

METODOLOGIAS ÁGEIS

Conceitos Básicos

INCIDÊNCIA EM PROVA: ALTA



Em meados de 2001, 17 especialistas proeminentes da área de desenvolvimento de software se reuniram em um *resort* em Utah (foto acima) para conversar, esquiar, **discutir e encontrar um terreno comum para suas ideias sobre métodos de desenvolvimento de software**. Essa galera pegou uma mesa, se sentaram, tomaram umas cervejas e começaram a desabafar sobre seus projetos de desenvolvimento de software que estavam falhando por diversos motivos.



Os 17 Engenheiros de Software

Foi quando um deles levantou a mão e **disse que usou o Modelo em Cascata e o projeto estourou o orçamento**; o outro disse que isso também já aconteceu com ele, mas recentemente um projeto falhou porque estourou o prazo; aí outro se compadeceu e disse que os projetos dele viviam falhando porque ele não conseguia construir todo o escopo que foi pedido pelos usuários do sistema. E assim foi...

Eles foram compartilhando suas experiências ruins com o uso das metodologias tradicionais, mas depois cada um desses caras foi dizendo: “*para remediar isso, agora eu uso iterações*”; aí o outro disse que não usa mais tanta documentação como antigamente; aí o outro levantou a mão e falou que não faz mais tanto planejamento; e assim por diante. Então, no decorrer da reunião, foi sendo criado um consenso entre os participantes.



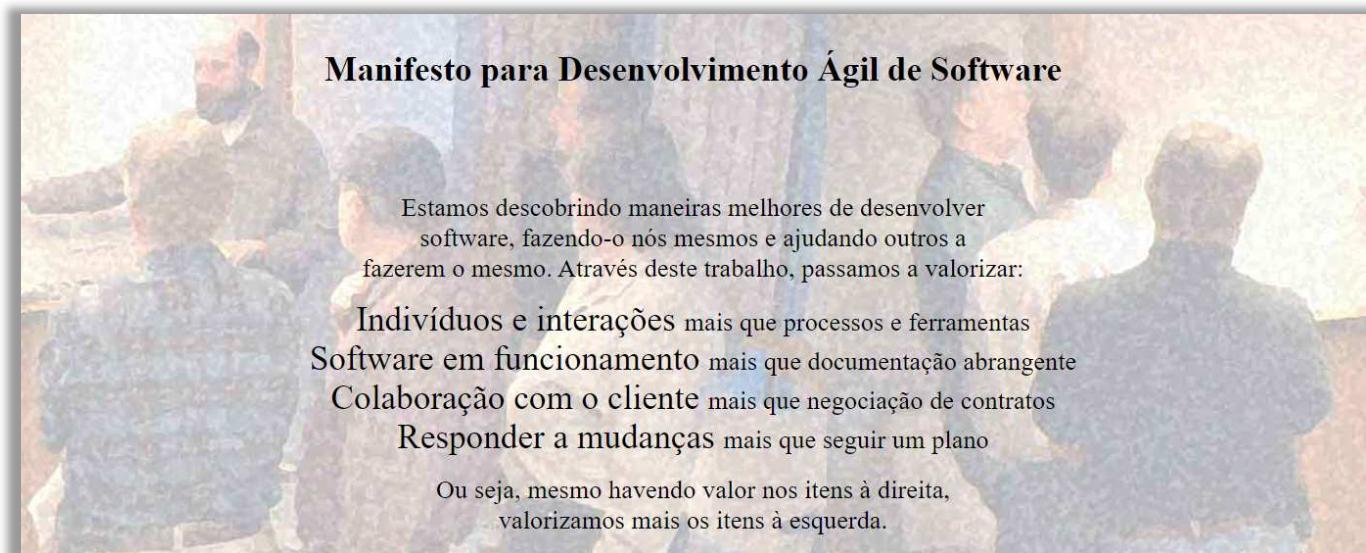
Foi aí que alguns acharam que era o momento de formalizar e elevar aquela reunião em um patamar maior. **Eles decidiram escrever um documento que serviria de grito de guerra contra os processos tradicionais de desenvolvimento de software que vigoravam naquela época.** Para isso, eles pensaram: “Nós precisamos de um nome que expresse bem o significado dessa reunião e das nossas ideias comuns”.

Na discussão, eles decidiram que a palavra “leve” não expressava tão bem o que eles queriam dizer e decidiram trocar pela palavra “ágil”, que captava melhor a abordagem que eles estavam propondo. Em um segundo momento, eles começaram a escrever um **documento bem pequeno, bem objetivo, bem claro e que conteria as crenças, valores e princípios** daqueles dezessete engenheiros de software.

Foi então que surgiu o documento chamado **Manifesto Ágil Para Desenvolvimento De Software**, que definia bem o que era ágil, o que não era ágil e o que essas pessoas defendiam. Além disso, eles passaram a se autodenominar como Aliança Ágil, que era um grupo de pensadores independentes sobre desenvolvimento de software e muitas vezes concorrentes entre si, mas que concordaram em um documento chamado Manifesto Ágil.

Isso se tornou uma organização sem fins lucrativos que procura promover conhecimento e discussões sobre os vários métodos ágeis que existem hoje em dia. A partir daí, galera... esses caras – que eram os líderes do movimento ágil – começaram a escrever artigos, fazer palestras e disseminar esse novo paradigma. Como vocês sabem, **esse negócio explodiu e hoje a imensa maioria dos projetos de software são feitos utilizando metodologias ágeis.**

Bem, para aqueles que não conhecem, nós trazemos a seguir a imagem original do próprio Manifesto Ágil com seus fundamentos. Vejamos...



Notem que a coluna da esquerda representa os anseios das metodologias ágeis, enquanto a coluna da direita representa o que as metodologias tradicionais costumavam executar. **Agora prestem**



atenção: o manifesto ágil afirma que, mesmo havendo valor nos itens à direita, valorizam-se mais os itens à esquerda. Uma pegadinha comum de prova é dizer que os métodos ágeis não possuem documentação. *Isso está correto?*

Não, isso evidentemente está errado! O manifesto ágil preconiza que se valorize mais software em funcionamento do que documentação abrangente, logo isso não significa que não tenha documentação. **E isso serve para os outros três fundamentos, isto é, os itens da direita tem o seu valor, por outro lado se valoriza mais os itens da esquerda. Tudo certo até aqui?** Agora vamos detalhar um pouco mais...

POR QUE VALORIZAR MAIS INDIVÍDUOS E SUAS INTERAÇÕES DO QUE PROCESSOS E FERRAMENTAS?

Porque, em última instância, quem gera produtos e serviços são os indivíduos, que possuem características únicas como talento e habilidade. Pessoal, programar é uma atividade humana e, como tal, depende de questões humanas para que obtenha sucesso. Jim Highsmith, um dos signatários do manifesto ágil, afirma que as habilidades, as personalidades e as peculiaridades de cada indivíduo são críticas para o sucesso dos projetos.

Ele diz também que pessoas muitas vezes são desorganizadas e difíceis de entender, por outro lado elas também são inovadoras, criativas, apaixonadas, entre outros. **E quanto às ferramentas e aos processos, professor? Galera, ambos são importantíssimos para guiar e apoiar o desenvolvimento, mas é a capacidade e o conhecimento dos indivíduos que ajudam a tomar decisões críticas no projeto.**

Dessa forma, basta eu ensinar um conjunto de processos para a minha equipe, assim como um conjunto de ferramentas para garantir que a equipe criará bons softwares? Claro que não! **Uma equipe possui características intrínsecas à personalidade, habilidades e capacidades de cada um dos seus integrantes e isso deve ser considerado e valorizado na construção de um software.** Entendido, pessoal? Seguindo...

POR QUE VALORIZAR MAIS SOFTWARE EM FUNCIONAMENTO DO QUE DOCUMENTAÇÃO ABRANGENTE?

Porque o que gera valor para o cliente é o resultado que você entrega e, não, a documentação em si. Respondam-me uma pergunta: *quando você compra um carro, você olha o motor, o design, o painel, o interior ou você sai correndo loucamente para ler o manual do carro e outras documentações?* Imagino que vocês tenham respondido a primeira opção! Dito isso, concluímos que o software em funcionamento é o único indicador do que, de fato, a equipe construiu.

Claro, não se exclui a necessidade de documentação, que é bastante útil para o desenvolvimento, mas é recomendável produzir somente a documentação necessária e suficiente para a realização do trabalho em si. **Nada de burocratizar demais e construir trezentas páginas de documentação com quatrocentos diagramas diferentes para representar o software.** Tudo certo? Eu vou repetir, porque esse assunto cai bastante em prova!



No ágil, documentação é descartável? Não, ela é útil para ajudar a comunicação e a colaboração dos integrantes da equipe, além de melhorar a transferência de conhecimento, preservar informações históricas, satisfazer necessidades contratuais ou legais, entre outros. **A documentação é importante, sim; mas valoriza-se mais o software em funcionamento, que é o que de fato agrega valor ao cliente.** Belê?

POR QUE VALORIZAR MAIS COLABORAÇÃO COM O CLIENTE DO QUE NEGOCIAÇÃO DE CONTRATOS?

Porque é importante o envolvimento contínuo do cliente! Aliás, desenvolvedores e clientes devem estar sempre lado a lado, visto que ambos possuem interesses em comum. Qual? Um software que agregue valor! No Modelo em Cascata, vocês devem se lembrar que o cliente até colaborava com a equipe no início do projeto (em geral, na fase de levantamento de requisitos), mas – depois disso – o cliente saía de cena e só aparecia novamente para ver o software já pronto.

E pior: muitas vezes, o cliente saía insatisfeito, porque o resultado não era o que ele esperava. Dessa forma, o manifesto ágil afirma que você tem que valorizar mais a sua relação com o cliente do que ficar discutindo itens de contrato: “*Isso não estava previsto no contrato*”; “*Isso não estava combinado previamente*”; “*Vou cobrar a mais porque você mudou tal coisa*”; entre outros. Professor, então contratos não são importantes? Claro que são!

Contratos regulam essa relação entre cliente e fornecedor, mas não se deve ser excessivamente rigoroso, porque isso pode acabar com a relação com seu cliente. Por falar em contrato, existem várias maneiras de fazer contratos de desenvolvimento ágil. Uma maneira comum é fixar o tempo e deixar o escopo variar. É o famoso: “**Tempo Fixo e Escopo Variável**”! Você fala para o seu cliente: “É o seguinte: eu faço tudo que você pedir desde que seja possível fazer no prazo tal”.

POR QUE VALORIZAR MAIS A RESPOSTA A MUDANÇAS DO QUE SEGUIR UM PLANEJAMENTO ESPECÍFICO?

Porque, em geral, é necessário obter respostas rápidas a mudanças e seguir um desenvolvimento contínuo do software. Todo projeto deve balancear o planejamento com a mudança, dependendo do nível de incerteza do projeto. Manter-se preso a um planejamento ultrapassado pode ser nocivo ao andamento do projeto. Galera, nós estamos no século 21! Uma empresa líder de mercado pode acabar de uma hora para outra – nós vemos isso o tempo todo.

Cadê o Orkut? Cadê o MSN? Cadê a Nokia? Cadê a Kodak? Cadê a BlockBuster? Todas essas empresas foram gigantes pouco tempo atrás e simplesmente morreram! Logo, a única certeza que você tem em um projeto é a instabilidade! **Logo, a equipe deve estar preparada para mudanças no escopo, tempo, custo, tecnologia, arquitetura, no paradigma de programação, regulamentações, leis, regras de conformidade, entre outros.**

Não tem como fazer um planejando e achar que ele vai ficar fixo ali ao longo do tempo – isso é pensamento do século passado (se muito!). Acreditam: mudanças vão ocorrer! Planejar é bom



demais. É tão bom que é recomendável refazer o planejamento a todo momento, de forma contínua e, não, fazer um planejamento estático e simplesmente segui-lo com todo rigor ignorando mudanças externas que venham a ocorrer. Fechou?



Agilidade x Velocidade

INCIDÊNCIA EM PROVA: BAIXA



Pessoal, agora vamos falar rapidamente sobre uma diferença importante! Vocês sabem qual a diferença entre agilidade e velocidade? Antes de explicá-la no contexto de desenvolvimento de software, eu vou explicar como uma metáfora em outros dois contextos para facilitar o entendimento. Vamos pensar no atleta Usain Bolt! O Usain Bolt é um cara veloz ou um cara ágil? Bem, em comparação com seres humanos normais, ele é mais ágil e mais veloz que todo mundo! No entanto, vamos pensar só no grupo dos grandes atletas que disputam mundiais e olimpíadas de atletismo. Nesse contexto, ele é absurdamente veloz, mas menos ágil que a maioria dos seus concorrentes. Como é, professor?

Vejam as duas imagens a seguir: observem que – à esquerda – temos cerca de vinte metros de corrida e o Bolt é o atleta de azul no meio. Notem também que ele está mais ou menos em quarto lugar na corrida. Por que? Porque agilidade é a capacidade de reagir ou responder adequadamente a mudanças e o Bolt sempre teve problemas de largada, uma vez que ele é mais alto e pesado que os outros.



Logo, ele acaba reagindo de forma mais lenta que seus adversários quando o tiro de início da corrida é disparado. Vejam: todos estão parados e, ao disparar o tiro, nós temos uma mudança. Essa mudança faz com que os atletas reajam e saiam da inércia. O Bolt demora mais que seus concorrentes a sair da inércia, uma vez que ele não responde tão bem quanto os outros a mudança. Nesse sentido, ele é ágil, mas não destoa dos outros pela sua agilidade.

Se nós tivéssemos corridas de 50 metros em vez de 100 metros, talvez ele não fosse tricampeão olímpico. Por outro lado, vejamos o final da corrida quando ele não tem mais que reagir a mudanças, ele só tem que correr até o fim dos 100 metros. Ele termina muito distante do segundo lugar! Por



que? Porque ele é um cara extremamente veloz, logo ele destoa de todos os outros com muita facilidade. Entenderam que agilidade não é velocidade? É a capacidade de reagir a mudanças!

A velocidade trata de quão rápido é possível entregar um software para o cliente. E, para isso, nós temos outras metodologias de desenvolvimento (Ex: *Rapid Application Development* (RAD) é capaz de desenvolver softwares em poucos meses). Utilizando outra metáfora, isso ocorre também quando você tem uma disputa entre um carro muito potente e pesado, e um carro menos potente e mais leve.

É provável que o carro mais leve, mesmo sendo menos potente, tenha uma arrancada melhor que o carro mais potente, logo ele é mais ágil. Ele é mais rápido? Não, o carro mais potente é mais rápido, mas ele é mais potente! Claro, pessoal, que esses são exemplos genéricos – apenas para entender a ideia. *Diego, e como esse conceito de agilidade pode ser utilizado no contexto de um desenvolvimento de software?*

No contexto de projetos de software, podemos imaginar: eu estou gerenciando meu projeto de um novo sistema e, de repente, descubro que vou ter que mudar a arquitetura do software – **não tem problema**; se eu descubro que, por conta de cortes de gastos, eu terei que reduzir o tamanho da minha equipe – **não tem problema**; se eu tiver que trocar a tecnologia utilizada porque ela se tornou defasada – **mais uma vez, não tem problema**.

Pressman afirma que a agilidade pode ser aplicada a qualquer processo de software. No entanto, para obtê-la, é essencial que o processo de software seja projetado para que a equipe possa adaptar e racionalizar suas tarefas; para que a equipe possa conduzir o planejamento compreendendo a fluidez de uma abordagem do desenvolvimento ágil; e para que a equipe possa eliminar tudo, exceto os artefatos essenciais do processo.

Além disso, deve enfatizar a estratégia de entrega incremental, entregando para o cliente o software operacional o mais rapidamente possível para o tipo de produto e ambiente operacional. **Essa são as diretrizes para que um processo de software qualquer possa ser, também, ágil.** Métodos ágeis são ágeis porque partem do princípio de que tem que responder adequadamente a mudanças que venham a ocorrer durante o ciclo de vida do projeto.

Eles são mais dinâmicos, adaptativos, interativos e colaborativos – eles se adaptam às necessidades de um projeto e às suas mudanças no decorrer do desenvolvimento; os métodos tradicionais são mais preeditivos/prescritivos, processuais, formais, documentais e contratuais – eles valorizam mais o planejamento de todos os aspectos do processo de desenvolvimento de software como um todo.



Princípios Ágeis

INCIDÊNCIA EM PROVA: ALTA

A seguir, nós vamos conhecer quais são os princípios do Manifesto Ágil. Eles vêm expressamente no manifesto e vocês podem encontrá-lo no site oficial:

WWW.AGILEMANIFESTO.ORG

Os 12 Princípios Ágeis

01



Satisfaça o consumidor

02



Aceite bem mudanças

03



Entregas frequentes

04



Trabalhe em conjunto

05



Confie e apoie

06



Conversas face a face

07



Softwares funcionando

08



Desenvolvimento sustentável

09



Atenção continua

10



Mantenha a simplicidade

11



Times auto-organizados

12



Refletir e ajustar

NÓS SEGUIMOS ESSES PRINCÍPIOS...

Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada e software com valor agregado.

Mudanças nos requisitos são bem-vindas, mesmo tardeamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.

Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.

Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.

Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.



O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face.

Software funcionando é a medida primária de progresso.

Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.

Contínua atenção à excelência técnica e bom design aumenta a agilidade.

Simplicidade – a arte de maximizar a quantidade de trabalho não realizado – é essencial.

As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.

Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

Professor, metodologias ágeis são recomendadas para projetos de qualquer tamanho e complexidade? Segundo Sommerville: "Todos os métodos têm limites, e os métodos ágeis são somente adequados para alguns tipos de desenvolvimento de sistema. Na minha opinião, eles são mais adequados para o desenvolvimento de sistemas de pequenas e médias empresas e produtos para computadores pessoais".

Diego, você concorda com essa afirmação? Não, eu discordo! **Acredito que ela já foi válida tempos atrás, mas hoje não é mais! Projetos Ágeis já são suficientemente maduros para serem aplicados a projetos complexos e de grande porte.** Pessoal, essa é só a minha opinião! Não é possível saber ainda a posição das bancas caso isso seja questionado em provas de concurso. Legal? Vamos ver agora exemplos de metodologias ágeis de desenvolvimento:

PRINCIPAIS METODOLOGIAS ÁGEIS		
SCRUM	CRYSTAL	XP
TDD	ATDD	BDD
FDD	DDD	MDD
DSDM	ASD	KANBAN
BADM	AUP	AGILE MODELING
OSSD	SCRUMBAN	

Agora vamos ver algumas diferenças básicas entre metodologias de desenvolvimento software tradicionais e metodologias ágeis:

CRITÉRIO	MODELOS TRADICIONAIS	MODELOS ÁGEIS
PLANEJAMENTO	Comumente realizado em detalhe para todo o projeto em sua fase inicial.	Planejamento de alto nível no início do projeto e os detalhes são realizados durante o projeto. Não é necessário possuir um planejamento detalhado de todo o projeto. A restrição se dá apenas em possuir os detalhes do trabalho para a próxima iteração.



RISCOS	Pode exigir um grande esforço e equipe para atuar com os riscos de todo o projeto.	Prioriza os riscos gerais do projeto, mas foca principalmente nos riscos das próximas iterações, atuando assim em um escopo bem reduzido. A própria equipe atua com os riscos e pode obter apoio externo.
EQUIPE	Possui profissionais com papéis bem definidos, quantificada e mobilizada conforme o planejamento do projeto. A equipe executa o projeto guiado pelo Gerente de Projetos conforme o plano estabelecido.	Equipe multidisciplinar, multifuncional e auto-organizada. Ela decide como fazer e atua de forma colaborativa.
TEMPO DE ENTREGA	É realizado conforme o plano estabelecido e pode durar semanas, meses ou até mesmo anos.	Fixo e é conforme a definição de duração das iterações que comumente varia entre 1 e 4 semanas.
ACEITAÇÃO DE MUDANÇAS	Gerenciamento formal de mudanças, pois exige alteração do planejamento já realizado e geralmente precisa passar por aprovações formais de um ou mais níveis hierárquicos.	Mudanças são bem-vindas. Evita-se mudar o escopo da iteração em andamento, mas o escopo das futuras iterações podem ser replanejado conforme a necessidade do cliente.
PREVISIBILIDADE	Depende do intervalo de monitoramento e controle do projeto. Quanto mais curto, maior a chance de prever as ocorrências futuras. Quanto maior o intervalo, menor a chance de prever as ocorrências futuras.	Tende a ter uma grande previsibilidade futura devido à constante análise e feedback através das oportunidades de inspeção e adaptação providas pelo método.
RESULTADOS AO LONGO DO TEMPO	Tende a demorar a dar resultados a curto prazo, pois as entregas são geralmente realizadas ao final do projeto. Melhores resultados são apresentados em projetos de maior duração.	Gera resultados a curto, médio e longo prazo, pois atua com entregas antecipadas e de valor agregado e contínuo ao cliente.
APRESENTAÇÃO DE INFORMAÇÕES DO PROJETO	Geralmente de uma apresentação formal previamente agendada com os stakeholders em intervalos de tempo. As informações podem ser detalhadas ou não conforme a necessidade do público envolvido.	Geralmente informal e utiliza radiadores de informação no ambiente de trabalho durante todo o projeto, de modo que as informações do projeto fiquem visíveis e transparentes a toda equipe e envolvidos.
PRAZO DE ENTREGA	Conforme estabelecido no planejamento do projeto. No caso de mudanças aprovadas, varia conforme os impactos das solicitações e podem ser traumáticas aos envolvidos quanto às suas expectativas.	Conforme o tamanho da iteração e o planejamento das releases para as entregas significativas.



DOCUMENTAÇÃO	Detalhada desde o início do projeto.	Abrangente no início e detalhada somente o necessário durante o projeto conforme os objetivos das iterações e releases.
ATUAÇÃO DO CLIENTE	Nas fases iniciais e nas principais validações do produto.	Durante todo o projeto, o cliente faz parte da equipe.
DISCUSSÕES E MELHORIAS	Geralmente em prazos longos através da realização de reuniões após uma grande etapa ou grande entrega do projeto.	Em prazos curtos, sempre ao final das iterações.
COMANDANTE	Gerente de Projetos.	Equipe do Projeto.
PAPÉIS	Claros e definidos.	Conforme a confiança na equipe e ambiente colaborativo.
PROCESSO	Guiado conforme o planejamento do projeto e nos processos estabelecidos no plano.	Empírico e guiado ao produto e às pessoas. Orientado à geração de valor e conforme priorização dos riscos.
RESULTADO	Melhor resultado em projetos com escopo muito bem definido e orientado a planejamento.	Melhor resultado em projetos cujo escopo é dinâmico e construído durante a execução do projeto.



Profissional Ágil

INCIDÊNCIA EM PROVA: BAIXÍSSIMA

No contexto do paradigma ágil e do manifesto ágil, o perfil de um profissional ágil é caracterizado por um conjunto de qualidades e habilidades que refletem os valores e princípios ágeis. Essas características são essenciais para navegar com sucesso em ambientes que adotam metodologias ágeis (Ex: Scrum, Kanban, XP, entre outras). Vejamos alguns dos aspectos mais importantes que definem um profissional ágil:

CARACTERÍSTICAS	MODELOS
FLEXIBILIDADE E ADAPTABILIDADE	Profissionais ágeis são capazes de se adaptar rapidamente a mudanças no escopo do projeto, requisitos ou prioridades. Eles entendem que mudanças são parte do processo de desenvolvimento e estão preparados para pivotar quando necessário.
COLABORAÇÃO E COMUNICAÇÃO	Eles valorizam a colaboração entre os membros da equipe e com os clientes ou stakeholders. A comunicação aberta é essencial, tanto para o sucesso do projeto quanto para a manutenção de um ambiente de trabalho positivo.
FOCO NO CLIENTE	A satisfação do cliente é uma prioridade. Profissionais ágeis trabalham de forma iterativa e incremental, buscando feedback contínuo para garantir que o produto final atenda ou exceda as expectativas do cliente.
APRENDIZADO CONTÍNUO	O aprendizado contínuo é fundamental em um ambiente ágil, dada a sua natureza evolutiva. Profissionais ágeis são proativos em seu desenvolvimento pessoal e profissional, buscando constantemente melhorar suas habilidades e conhecimentos.
PROATIVIDADE E AUTONOMIA	São profissionais que tomam a iniciativa, não esperando serem ditados para agir. Eles têm um senso de propriedade sobre seu trabalho e são capazes de trabalhar de forma independente, ao mesmo tempo em que entendem a importância da colaboração e do trabalho em equipe.
RESILIÊNCIA E PERSISTÊNCIA	Enfrentar desafios e superar obstáculos é parte do trabalho. Profissionais ágeis são resilientes frente às dificuldades e persistem até alcançar os objetivos desejados, vendo falhas como oportunidades de aprendizado.
HABILIDADE DE PRIORIZAÇÃO	Eles são adeptos de priorizar tarefas e recursos de forma eficaz, focando no que entrega o maior valor. Além disso, baseiam suas decisões em dados e feedbacks reais, em vez de suposições.
EMPATIA E SUPORTE AOS OUTROS	Profissionais ágeis valorizam a contribuição de cada membro da equipe e promovem um ambiente onde todos possam crescer. A empatia permite que entendam as perspectivas dos outros, facilitando a resolução de conflitos e a construção de relações saudáveis.

O perfil de um profissional ágil, portanto, vai além das habilidades técnicas, englobando uma série de competências comportamentais e uma mentalidade alinhada aos princípios do Manifesto Ágil. Estes incluem indivíduos e interações mais do que processos e ferramentas; software funcionando mais do que documentação abrangente; colaboração com o cliente mais do que negociação de contratos; e responder a mudanças mais do que seguir um plano.

Este conjunto de características habilita profissionais a navegar efetivamente em ambientes dinâmicos e orientados à inovação. Ainda nesse contexto, podemos falar sobre como identificar e entregar valor para o cliente digital. **Isso envolve entender profundamente as necessidades,**



preferências e comportamentos dos clientes no ambiente digital, e responder a eles de forma rápida e eficaz.

Identificar e entregar valor para clientes digitais é um processo contínuo de aprendizado e adaptação. Requer uma abordagem ágil que coloque o cliente no centro do desenvolvimento de produtos, garantindo que as soluções não apenas atendam às suas necessidades atuais, mas também antecipem e se adaptem às suas necessidades futuras. Enfatizo que o conceito de valor pode variar significativamente dependendo do cliente e do contexto.

Dessa forma, é crucial adotar uma abordagem centrada no cliente para garantir que o produto ou serviço entregue atenda ou exceda suas expectativas. Vejamos:

ESTRATÉGIAS	MODELOS
PESQUISA E FEEDBACK	Utilize pesquisas, entrevistas, análises de comportamento no site/app e feedback direto para entender o que os clientes valorizam, quais problemas eles precisam resolver, e como preferem interagir digitalmente.
PERSONA E JORNADAS DO CLIENTE	Crie personas detalhadas e mapeie as jornadas do cliente para identificar pontos de dor, necessidades não atendidas e oportunidades para agregar valor.
MVP (MINIMAL VISIBLE PRODUCT)	Lance versões iniciais do produto para testar hipóteses sobre o valor para o cliente, permitindo ajustes rápidos com base no feedback real.
ENTREGA CONTÍNUA	Adote práticas de entrega contínua para fornecer melhorias e novas funcionalidades de forma regular, mantendo o produto alinhado às expectativas e necessidades em evolução do cliente.
ANÁLISE DE DADOS E MÉTRICAS DE SUCESSO	Monitore métricas relevantes (como engajamento do usuário, conversão, retenção) para avaliar o sucesso das iniciativas e entender melhor o que os clientes valorizam.
TESTES A/B E EXPERIMENTAÇÃO	Realize testes A/B e experimentos para iterar sobre diferentes abordagens e identificar o que melhor atende às necessidades do cliente.
DESIGN CENTRADO NO USUÁRIO	Garanta que o produto seja intuitivo, fácil de usar e esteticamente agradável, criando uma experiência positiva que aumente a satisfação e a fidelidade do cliente.
USABILIDADE E ACESSIBILIDADE	Certifique-se de que o produto seja acessível a todos os usuários, incluindo aqueles com deficiências, e otimize a usabilidade para diferentes dispositivos e plataformas.
COMUNICAÇÃO PERSONALIZADA	Use a comunicação para educar, informar e engajar os clientes de maneira personalizada, baseando-se em seus interesses e comportamentos.
CONSTRUÇÃO DE COMUNIDADE	Fomente uma comunidade em torno do seu produto ou marca, incentivando o feedback, a participação e o engajamento dos usuários.
NOVAS TENDÊNCIAS E TECNOLOGIAS	Mantenha-se atualizado com as últimas tendências digitais e tecnologias para explorar novas maneiras de criar valor para o cliente.



**CULTURA DE
INOVAÇÃO**

Promova uma cultura que incentive a experimentação e a inovação, permitindo que a equipe explore novas ideias que possam entregar valor adicional aos clientes.



Ferramentas, Artefatos, Métricas e Indicadores

INCIDÊNCIA EM PROVA: BAIXÍSSIMA

Nesse tópico, vamos nos concentrar nas principais **ferramentas, artefatos, métricas e indicadores** ágeis. Vamos lá...

Ferramentas

Ferramentas ágeis são aplicativos de software projetados para apoiar equipes e organizações na implementação de práticas ágeis e na gestão de projetos ágeis. Elas facilitam a colaboração, o planejamento, a execução e o monitoramento de projetos, permitindo que as equipes sejam mais eficientes, transparentes e adaptáveis. Vejamos os principais tipos, características e exemplos de ferramentas ágeis:

TIPOS DE FERRAMENTAS	DESCRIÇÃO
GERENCIAMENTO DE PROJETO E COLABORAÇÃO	Suportam a planejamento de sprints, rastreamento de tarefas, quadros Kanban, e comunicação entre membros da equipe.
INTEGRAÇÃO E ENTREGA CONTÍNUA	Automatizam o processo de desenvolvimento de software, desde a integração do código até o teste e a entrega.
MONITORAMENTO E RELATÓRIOS	Oferecem visibilidade sobre o progresso do projeto, a saúde do código, e métricas de desempenho.
RASTREAMENTO DE BUGS E ISSUES	Facilitam a identificação, atribuição e resolução de bugs e outros problemas.
REPOSITÓRIOS E REVISÃO DE CÓDIGO	Permitem que as equipes gerenciem versões do código e revisem o trabalho uns dos outros para manter a qualidade.

EXEMPLOS	DESCRIÇÃO
JIRA	Oferece suporte para Scrum e Kanban, além de permitir o rastreamento de issues e bugs, planejamento de sprints, e relatórios ágeis.
TRELLO	Baseado em quadros Kanban, o Trello é intuitivo e fácil de usar para o rastreamento de tarefas e a colaboração em projetos de menor escala.
ÁSANA	Um aplicativo de gerenciamento de trabalho que facilita o planejamento de projetos, o rastreamento de tarefas e a colaboração entre equipes.
CONFLUENCE	Uma ferramenta de colaboração que permite que as equipes criem, compartilhem e colaborem em conteúdo e documentação.
GITHUB	Um repositório de código que facilita a colaboração em projetos de software, oferecendo ferramentas de revisão de código, gerenciamento de projetos e integração contínua.
GITLAB	Similar ao GitHub, oferece um repositório de código e ferramentas CI/CD integradas, além de funcionalidades para gerenciamento de issues e revisão de código.
SLACK	Uma plataforma de comunicação que promove a colaboração em tempo real, integrando-se facilmente com muitas outras ferramentas ágeis.

BENEFÍCIOS	DESCRIÇÃO
------------	-----------



MELHORIA DA COLABORAÇÃO	Facilitam a comunicação e a colaboração entre membros da equipe e stakeholders, independentemente de sua localização.
VISIBILIDADE E TRANSPARÊNCIA	Proporcionam uma visão clara do progresso do projeto, ajudando na tomada de decisões baseadas em dados.
EFICIÊNCIA OPERACIONAL	Automatizam tarefas repetitivas e promovem a eficiência na gestão de projetos e no desenvolvimento de software.
ADAPTABILIDADE E FLEXIBILIDADE	Permitem que as equipes respondam rapidamente a mudanças no projeto ou no ambiente de mercado.
MELHORIA CONTÍNUA	Facilitam o rastreamento de métricas e feedbacks, apoiando a melhoria contínua dos processos e produtos.

Artefatos

Os artefatos ágeis são elementos tangíveis produzidos durante o ciclo de desenvolvimento ágil.

Eles servem para ajudar as equipes a planejar, organizar, executar e revisar seu trabalho, mantendo a transparência e facilitando a comunicação entre todos os envolvidos no projeto. Eles são fundamentais para a implementação bem-sucedida de práticas ágeis e para a entrega de valor contínuo aos clientes. Vejamos alguns exemplos de artefatos:

ARTEFATOS	DESCRIÇÃO
BACKLOG DO PRODUTO	Uma lista ordenada de tudo que é necessário no produto final. É dinâmico e constantemente revisado e priorizado pelo Product Owner. Serve como uma fonte única de requisitos para qualquer mudança a ser feita no produto.
BACKLOG DA SPRINT	Uma lista de itens, escolhidos do Product Backlog, que a equipe se compromete a completar durante um Sprint (um período fixo durante o qual determinadas tarefas devem ser completadas). Guia a equipe sobre o trabalho a ser feito no Sprint atual, promovendo foco e comprometimento.
INCREMENTO	O conjunto de itens do Product Backlog completados durante um Sprint e as versões anteriores do produto. Deve estar em um estado utilizável e pronto para ser entregue ao cliente. Representa o progresso tangível em direção ao objetivo final do projeto.
QUADROS KANBAN	Um quadro visual para gerenciar o trabalho à medida que ele avança através de vários estágios do processo de desenvolvimento (por exemplo, "A Fazer", "Em Progresso", "Concluído"). Maximiza a eficiência do fluxo de trabalho ao limitar o trabalho em progresso e identificar gargalos.
GRÁFICO DE BURNDOWN	Gráficos que mostram o trabalho restante versus tempo. Eles podem ser usados tanto para o Sprint quanto para o Product Backlog. Fornecem uma visualização clara do progresso da equipe em direção à conclusão dos itens do backlog.
HISTÓRIAS DE USUÁRIO	Descrições curtas e simples da perspectiva do usuário final sobre uma funcionalidade específica que ele deseja que o produto tenha. Garantir que o desenvolvimento esteja focado nas necessidades e valores do usuário, facilitando a priorização e o planejamento do trabalho.



DEFINIÇÃO DE PRONTO	Um conjunto claro de critérios que especifica quando um item do backlog, como uma User Story ou uma tarefa, é considerado completo. Garante a qualidade e a consistência ao definir claramente o que é necessário para que o trabalho seja considerado finalizado.
ÉPICOS	Uma grande body of work que pode ser dividida em várias menores User Stories. É uma maneira de agrupar tarefas relacionadas que contribuem para um objetivo comum em larga escala. Facilita o gerenciamento e a priorização de grandes features ou funcionalidades que precisam ser desdobradas em tarefas mais gerenciáveis.

Métricas e Indicadores

As métricas ágeis são indicadores utilizados para medir o progresso, a eficiência e a eficácia das equipes e dos processos em ambientes que adotam metodologias ágeis de desenvolvimento, como Scrum, Kanban, entre outros. Elas fornecem insights valiosos sobre o desempenho da equipe, a qualidade do produto, a satisfação do cliente e outros aspectos críticos do processo de desenvolvimento de software. Vejamos exemplos de métricas ágeis:

MÉTRICAS E INDICADORES	DESCRIÇÃO
VELOCIDADE DA EQUIPE	Mede a quantidade de trabalho que uma equipe consegue completar durante um sprint. Geralmente, é calculada somando os pontos de história (ou qualquer outra unidade de medida) de todas as tarefas concluídas. Ajuda a prever a capacidade de entrega da equipe para futuros sprints, permitindo um planejamento mais preciso.
GRÁFICO DE BURNDOWN	Um gráfico que mostra a quantidade de trabalho restante versus tempo. Pode ser usado para sprints ou para o projeto como um todo. Fornece uma visualização clara de como a equipe está progredindo em direção à conclusão das tarefas dentro do prazo estabelecido.
LEAD/CYCLE TIME	Lead Time é o tempo total desde a solicitação até a entrega de uma tarefa. Cycle Time é o tempo que a tarefa leva para ser concluída, começando quando o trabalho efetivamente inicia. Indica a eficiência do processo de desenvolvimento, ajudando a identificar gargalos e a melhorar o fluxo de trabalho.
TAXA DE FALHAS EM PRODUÇÃO	Mede a frequência de falhas ou bugs que ocorrem no ambiente de produção. Ajuda a avaliar a qualidade do código e a eficácia das práticas de teste.
SATISFAÇÃO DO CLIENTE	Geralmente avaliada através de pesquisas ou Net Promoter Score (NPS), mede o quanto satisfeitos os clientes estão com o produto ou serviço entregue. Fornece feedback direto sobre o valor que o produto está entregando aos usuários finais.
VAZÃO (THROUGHPUT)	Número de itens de trabalho (como Histórias de Usuário) completados em um período de tempo específico. Avalia a produtividade da equipe e pode ajudar a prever a entrega de futuras funcionalidades.
WORK IN PROGRESS (WIP)	Quantidade de tarefas em andamento em um determinado momento. Monitorar o WIP ajuda a evitar sobrecarga da equipe e a identificar gargalos no fluxo de trabalho.



MÉTRICA DE FELICIDADE

Avaliação do nível de satisfação ou felicidade da equipe. Equipes felizes tendem a ser mais produtivas e engajadas. Esta métrica ajuda a identificar problemas que podem estar afetando o moral da equipe.

RETENÇÃO DE CLIENTE

Mede a porcentagem de clientes que continuam utilizando o produto ou serviço ao longo do tempo. Indica o sucesso do produto em manter os usuários engajados e satisfeitos.

Esses indicadores fornecem uma visão abrangente do desempenho do projeto e da equipe, permitindo ajustes contínuos para otimizar os processos e as entregas. No entanto, é muito importante escolher os indicadores que melhor se alinham aos objetivos específicos do projeto e da organização para garantir que as métricas suportem, e não atrapalhem, a entrega de valor.
Entendido?



Arquitetura Ágil

INCIDÊNCIA EM PROVA: BAIXÍSSIMA

A Arquitetura Ágil é um conjunto de valores, práticas e colaborações que apoiam ativamente o projeto e a arquitetura evolutiva de um sistema. Ela combina os princípios da arquitetura de software com os valores e práticas da metodologia ágil, buscando adaptar os princípios tradicionais de arquitetura de software para se alinhar melhor com as abordagens ágeis de desenvolvimento de software.

A arquitetura ágil busca reunir princípios e práticas que visam criar um software adaptável, flexível e resiliente em um ambiente de desenvolvimento ágil. Em contraste com a arquitetura tradicional, que define a estrutura geral do sistema antes do início do desenvolvimento, a arquitetura ágil se baseia em: planejamento incremental; colaboração e comunicação; qualidade e refatoração; e abordagem empírica.

CARACTERÍSTICA	Descrição
PLANEJAMENTO INCREMENTAL	A arquitetura do sistema é definida e implementada em iterações curtas e incrementais, permitindo adaptações às mudanças e feedback dos stakeholders. O foco está na entrega de valor funcional ao cliente o mais rápido possível.
COLABORAÇÃO E COMUNICAÇÃO	Arquitetos, desenvolvedores, testadores e stakeholders trabalham em conjunto para definir e refinar a arquitetura do sistema ao longo do projeto. A comunicação aberta e transparente é fundamental para garantir o alinhamento entre as diferentes partes interessadas.
QUALIDADE E REFATORAÇÃO	A qualidade da arquitetura é garantida através de práticas como revisão de código, testes automatizados e refatoração contínua. O código é constantemente aprimorado para eliminar duplicação, melhorar a legibilidade e aumentar a flexibilidade.
ABORDAGEM EMPÍRICA	A Arquitetura Ágil se baseia em feedback e aprendizado contínuo. As decisões de arquitetura são tomadas com base em dados e experimentos, ao invés de suposições ou preconceitos.
DESIGN EMERGENTE	A arquitetura evolui gradualmente ao longo do tempo, em vez de ser definida completamente no início do projeto. Ela emerge à medida que os requisitos são entendidos melhor e a equipe ganha experiência.
FEEDBACK RÁPIDO	A equipe busca obter feedback rápido sobre suas decisões arquiteturais por meio de revisões de código, testes e demonstrações frequentes do produto.
PADRÕES E PRÁTICAS	Utilização de padrões de projeto e práticas de engenharia de software que promovam a flexibilidade, escalabilidade, manutenibilidade e testabilidade do sistema.
REFATORAÇÃO CONSTANTE	A equipe está sempre disposta a refatorar o código e a arquitetura para melhorar sua qualidade e adaptá-la às mudanças nos requisitos e no ambiente.

PRINCÍPIOS	Descrição
SIMPLEZIDADE	Favorecer soluções simples e diretas, evitando complexidade desnecessária.
FLEXIBILIDADE	Adaptar a arquitetura às mudanças nos requisitos e no ambiente de desenvolvimento.



TESTABILIDADE	Facilitar o teste e a validação da arquitetura.
REUTILIZABILIDADE	Criar componentes reutilizáveis para reduzir o tempo e o esforço de desenvolvimento.
EVOLUTIVIDADE	Permitir que a arquitetura evolua facilmente para atender às novas necessidades do negócio.

BENEFÍCIOS

Maior adaptabilidade a mudanças nos requisitos e no ambiente.

Maior qualidade do software através de refatoração contínua.

Maior velocidade de desenvolvimento e entrega de valor ao cliente.

Maior colaboração e comunicação entre as diferentes partes interessadas.

Maior flexibilidade para lidar com incertezas e riscos.

A Arquitetura Ágil não é uma solução mágica para todos os problemas de desenvolvimento de software. É importante ter em mente que a implementação da Arquitetura Ágil exige tempo, esforço e compromisso de toda a equipe. Não existe uma fórmula única para o sucesso. **O importante é adaptar os princípios da Arquitetura Ágil de acordo com as necessidades específicas da organização e buscar a melhoria contínua dos processos.**



Qualidade Ágil

INCIDÊNCIA EM PROVA: BAIXÍSSIMA

A Qualidade Ágil refere-se à abordagem de garantia de qualidade que está alinhada com os princípios e práticas ágeis de desenvolvimento de software. **Em um contexto ágil, a qualidade não é vista como um aspecto separado do processo de desenvolvimento, mas sim como uma responsabilidade compartilhada por toda a equipe.** A importância da Qualidade Ágil no contexto ágil é significativa por várias razões:

CARACTERÍSTICA	DESCRIÇÃO
ENTREGA DE VALOR AO CLIENTE	A qualidade do produto é essencial para atender às expectativas do cliente e garantir sua satisfação. Em uma abordagem ágil, a qualidade é priorizada desde o início do projeto, o que resulta em entregas mais rápidas e consistentes de valor ao cliente.
FEEDBACK CONTÍNUO	A Qualidade Ágil é fundamentada no princípio de feedback contínuo. Isso significa que a equipe está constantemente buscando feedback dos usuários, clientes e stakeholders para identificar áreas de melhoria e garantir que o produto atenda às suas necessidades e expectativas.
ADAPTAÇÃO RÁPIDA	Em ambientes ágeis, as mudanças são inevitáveis e frequentes. Uma abordagem de qualidade ágil permite que a equipe se adapte rapidamente a essas mudanças, ajustando o produto conforme necessário para manter ou melhorar sua qualidade.
COLABORAÇÃO E COMUNICAÇÃO	A Qualidade Ágil promove a colaboração e a comunicação eficaz entre todos os membros da equipe, incluindo desenvolvedores, testadores, analistas de negócios e stakeholders. Isso ajuda a garantir que todos tenham uma compreensão clara dos requisitos e expectativas de qualidade.
REDUÇÃO DE RISCOS	Uma abordagem de Qualidade Ágil ajuda a reduzir os riscos associados ao desenvolvimento de software, identificando e corrigindo problemas mais cedo no ciclo de vida do projeto. Isso minimiza a probabilidade de erros e retrabalho, aumentando assim a eficiência e eficácia do processo de desenvolvimento.
CULTURA DE MELHORIA E CONTÍNUA	A Qualidade Ágil promove uma cultura de melhoria contínua, onde a equipe está constantemente buscando maneiras de aprimorar seus processos, práticas e produtos. Isso leva a uma maior inovação, eficiência e qualidade geral do produto.

Os princípios do **Manifesto Ágil** influenciam diretamente a garantia de qualidade no desenvolvimento de software. Vejamos:

- **Satisfação do Cliente Através da Entrega Contínua de Software Funcionando:** esse princípio destaca a importância de entregar software funcional de forma contínua e incremental. Isso implica em priorizar a entrega de funcionalidades que agreguem valor ao cliente, o que está intrinsecamente ligado à qualidade do produto. A equipe busca garantir que cada incremento entregue atenda aos padrões de qualidade definidos e às expectativas do cliente.



- **Mudanças nos Requisitos São Bem-Vindas, Mesmo Tardiamente no Desenvolvimento:** esse princípio reconhece a inevitabilidade das mudanças nos requisitos do projeto e enfatiza a capacidade de adaptação do time. Isso influencia a garantia de qualidade ao permitir que a equipe ajuste continuamente o produto para atender às novas necessidades e expectativas do cliente, mantendo a qualidade do produto ao longo do tempo.
- **Entregue Software Funcionando Frequentemente, de Preferência em Escalas de Semanas a Meses:** esse princípio promove entregas frequentes e incrementais de software funcional. Essa abordagem permite que a equipe receba feedback regular dos usuários e stakeholders, possibilitando a identificação precoce de problemas e a garantia de qualidade contínua ao longo do ciclo de desenvolvimento.
- **Colaboração Entre Clientes e Desenvolvedores Ao Longo do Projeto:** esse princípio destaca a importância da comunicação e colaboração contínuas entre a equipe de desenvolvimento e os clientes ou stakeholders. Essa colaboração contribui para a garantia de qualidade ao garantir que as necessidades do cliente sejam compreendidas e traduzidas corretamente em funcionalidades de software de alta qualidade.
- **Construa Projetos em Torno de Indivíduos Motivados. Dê a Eles o Ambiente e o Suporte Necessários e Confie:** este princípio enfatiza a importância de uma equipe motivada. Uma equipe engajada e confiável é fundamental para garantir a qualidade do produto, pois são eles que desenvolvem, testam e mantêm o software. Ao fornecer o ambiente e o suporte adequados, e confiar na equipe para realizar o trabalho, a qualidade do produto é promovida.

Nesse contexto, testes ágeis são uma parte fundamental das metodologias ágeis e são projetados para se adaptar aos princípios e práticas dessas metodologias:

CARACTERÍSTICA	DESCRIÇÃO
TEST-DRIVEN DEVELOPMENT (TDD)	TDD é uma prática de desenvolvimento que envolve escrever testes automatizados antes mesmo de escrever o código de produção. Os passos básicos do TDD são: escrever um teste automatizado que falha, escrever o código de produção para fazer o teste passar e, em seguida, refatorar o código para melhorá-lo. O TDD promove o desenvolvimento incremental e a qualidade do código desde o início, garantindo que cada funcionalidade tenha testes automatizados associados a ela.
BEHAVIOR-DRIVERED DEVELOPMENT (BDD)	BDD é uma abordagem que se concentra no comportamento esperado do software, expresso em termos de cenários de usuário. Os testes BDD são escritos em uma linguagem natural compreensível por todas as partes interessadas, como clientes, gerentes de projeto e desenvolvedores. Esses testes são então automatizados e executados para verificar se o software está se comportando conforme o esperado. O BDD promove uma compreensão compartilhada dos requisitos e uma abordagem centrada no usuário para o desenvolvimento.
AUTOMAÇÃO DE TESTES	A automação de testes é essencial para a entrega contínua de software de alta qualidade em ambientes ágeis. Os testes manuais consomem muito tempo e recursos, tornando-os impraticáveis para a entrega contínua. A automação de testes permite que a equipe execute testes de regressão de forma rápida e repetitiva, garantindo que as alterações no código não quebrem funcionalidades existentes. Isso aumenta a confiança na estabilidade do software e permite que a equipe mantenha um ritmo de entrega rápido e consistente.



AUTOMAÇÃO DA ENTREGA CONTÍNUA

Na entrega contínua, o software é entregue em ciclos curtos e frequentes. A automação de testes desempenha um papel crucial nesse processo, garantindo que o software entregue atenda aos padrões de qualidade definidos. Sem automação de testes eficaz, seria difícil ou impossível manter o ritmo de entrega contínua, pois os testes manuais seriam muito lentos e propensos a erros.

COBERTURA DE TESTES

A cobertura de testes é um aspecto importante dos Testes Ágeis. É essencial que a equipe tenha uma cobertura abrangente de testes automatizados para garantir que todas as funcionalidades do software sejam testadas de maneira adequada. Isso ajuda a identificar e corrigir problemas precocemente, reduzindo a probabilidade de bugs e garantindo a qualidade do produto final.



Gestão Ágil

INCIDÊNCIA EM PROVA: BAIXA

A abordagem tradicional de gerenciamento de projetos estava sendo apedrejada com diversas críticas a sua estrutura inflexível, complexidade, etc¹. A partir daí, começou a surgir diversas propostas de abordagens de gerenciamento de projetos, alternativas ao modelo tradicional, para o desenvolvimento de produtos inovadores em ambientes dinâmicos de projeto.

O gerenciamento tradicional de projetos apresenta uma redução em sua eficiência quando aplicados em projetos com elevado nível de incertezas, com equipes pequenas e co-localizadas, e elevado nível de interação entre membros da equipe de projeto e clientes. **Na abordagem tradicional, era praticamente possível antecipar ou prever variáveis relacionadas ao desenvolvimento do projeto.**

Professor, o que as novas abordagens propunham? Elas propunham novos princípios de gerenciamento em que o envolvimento da equipe e a utilização de técnicas simplificadas com enfoque no desenvolvimento da autogestão melhor se adequam a ambientes em que as incertezas e mudanças predominam. Professor, mas qual a definição exatamente de Gerenciamento Ágil de Projetos?

Highsmith diz que é um conjunto de princípios, valores e práticas que auxiliam a equipe a entregar produtos ou serviços de valor em um ambiente de projetos desafiador. Chin afirma que é um novo elemento base que pode contribuir para o desenvolvimento da teoria tradicional de gestão de projetos de modo que permita que as empresas sejam mais eficientes na gestão de projetos em ambientes incertos.

Já o termo “agilidade”, no contexto de desenvolvimento ágil e adaptável, significa que o time de projeto desenvolva habilidade necessária para criar e responder às mudanças ocorridas no projeto. **A abordagem do gerenciamento ágil de projetos deve ser encarada como uma habilidade para balancear flexibilidade e estabilidade.** Para tal, buscam-se pessoas com habilidade de improvisação

Augustine afirma que é necessário existir um nível adequado de flexibilidade com estabilidade, caos com ordem, execução com planejamento e exploração com otimização. **Além disso, ele declara que é imprescindível ter habilidade para entregar valor para o cliente, frente à imprevisibilidade e o dinamismo inerente aos projetos de novos produtos.** DeCarlo diz que é a arte e ciência de facilitar e gerenciar o fluxo de pensamentos, emoções e interações. *Para quê?*

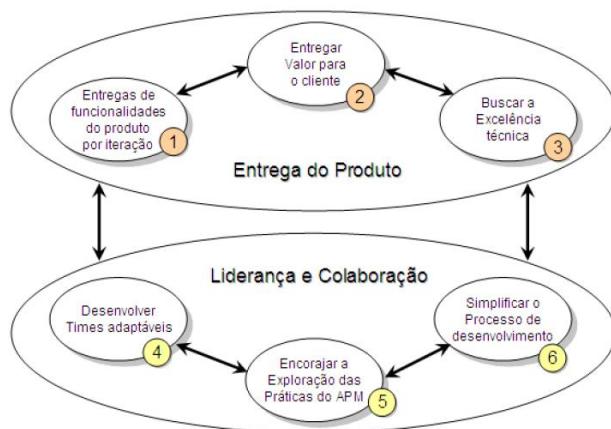
Para produzir resultados de valor em condições adversas e complexas que requerem velocidade, e estão sujeitas a mudanças constantes e elevados níveis de incertezas e estresse. *Galera... são muitas*

¹ Por exemplo: guias de melhores práticas e corpos de conhecimento como PMBOK.



definições, concordam? No entanto, elas têm algumas coisas em comum! **Todas falam sobre a necessidade de flexibilidade e habilidade para absorver mudanças durante o ciclo de vida do projeto.**

Além disso, há um forte enfoque humanista, valorização do aprendizado contínuo, e a capacidade dos indivíduos como participantes ativos do processo, ao invés de valorizar excessivamente as técnicas e processos de gestão de projetos. O Gerenciamento Ágil de Projetos (GAP) ou Agile Project Management (APM) possui alguns princípios básicos que regem sua aplicação e exploração apresentados na imagem a seguir:



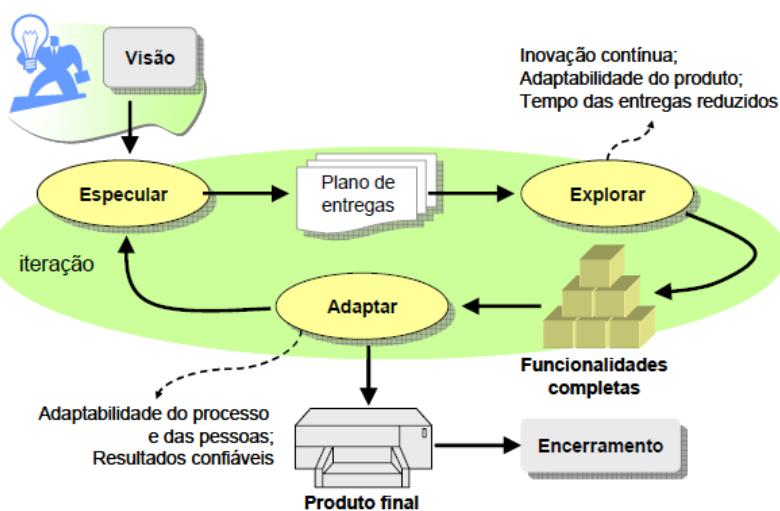
Esses princípios remetem a uma reflexão sobre o enfoque dado na gestão de projetos. Muitos dos autores da teoria tradicional enfatizam o valor do plano de projeto e a antecipação de eventos. **Situações incomuns quando se trata de desenvolvimento de produtos em contextos dinâmicos.** Por exemplo: entregar valor para o cliente também é uma diretriz da abordagem tradicional. *Bacana?*

No entanto, o que se questiona dessa abordagem é como agregar valor para o cliente, como simplificar o processo de desenvolvimento e como criar times adaptáveis em frente às dificuldades em antecipar atividades e seguir um planejamento sem a flexibilidade necessária para absorver eventuais mudanças no projeto. Galera, vamos ver uma tabelinha com as diferenças entre os dois:

ABORDAGEM	TRADICIONAL	ÁGIL
METAS DO PROJETO	Enfoque na finalização do projeto no tempo, custo e requisitos de qualidade.	Enfoque nos resultados do negócio, atingir múltiplos critérios de sucesso.
PLANO DO PROJETO	Uma coleção de atividades que são executadas como planejado para atender a tempo, custo e qualidade.	Organização e o processo para atingir as metas esperadas e os resultados para o negócio.
ABORDAGEM GERENCIAL	Rígida, com foco no plano inicial.	Flexível, variável e adaptativa.
TRABALHO E EXECUÇÃO	Previsível, mensurável, linear e simples.	Imprevisível, não-mensurável, não-linear e complexo.



INFLUÊNCIA DA ORGANIZAÇÃO	Mínimo, imparcial a partir do kick-off do projeto.	Afeta o projeto ao longo de sua execução.
CONTROLE DO PROJETO	Identificar desvios do plano inicial e corrigir o trabalho para seguir o plano.	Identificar mudanças no ambiente, e ajustar o plano adequadamente.
APLICAÇÃO DA METODOLOGIA	Aplicação genérica e igualitária em todos os projetos.	Adaptação do processo dependendo do tipo de projeto.
ESTILO DE GESTÃO	Um modelo atende todos os tipos de projetos.	Abordagem adaptativa, um único modelo não atende todos os tipos de projetos.



FASES	DESCRIÇÃO
VISÃO	O objetivo é determinar a visão do produto e o escopo de projeto, a comunidade do projeto, e definir como a equipe irá trabalhar e interagir. Define-se o que será entregue, os envolvidos e como o time pretende trabalhar.
ESPECULAÇÃO	O objetivo é planejar o projeto com base na visão preliminar construída com o apoio do time do projeto, sobre o que precisa ser entregue, quem são os envolvidos e qual será a estratégia adotada.
EXPLORAÇÃO	O objetivo é executar o que foi planejado, executando as entregas, promovendo a auto-organização e auto-disciplina da equipe de projetos e tratando da gestão das interações da equipe do projeto com o cliente.
ADAPTAÇÃO	O objetivo é rever os resultados da fase anterior, analisar o progresso do projeto e o desempenho da equipe de projetos, para eventuais adaptações no plano de projeto, entregas e plano de iterações, caso seja necessário.
ENCERRAMENTO	Transferem-se os conhecimentos-chave adquiridos no projeto, e celebrado os resultados obtidos. Recomendam-se mini-fechamentos ao final de cada iteração no projeto.

A gestão ágil de projetos é uma abordagem que prioriza a entrega rápida e contínua de valor ao cliente, adaptando-se às mudanças de forma dinâmica e promovendo a colaboração entre as equipes. Vamos explorar as noções básicas dessa abordagem, focando em cooperação,



flexibilidade de escopo, interatividade, autonomia e empoderamento de equipes e a programação em pares:

Cooperação

Descrição: a cooperação é um dos pilares da gestão ágil de projetos. Envolve a colaboração contínua entre todos os membros da equipe, bem como entre a equipe e os stakeholders, para garantir que todos estejam alinhados e trabalhando em direção ao mesmo objetivo.

Na prática, isso significa que as equipes ágeis operam em ciclos curtos e interativos, como sprints, onde feedback regular é compartilhado. Reuniões diárias (daily stand-ups) são comuns, permitindo que os membros da equipe atualizem uns aos outros sobre seu progresso e desafios. A cooperação não se limita à equipe interna; envolve também os clientes e stakeholders, que são integrados ao processo de desenvolvimento através de revisões e validações constantes.

Flexibilidade de Escopo

Descrição: diferente da gestão tradicional de projetos, onde o escopo é definido rigidamente no início, a gestão ágil promove a flexibilidade de escopo. Isso significa que o escopo do projeto pode ser ajustado continuamente com base no feedback recebido e nas necessidades emergentes do cliente.

Na prática, essa flexibilidade é gerenciada através de um backlog priorizado, que é constantemente revisado e atualizado conforme o projeto avança. O Product Owner (dono do produto) trabalha junto com a equipe para ajustar o backlog, garantindo que as funcionalidades mais valiosas sejam desenvolvidas primeiro. Isso permite que a equipe responda rapidamente a mudanças no mercado ou nos requisitos do cliente, sem comprometer o cronograma geral do projeto.

Interatividade

Descrição: a gestão ágil enfatiza a interatividade, tanto no processo de desenvolvimento quanto na comunicação entre os membros da equipe e os stakeholders. Isso significa que o trabalho é dividido em pequenos incrementos, cada um dos quais é desenvolvido, testado e revisado antes de avançar.

Na prática, iterações curtas, geralmente chamadas de sprints (em metodologias como Scrum), são usadas para criar versões incrementais e funcionais do produto. Ao final de cada sprint, uma revisão é realizada para demonstrar o progresso, obter feedback e ajustar o curso do projeto se necessário. Isso cria um ciclo contínuo de melhoria e garante que o produto final evolua de acordo com as necessidades reais do usuário.

Autonomia



Descrição: na gestão ágil, as equipes são incentivadas a serem autônomas. Isso significa que elas têm a liberdade e a responsabilidade de tomar decisões sobre como o trabalho será realizado, sem a necessidade de aprovações ou supervisão constante.

Na prática, a autonomia é promovida através de uma clara definição de papéis e responsabilidades, como Scrum Master, Product Owner, e equipe de desenvolvimento. Cada membro da equipe tem a liberdade de escolher a melhor maneira de realizar suas tarefas, dentro das diretrizes gerais do projeto. Isso não apenas acelera a tomada de decisões, mas também aumenta a motivação e o comprometimento da equipe.

Empoderamento de Equipes

Descrição: o empoderamento das equipes é crucial para o sucesso na gestão ágil. Equipes empoderadas têm o poder de tomar decisões importantes, sugerir melhorias e influenciar diretamente o rumo do projeto.

Na prática, empoderar uma equipe significa fornecer a ela os recursos, informações e autoridade necessárias para cumprir seus objetivos. Isso pode incluir acesso a ferramentas, treinamento, ou apoio da liderança. O empoderamento também envolve a criação de um ambiente onde a equipe se sinta segura para experimentar, falhar e aprender com seus erros. Líderes ágeis atuam como facilitadores, removendo obstáculos e apoiando a equipe, em vez de microgerenciá-la.

Programação em Pares

Descrição: prática de desenvolvimento de software onde duas pessoas trabalham juntas no mesmo código em um único computador. Um dos programadores, chamado de "piloto", escreve o código, enquanto o outro, o "navegador", revisa cada linha de código à medida que é escrita, pensando em melhorias, erros potenciais e considerando o impacto geral da solução.

Na prática, a programação em pares promove a colaboração, a melhoria contínua e a qualidade do código. Embora possa parecer menos eficiente inicialmente, devido ao envolvimento de dois desenvolvedores em uma única tarefa, os benefícios de um código mais limpo, menos erros e maior troca de conhecimento podem compensar essa percepção, resultando em um desenvolvimento de software mais eficaz e sustentável.

Na gestão ágil de projetos, cooperação, flexibilidade de escopo, interatividade, autonomia e empoderamento de equipes são elementos fundamentais que trabalham juntos para promover uma abordagem mais adaptável e orientada para o valor. Essas práticas não apenas aumentam a eficiência e a eficácia do processo de desenvolvimento, mas também criam um ambiente onde as equipes podem prosperar, inovar e entregar produtos de alta qualidade que atendem às necessidades reais dos clientes.



Método Ágil x Método Lean

INCIDÊNCIA EM PROVA: BAIXA

Método Lean, ou Método Enxuto, é uma filosofia de gestão focada na redução de desperdícios dentro de sistemas de manufatura enquanto simultaneamente maximiza a produção. Originada no Sistema Toyota de Produção, esta abordagem busca melhorar a eficiência e a eficácia dos processos produtivos através da eliminação contínua de tudo que não agrega valor ao produto final na perspectiva do cliente.

Sendo rigoroso, há diferenças de nomenclatura: Método Lean é uma filosofia de gestão centrada na criação de valor para o cliente final com o mínimo de desperdício possível, criada pela Toyota; Lean Manufacturing é a aplicação específica da filosofia Lean no contexto da produção industrial; e Lean IT é a aplicação dos princípios Lean no contexto da tecnologia da informação. No entanto, a maioria das provas utilizará o termo Método Lean para o contexto de tecnologia da informação.

O Lean Manufacturing possui cinco princípios:

#	PRINCÍPIO	Descrição
PRINCÍPIO #1	IDENTIFICAÇÃO DE VALOR	O valor é definido do ponto de vista do cliente final. O primeiro princípio do Lean Manufacturing é entender o que realmente agrega valor ao produto ou serviço do ponto de vista do cliente. Tudo o que não agrega valor é considerado desperdício e deve ser eliminado.
PRINCÍPIO #2	MAPEAMENTO DO FLUXO DE VALOR	Mapear o fluxo de valor envolve identificar todos os passos – tanto os que agregam valor quanto os que não agregam – necessários para trazer um produto do conceito ao cliente final. O objetivo é entender o fluxo de materiais e informações e identificar oportunidades de melhoria.
PRINCÍPIO #3	CRIAÇÃO DE FLUXO CONTÍNUO	Depois de eliminar os desperdícios identificados, os processos restantes devem ser organizados de maneira que o fluxo de produção seja contínuo e suave. Isso significa garantir que os produtos se movam rapidamente e sem interrupções através do processo de produção, minimizando esperas e gargalos.
PRINCÍPIO #4	PRODUÇÃO PUXADA	A produção puxada significa que a produção é desencadeada pela demanda do cliente, e não por previsões de vendas ou produção em massa. Isso ajuda a evitar excesso de estoque e sobreprodução, produzindo apenas o que é necessário, quando é necessário.
PRINCÍPIO #5	PERFEIÇÃO E MELHORIA CONTÍNUA	O objetivo final do Lean Manufacturing é a busca pela perfeição através da melhoria contínua. O processo de melhoria contínua, ou Kaizen, envolve todos os níveis da organização e se concentra em pequenas melhorias incrementais em todos os aspectos do processo de produção.

Já o Método Lean para Desenvolvimento de Software tem sete princípios:



#	PRINCÍPIO	DESCRIÇÃO
PRINCÍPIO #1	ELIMINAR DESPERDÍCIO	Deve-se eliminar tudo aquilo que não é percebido pelo cliente, por não agregar valor para ele. Ex: passos extras, burocracia, documentação que não será lida, processo pesado, etc. Também existem aqueles desperdícios que são trabalhos parcialmente prontos - tudo que teve um começo, mas não teve fim e, portanto, não será utilizado.
PRINCÍPIO #2	AMPLIFICAR/criar CONHECIMENTO	Deve-se garantir que o conhecimento sobre o software seja criado durante o desenvolvimento, em vez de ter uma lista de requisitos e/ou um layout recomendando como deve ser o resultado da aplicação antes do início de seu desenvolvimento.
PRINCÍPIO #3	FORTALECER O TIME / RESPEITAR AS PESSOAS	O software que está sendo produzido é uma espécie de espelho da equipe que o está desenvolvendo. Para que as pessoas se sintam motivadas e engajadas na atuação em equipe, eles precisam de respeito e confiança. Deve-se criar um ambiente onde a equipe trabalhe de forma auto-organizada e auto-dirigida, evitando micro-gerenciamento.
PRINCÍPIO #4	ENTREGAS RÁPIDAS	Uma dica importante é que, sem entregas rápidas, você não consegue receber um retorno, ou seja, você não consegue saber o que errou para tentar corrigir. Por isso, procurar a velocidade na entrega é uma maneira de garantir que o cliente tenha em mãos aquilo que ele precisava para hoje e não o que precisou.
PRINCÍPIO #5	CONSTRUIR / INTEGRAR QUALIDADE	Segundo os criadores da teoria, a qualidade é inegociável e deve ser entregue em duas dimensões: a integridade percebida e conceitual. A integridade percebida quer dizer que foi entregue ao cliente um produto usual, funcional, confiável. A integridade conceitual quer dizer que o sistema tem pontos centrais altamente coesos e fáceis.
PRINCÍPIO #6	OTIMIZAR O TODO	Deve-se entender que o software concluído é muito mais que a soma das partes entregues e verificar como ele está alinhado com os objetivos da empresa. O ideal não é olhar apenas para o desenvolvimento, mas para como aquele requisito está sendo atendido, como ele está sendo detalhado e repassado para entrar em desenvolvimento, entre outros.
PRINCÍPIO #7	ADIAR DECISÕES/ COMPROMISSOS	Deve-se diminuir as incertezas, retardando decisões até que elas sejam formuladas em cima de acontecimentos mais conhecidos, previsíveis e firmes. Decisões tomadas tardiamente devem ser mais corretas, uma vez que as melhores são baseadas em fatos ocorridos e não em suposições ou especulações.

O Lean Manufacturing tem diversos benefícios:

- **Redução de Custos:** a diminuição de desperdícios e a melhoria da eficiência operacional levam a uma redução significativa nos custos de produção. Isso inclui custos associados a materiais, armazenamento, transporte, e tempo de inatividade, permitindo que a empresa seja mais competitiva com preços e margens de lucro;
- **Melhoria de Qualidade:** ao focar na eliminação de defeitos e na implementação de um processo de melhoria contínua, o Lean Manufacturing ajuda a aumentar a qualidade dos produtos. Isso reduz a quantidade de retrabalho e desperdício associado à correção de erros, além de melhorar a reputação da marca;



- **Aumento da Eficiência:** a implementação de processos mais eficientes e a eliminação de atividades que não agregam valor resultam em um aumento significativo da produtividade. Os trabalhadores podem focar suas energias em tarefas que realmente contribuem para o valor do produto, otimizando o uso do tempo e dos recursos;
- **Maior Flexibilidade:** a simplificação e a otimização dos processos de produção aumentam a flexibilidade da empresa, permitindo uma adaptação mais rápida às mudanças nas preferências dos consumidores e às condições do mercado. Isso é crucial em um ambiente empresarial cada vez mais volátil e competitivo;
- **Redução de Desperdícios:** o principal objetivo do Lean é identificar e eliminar desperdícios (atividades que não agregam valor ao produto final). Isso inclui desperdícios de tempo, material, movimento, espaço, e outros recursos, levando a processos mais eficientes e à redução de custos de produção;
- **Melhoria no Tempo de Entrega:** com processos mais eficientes e um sistema de produção puxada que responde diretamente à demanda do cliente, o Lean Manufacturing pode reduzir significativamente os tempos de entrega. Isso aumenta a satisfação do cliente e permite uma resposta mais rápida às mudanças do mercado;
- **Engajamento dos Funcionários:** o Método Lean promove um ambiente de trabalho inclusivo e colaborativo, onde os funcionários são encorajados a participar ativamente do processo de melhoria contínua. Isso pode aumentar a satisfação e o engajamento dos funcionários, reduzir a rotatividade e melhorar a cultura da empresa.
- **Sustentabilidade:** ao reduzir desperdícios e otimizar o uso de recursos, o Lean contribui para práticas de negócios mais sustentáveis. Isso não só reduz o impacto ambiental, mas também melhora a imagem da empresa junto aos consumidores, que estão cada vez mais conscientes da sustentabilidade.

Ele serviu de base para o método ágil e tem várias características em comum, mas são diferentes. A tabela abaixo organiza um comparativo para vocês terem noção das diferenças:

CARACTERÍSTICA	MÉTODO LEAN	MÉTODO ÁGIL
ORIGEM E FOCO	Originou-se no Sistema Toyota de Produção na indústria automobilística japonesa, com foco principal na eliminação de desperdícios ("muda") para otimizar a eficiência do processo de produção. Embora tenha começado na manufatura, os princípios Lean foram adaptados para outras áreas, incluindo desenvolvimento de software e serviços.	Desenvolvido inicialmente para o campo do desenvolvimento de software como uma resposta às limitações dos métodos tradicionais de gerenciamento de projetos (como o modelo cascata). O foco é na adaptabilidade, na entrega incremental de produtos e na colaboração constante com o cliente.



PRINCÍPIOS E PRÁTICAS	Baseia-se em princípios como Kaizen (melhoria contínua), eliminação de desperdícios, e Just-In-Time. Práticas incluem mapeamento do fluxo de valor e otimização dos processos.	Baseia-se nos princípios do Manifesto Ágil, como colaboração cliente-desenvolvedor, resposta a mudanças e entrega incremental. Práticas incluem Scrum, Kanban, programação em pares, e integração contínua.
ABORDAGEM DE IMPLEMENTAÇÃO	Pode ser aplicado de maneira mais ampla além do desenvolvimento de produtos, incluindo processos administrativos e operacionais, com um forte foco em eficiência operacional e redução de desperdícios em todos os aspectos da organização.	Focado primariamente no desenvolvimento de software e projetos que beneficiam de uma abordagem iterativa e incremental. A implementação Ágil é caracterizada por sprints ou iterações, planejamento adaptativo e equipe multidisciplinar.
MEDIDA DE SUCESSO	O sucesso é medido pela eficiência do processo, a redução de desperdícios e a capacidade de entregar valor contínuo ao cliente com o mínimo de recursos possíveis.	O sucesso é frequentemente medido pela satisfação do cliente, a capacidade de responder rapidamente a mudanças e a entrega frequente de incrementos de software que funcionam.

Embora o Método Lean e o Método Ágil compartilhem princípios de melhoria contínua e eficiência, eles se diferenciam em suas origens, focos principais, e métodos de implementação. **O Método Lean é mais abrangente em termos de aplicação a processos de negócios e operações, enquanto o Método Ágil é mais específico para o desenvolvimento de software e projetos que se beneficiam de uma abordagem flexível e iterativa.**

Há também o Método Kaizen: conceito japonês que significa "melhoria contínua". **Diferente do Lean, que pode ser aplicado com projetos específicos para melhorias, o Kaizen é um processo contínuo e constante.** Envolve todos os empregados, desde a alta administração até os trabalhadores da linha de frente, na busca por pequenas melhorias diárias que, somadas, resultam em uma significativa melhoria da eficiência e qualidade com o passar do tempo.

O Kaizen pode usar muitas das mesmas ferramentas que o Lean, mas com ênfase na participação dos funcionários e na cultura de melhoria contínua. *Fechado?* Seguindo...



RESUMO

ESTAMOS DESCOBRINDO MANEIRAS MELHORES DE DESENVOLVER SOFTWARE, FAZENDO-O NÓS MESMOS E AJUDANDO OUTROS A FAZEREM O MESMO. ATRAVÉS DESTE TRABALHO. PASSAMOS A VALORIZAR:

INDIVÍDUOS E INTERAÇÕES

MAIS QUE

PROCESSOS E FERRAMENTAS

SOFTWARE EM FUNCIONAMENTO

MAIS QUE

DOCUMENTAÇÃO ABRANGENTE

COLABORAÇÃO COM O CLIENTE

MAIS QUE

NEGOCIAÇÃO DE CONTRATOS

RESPONDER A MUDANÇAS

MAIS QUE

SEGUIR UM PLANO

OU SEJA, MESMO HAVENDO VALOR NOS ITENS À DIREITA, VALORIZAMOS MAIS OS ITENS À ESQUERDA.

INDIVÍDUOS E INTERAÇÕES MAIS QUE
PROCESSOS E FERRAMENTAS

Devemos entender que o desenvolvimento de software é uma atividade humana e que a qualidade da interação entre as pessoas pode resolver problemas crônicos de comunicação. Processos e Ferramentas são importantes, mas devem ser simples e úteis.

SOFTWARE EM FUNCIONAMENTO MAIS
QUE DOCUMENTAÇÃO ABRANGENTE

O maior indicador de que sua equipe realmente construiu algo é software funcionando. Clientes querem é resultado e isso pode ser com software funcionando. Documentação também é importante, mas que seja somente o necessário e que agregue valor.

COLABORAÇÃO COM O CLIENTE MAIS
QUE NEGOCIAÇÃO DE CONTRATOS

Devemos atuar em conjunto com o cliente e não “contra” ele ou ele “contra” a gente. O que deve acontecer é colaboração, tomada de decisões em conjunto e trabalho em equipe, fazendo que todos sejam um só em busca de um objetivo.

RESPONDER A MUDANÇAS MAIS QUE
SEGUIR UM PLANO

Desenvolver software e produtos é um ambiente de alta incerteza e por isso não podemos nos debruçar em planos enormes e cheio de premissas. O que deve ser feito é aprender com as informações e feedbacks e adaptar o plano a todo momento.

PRINCIPAIS METODOLOGIAS ÁGEIS

SCRUM

CRYSTAL

XP

TDD

ATDD

BDD

FDD

DDD

MDD

DSDM

ASD

KANBAN

BADM

AUP

AGILE MODELING

OSSD

SCRUMBAN



Os 12 Princípios Ágeis

01



Satisfaça o consumidor

02



Aceite bem mudanças

03



Entregas frequentes

04



Trabalhe em conjunto

05



Confie e apoie

06



Conversas face a face

07



Softwares funcionando

08



Desenvolvimento sustentável

09



Atenção continua

10



Mantenha a simplicidade

11



Times auto-organizados

12



Refletir e ajustar

NÓS SEGUIMOS ESSES PRINCÍPIOS...

Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada e software com valor agregado.

Mudanças nos requisitos são bem-vindas, mesmo tardivamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.

Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.

Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.

Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.

O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face.

Software funcionando é a medida primária de progresso.

Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.

Contínua atenção à excelência técnica e bom design aumenta a agilidade.

Simplicidade – a arte de maximizar a quantidade de trabalho não realizado – é essencial.



As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.

Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

CRITÉRIO	MODELOS TRADICIONAIS	MODELOS ÁGEIS
PLANEJAMENTO	Comumente realizado em detalhe para todo o projeto em sua fase inicial.	Planejamento de alto nível no início do projeto e os detalhes são realizados durante o projeto. Não é necessário possuir um planejamento detalhado de todo o projeto. A restrição se dá apenas em possuir os detalhes do trabalho para a próxima iteração.
RISCOS	Pode exigir um grande esforço e equipe para atuar com os riscos de todo o projeto.	Prioriza os riscos gerais do projeto, mas foca principalmente nos riscos das próximas iterações, atuando assim em um escopo bem reduzido. A própria equipe atua com os riscos e pode obter apoio externo.
EQUIPE	Possui profissionais com papéis bem definidos, quantificada e mobilizada conforme o planejamento do projeto. A equipe executa o projeto guiado pelo Gerente de Projetos conforme o plano estabelecido.	Equipe multidisciplinar, multifuncional e auto-organizada. Ela decide como fazer e atua de forma colaborativa.
TEMPO DE ENTREGA	É realizado conforme o plano estabelecido e pode durar semanas, meses ou até mesmo anos.	Fixo e é conforme a definição de duração das iterações que comumente varia entre 1 e 4 semanas.
ACEITAÇÃO DE MUDANÇAS	Gerenciamento formal de mudanças, pois exige alteração do planejamento já realizado e geralmente precisa passar por aprovações formais de um ou mais níveis hierárquicos.	Mudanças são bem-vindas. Evita-se mudar o escopo da iteração em andamento, mas o escopo das futuras iterações podem ser replanejado conforme a necessidade do cliente.
PREVISIBILIDADE	Depende do intervalo de monitoramento e controle do projeto. Quanto mais curto, maior a chance de prever as ocorrências futuras. Quanto maior o intervalo, menor a chance de prever as ocorrências futuras.	Tende a ter uma grande previsibilidade futura devido à constante análise e feedback através das oportunidades de inspeção e adaptação providas pelo método.
RESULTADOS AO LONGO DO TEMPO	Tende a demorar a dar resultados a curto prazo, pois as entregas são geralmente realizadas ao final do projeto. Melhores resultados são apresentados em projetos de maior duração.	Gera resultados a curto, médio e longo prazo, pois atua com entregas antecipadas e de valor agregado e contínuo ao cliente.
APRESENTAÇÃO DE INFORMAÇÕES DO PROJETO	Geralmente de uma apresentação formal previamente agendada com os stakeholders em intervalos de tempo. As informações podem ser detalhadas ou não conforme a necessidade do público envolvido.	Geralmente informal e utiliza radiadores de informação no ambiente de trabalho durante todo o projeto, de modo que as informações do projeto fiquem visíveis e transparentes a toda equipe e envolvidos.
PRAZO DE ENTREGA	Conforme estabelecido no planejamento do projeto. No caso de mudanças aprovadas, varia conforme os impactos das solicitações	Conforme o tamanho da iteração e o planejamento das releases para as entregas significativas.



	e podem ser traumáticas aos envolvidos quanto às suas expectativas.	
DOCUMENTAÇÃO	Detalhada desde o início do projeto.	Abrangente no início e detalhada somente o necessário durante o projeto conforme os objetivos das iterações e releases.
ATUAÇÃO DO CLIENTE	Nas fases iniciais e nas principais validações do produto.	Durante todo o projeto, o cliente faz parte da equipe.
DISCUSSÕES E MELHORIAS	Geralmente em prazos longos através da realização de reuniões após uma grande etapa ou grande entrega do projeto.	Em prazos curtos, sempre ao final das iterações.
COMANDANTE	Gerente de Projetos.	Equipe do Projeto.
PAPÉIS	Claros e definidos.	Conforme a confiança na equipe e ambiente colaborativo.
PROCESSO	Guiado conforme o planejamento do projeto e nos processos estabelecidos no plano.	Empírico e guiado ao produto e às pessoas. Orientado à geração de valor e conforme priorização dos riscos.
RESULTADO	Melhor resultado em projetos com escopo muito bem definido e orientado a planejamento.	Melhor resultado em projetos cujo escopo é dinâmico e construído durante a execução do projeto.

CARACTERÍSTICA	LEAN	ÁGIL
OBCECADO COM...	DESPERDÍCIO	CLIENTES E MERCADOS
GERENCIA...	PROCESSOS	INCERTEZAS
ENTREGA DE...	VALOR	PRODUTO EM FUNCIONAMENTO
APLICA...	HEURÍSTICAS	PRINCÍPIOS
FOCA NO PROCESSO DE...	PADRONIZAÇÃO E CONFORMIDADE	AUTOGERENCIAMENTO P/ MAXIMIZAR AUTONOMIA

 PARA MAIS DICAS:

WWW.INSTAGRAM.COM/PROFESSORDIEGOCARVALHO

