenvolvido é frequentemente integrado ao sistema ou projeto principal várias vezes ao dia. Isso permite detectar problemas rapidamente, evitando que conflitos e bugs se acumulem.

5 - Feedback Contínuo

O método valoriza o feedback contínuo, seja dos testes, dos clientes ou dos próprios desenvolvedores. Deve-se realizar reuniões diárias (ou daily stand-ups), e documentar todos os trabalhos, garantindo que todos os envolvidos estejam alinhados com o progresso do projeto.

6 - Propriedade Coletiva do Código

Apesar do trabalho em duplas, qualquer outra pessoa pode ter acesso e sugerir alterações. Isso incentiva a colaboração, evitando que apenas algumas pessoas tenham conhecimento sobre o projeto.

7 - Simples é Melhor

As premissas do XP, sugere que um software deve ser o mais simples possível para atender aos requisitos, orientando a não adicionar complexidades desnecessárias.

8 - Trabalho Sustentável

O XP valoriza um trabalho sustentável, sem sobrecargas, evitando o burnout das equipes. Isso significa que todos devem trabalhar em um ritmo constante, porém, sem a pressão de prazos impossíveis.

9 - Participação do Cliente

No XP, o cliente deve estar ativamente envolvido durante o processo de desenvolvimento. Deve estar disponível para conhecer as etapas, fornecer feedback contínuo, e contribuir com informações sobre prioridades e requisitos.

Nessa metodologia, destaca-se a ênfase no trabalho de equipe cujos gestores, desenvolvedores e consumidores são peças de igual importância na busca pela solução do problema.

Os feedbacks do cliente são constantes. Eles devem ocorrer a cada entrega que, uma vez testada, deve ser respondida rapidamente de maneira que as modificações necessárias sejam implementadas o quanto antes. Os ciclos curtos diminuem o grau de incertezas e riscos.

Adaptive Software Development

Assim como outros, também é uma técnica para o desenvolvimento de *softwares*. Nessa metodologia, o ciclo de vida de um projeto é composto por três fases a saber:

- 1- Especulação: Aqui a especulação corresponde ao sentido de indagar e pesquisar, questionando as principais causas de uma situação problema. Nesse sentido, poderia tranquilamente ser substituída pelo termo “planejamento”. Contudo, como o grau de incertezas é bastante grande, as mudanças e ajustes certamente ocorrerão. Por essa razão é que se utiliza o termo “especulação”.
- 2- Colaboração: O levantamento das especificações do software e das necessidades que ele deve corresponder. Portanto, necessita da colaboração de todos os envolvidos. Nesse cenário, a comunicação se apresenta com um papel de destaque. A confiança e a transparência contribuem para o desenvolvimento dos trabalhos.
- 3- Aprendizado: A compilação das lições aprendidas ao longo do processo resulta em melhorias futuras.

A tônica dessa metodologia é a possibilidade de modificações ao longo do processo, ou seja, fazendo jus ao nome, parte-se do princípio que é melhor realizar adaptações necessárias do que controlar. Seguindo esta lógica, as avaliações são frequentes de maneira a indicar a necessidade de modificações.

Referências:

<https://artia.com/blog/sistema-kanban-aliado-na-gestao-de-projetos/#:~:text=De%20origem%20japonesa%2C%20o%20termo,quais%20essa%20atividade%20ir%C3%A1%20passar>

<https://blog.mastertech.com.br/negocios/extreme-programming-metodologia-agil-para-o-desenvolvimento-de-software/>

<https://blog.trello.com/br/scrum-metodologia-agil>

<https://leandromtr.com/metodologia-agil-crystal/>

<https://leandromtr.com/metodologia-agil-fdd/>

<https://robsoncamargo.com.br/blog/Extreme-Programming>

<https://www.treasy.com.br/blog/scrum/>

<https://www.treinaweb.com.br/blog/metodologia-crystal-o-que-e-e-para-que-serve>

<https://www.voitto.com.br/blog/artigo/scrum-master>