UniFil
EDUCAÇÃO A DISTÂNCIA

Unidade 3

Metodologias Ágeis SAFe e FDD

Abertura

Olá, amigo(a) discente! Seja bem-vindo(a)!

Diante da existência de uma grande demanda por desenvolvimento de software, diversas empresas vêm adotando os métodos ágeis como uma alternativa aos métodos tradicionais de desenvolvimento de software. Seguindo essa visão, nesta unidade vamos compreender e discutir os principais conceitos que corroboram para o sucesso das metodologias ágeis.

Na aula 01, vamos estudar os conceitos e práticas da metodologia *SAFe* e como esse método promove o alinhamento, a colaboração e a entrega entre diversas equipes ágeis.

Na aula 02, vamos falar um pouco sobre a metodologia FDD, essa metodologia é guiada por funcionalidade. Trata-se de um modelo incremental e interativo que assume alguns lemas como premissa: resultados frequentes, tangíveis e funcionais.

Objetivos

- Conhecer os conceitos das metodologias *Scaled Agile Framework e Feature Driven Development*
- Conhecer como aplicar na prática os conceitos do *Scaled Agile Framework e Feature Driven Development*

Conteúdo Programático

Aula 01 – *Scaled Agile Framework* conceitos e aplicação prática

Aula 02 – *Feature Driven Development* fundamentos e aplicação prática

Quer assistir às videoaulas em seu celular? Basta apontar a câmera para os **QR Codes** distribuídos neste conteúdo.

Caso necessário, instale um aplicativo de leitura QR Code no celular e efetue o login na sua conta Gmail.

Scaled Agile Framework conceitos e aplicação prática

Alternativas de investimentos em Sistemas da Informação

Olá, estudante! Vamos começar agora um estudo muito interessante sobre as metodologias ágeis, conhecer como funciona o *SAFe* e sua aplicação no cotidiano.

A boa aplicação dessa metodologia pode alterar totalmente a experiência de construção de um software. Com a aplicação correta deste método as chances de entregar um software com qualidade e que gere valor para o usuário é muito maior.

Assim, nesta aula vamos conhecer melhor a metodologia *SAFe* sua história, conceitos e como podemos aplicar essa metodologia na prática.

Videoaula 1

Utilize o QR Code para assistir!

Assista o vídeo para entender melhor o assunto.

O *SAFe (Scaled Agile Framework)* tem seu início em 2011 e foi lançado por *Dean Leffingwell e Drew Jemilo* com o foco em auxiliar grandes empresas a desenvolver sistemas que melhor atendessem às necessidades dos seus clientes, considerando que essas necessidades passam por mudanças constantemente.

Com isso, a proposta desta metodologia é implementar práticas ágeis em escala dentro das empresas, promovendo o conhecimento e colaboração entre todos. Para tornar isso possível o SAFe é baseado nos princípios do *Lean*.

Dessa forma desenvolvendo consigo algumas características, como:

- Economia
- Rapidez na entrega
- Qualidade na entrega
- Melhoria contínua

Videoaula 2

Utilize o QR Code para assistir!

Assista o vídeo para entender melhor o assunto.

Assim como em outras metodologias, para conseguir aplicar as práticas ágeis em escala nas empresas existem princípios e valores, esses que incentivam a liderança, comunicação e transparência entre todos. São valores do *SAFe*:

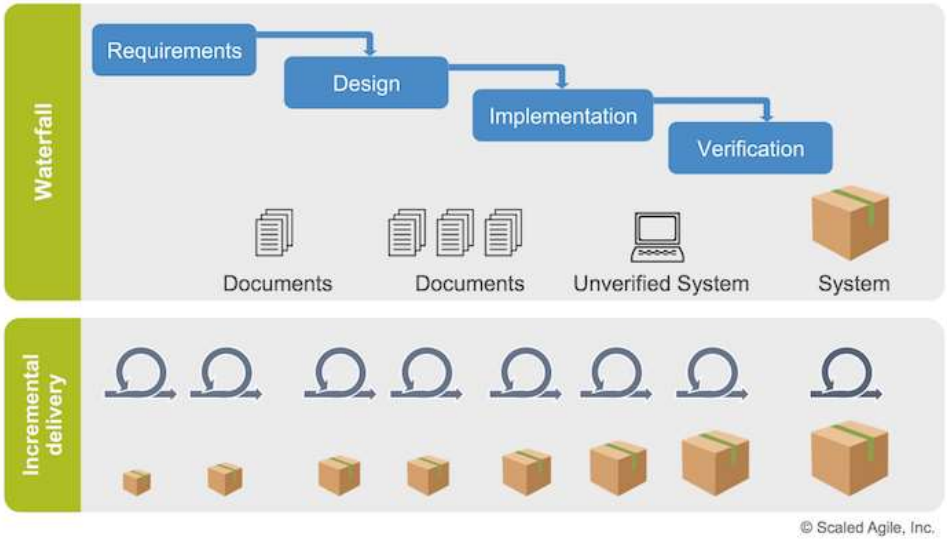
- **Alinhamento**: sempre possuir cadências onde é possível realizar o planejamento e reflexão em todos os níveis da organização para que todos saibam o atual estágio do negócio e assim sincronizar suas tarefas de modo regular.
- **Qualidade integrada**: a agilidade nunca deve prejudicar a qualidade, dessa forma todas as pessoas e equipes dentro da empresa precisam saber o que significa concluir uma tarefa com qualidade.
- **Transparência**: incentivar o comportamento de confiança entre as pessoas e assim planejar o trabalho com entregas menores para que os erros venham à tona mais cedo possível, deste modo garantir a qualidade e melhoria contínua.
- **Execução do programa**: equipes e programas precisam conseguir realizar entregas de qualidade e que geram valor para seu cliente de maneira regular.
- **Liderança**: incentivar a liderança ágil e enxuta, pois somente os líderes podem envolver todos os principais valores dentro do sistema.

Esses são os principais valores que o *Scaled Agile Framework* defende, além disso existem alguns princípios, são eles:

Levante uma visão econômica

Assim como no *Lean* é preciso compreender a necessidade para entregar o que é preciso no agora, e assim construir um ritmo sustentável de desenvolvimento. As empresas que decidem aplicar metodologias ágeis é porque já vivenciaram a metodologia cascata sem sucesso.

Figura 1. Modelo Cascata x Incremental



Fonte: Scaled Agile, Inc.

Na Figura 1 é possível visualizar a diferença entre o modelo cascata e metodologias que defendem entregas incrementais. Pequenas entregas facilitam e promovem a economia dentro do processo de desenvolvimento de um software. Além de sempre produzir entregas rápidas e de valor para o cliente.

Aplicar o pensamento de sistemas

Dentro do *SAFe* existe um processo holístico para facilitar a aplicar este princípio, que consiste em visualizar que:

Figura 2. Pensamento de sistemas fluxo holístico

Fonte: Scaled Agile, Inc.

A Figura 2 descreve o fluxo holístico proposto pelo *SAFe* para desenvolver um pensamento de sistema, com isso compreender que a solução é um sistema. A empresa que desenvolve

também é um sistema, assim entender e otimizar o fluxo de valor do desenvolvimento ajuda a compreender o quão completa é a solução.

Indicação de Leitura

Aplique o pensamento sistêmico. Disponível em: https://www.scaledagileframework.com/apply-systems-thinking/

Preserve as opções

O desenvolvimento de um software pode muitas vezes ser algo incerto e com isso é necessário preservar as opções de desenvolvimento, suas tarefas e ações. Levando em consideração o valor do alinhamento com o cliente, neste princípio é preciso conhecer as opções que devem ser tomadas, e escolher a que melhor se encaixa para cada etapa do desenvolvimento de um sistema.

Crie uma maneira incremental

Desenvolver um sistema de modo incremental permite que o retorno do usuário sobre o funcionamento do sistema seja mais rápido, ou seja, reduz o risco de grandes mudanças no futuro, além de melhorar o relacionamento entre empresa e cliente.

A cada entrega de um incremento o time é incentivado a refletir sobre essa entrega, a expandir seu conhecimento sobre o negócio e a parte técnica do sistema, desse modo nivelando o conhecimento entre todos.

Crie marcos base para avaliação

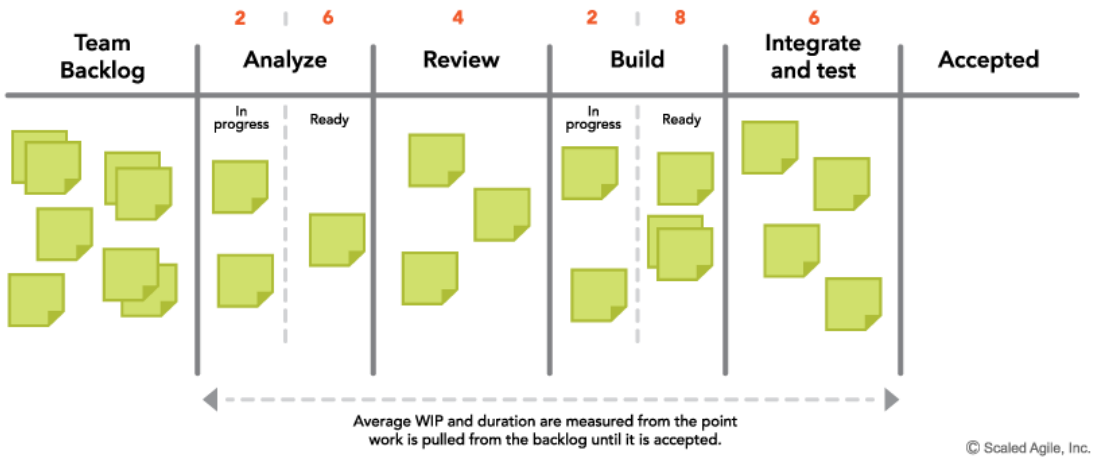
A proposta deste princípio é estabelecer pontos “marcos” durante o processo de desenvolvimento de software para avaliar as entregas e o sistema que está sendo construído, visando pensar como um sistema.

Pois cada pessoa é responsável por garantir que as novas decisões que vão ser tomadas gerem lucro para a empresa e qualidade para o sistema em desenvolvimento.

Possua um WIP limitado

Limitar o que está sendo desenvolvido é uma prática necessária segundo o *SAFe,* pois dessa forma o time consegue ter clareza da sua capacidade de desenvolvimento e isso pode impactar nos prazos de entregas.

Figura 3. Kanban com WIP



Fonte: Scaled Agile, Inc.

Como pode-se observar na Figura 3 para organizar e tornar transparente todo o limite de WIP, pode-se utilizar do quadro *Kanban* para melhor organizar as tarefas e entregas de acordo com a capacidade da equipe.

Aplique uma cadência planejada

Aplicar um ritmo dentro do processo de desenvolvimento cria previsibilidade e com isso reduz os riscos durante o desenvolvimento e com isso também é necessário sincronizar esse planejamento com todos da empresa.

Com a cadência correta e o sincronismo entre os times é possível operar o desenvolvimento do sistema mesmo com grandes incertezas dentro do projeto.

Indicação de Leitura

Aplique uma cadência planejada. Disponível em: https://www.scaledagileframework.com/apply-cadence-synchronize-with-cross-domain-planning/

Destrave a motivação íntima dos colaboradores com o conhecimento

Esse princípio consiste em demonstrar que a motivação necessária para que os colaboradores se sintam bem e construam o conhecimento não é somente por motivação monetária, mas sim por motivos íntimos de cada um.

É preciso tornar visível para cada pessoa do time que ela tem um propósito dentro da empresa e possibilitar a ele autonomia e respeito, com isso, cada membro do time se sente motivado e incentivado a buscar, inovação, ideação e engajamento no seu cotidiano

Descentralize a tomada de decisão

Para realizar entregas rápidas é preciso descentralizar as tomadas de decisão, essa atitude reduz atrasos, além de fortalecer o time. Pois possibilita a autonomia dos integrantes além de favorecer o *feedback* e melhorias que precisam ser aplicadas.

O desenvolvimento de um software não pode depender somente de uma pessoa, se a construção do sistema é construída pela equipe então a equipe deve tomar as decisões em conjunto também.

Organize em torno do valor

A maior vantagem das organizações é compreender as necessidades dos seus clientes e com isso gerar valor para eles, como as necessidades mudam rapidamente, ser ágil e se organizar por intermédio do valor que a solução tem, faz com que as empresas se tornem cada vez mais competitivas. Assim, este princípio defende que a empresa deve organizar o seu portfólio para prover valor aos seus clientes cotidianamente

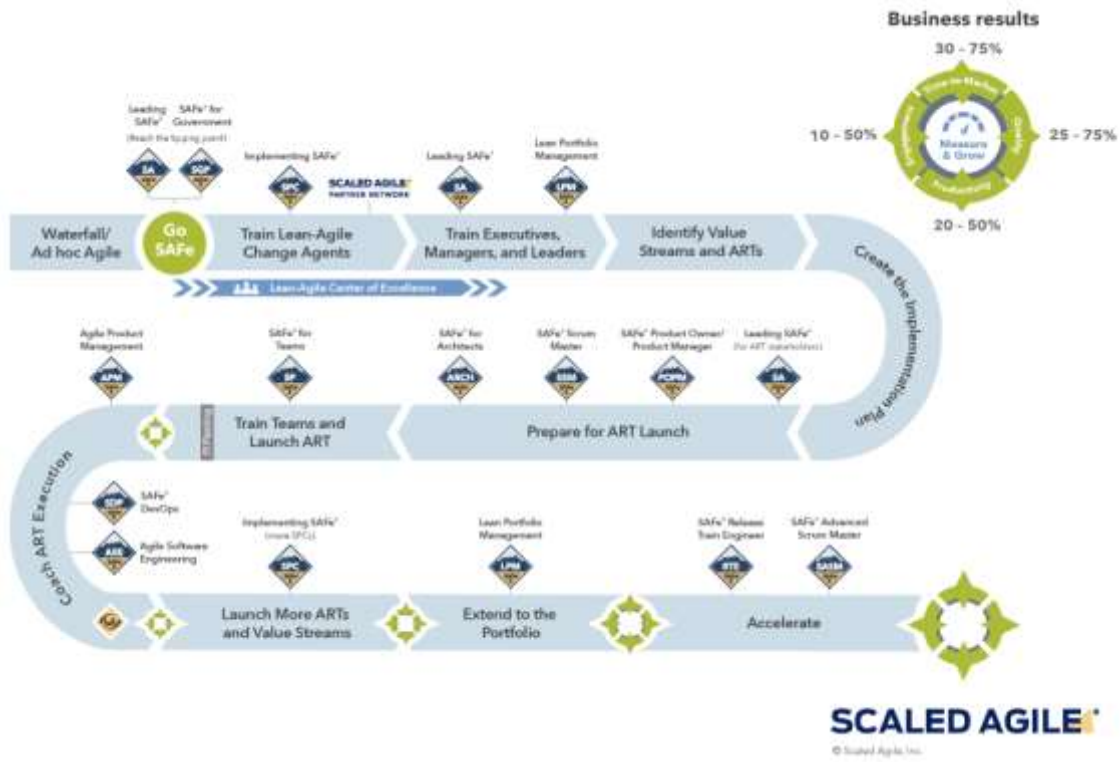
Videoaula 3

Utilize o QR Code para assistir!

Assista o vídeo para entender melhor o assunto.

Para facilitar a aplicação do *SAFe* existe um modelo de implementação para as empresas, após várias implementações diferentes a documentação do *SAFe* apresenta uma proposta de modelo que consiste em 12 passos para que uma organização consiga implementar e utilizar essa metodologia na construção de soluções.

Figura 4. Modelo para implementação SAFe



Fonte: Scaled Agile, Inc.

A Figura 4 relata de forma gráfica os passos de modelo para a implementação do *SAFe*, além de demonstrar pontos de organização econômica, princípios e valores. Os 12 passos são:

- Ponto de Virada
- Treinar agentes para o Ágil
- Treinar executivos, gerentes e líderes
- Criar um centro de excelência
- Identificar fluxos de valores
- Criar um plano de implementação
- Preparar plano de lançamento
- Treinar equipes e lançar
- Execução de times ágeis do Coach
- Lançar mais times ágeis e fluxos de valor
- Estender para o portfólio
- Acelerar

Indicação de Leitura

Roteiro de implementação do SAFe. Disponível em: https://www.scaledagileframework.com/implementation-roadmap/. Acesso em: 14 mar. 2022.

A aplicação do *SAFe* se relaciona com outras metodologias ágeis como o *XP*, *Kanban*, *Scrum*, *Lean* e outros, não se limita a somente uma delas, com isso, e seguindo o roteiro proposto para implementação dentro da empresa, é possível aplicar metodologias ágeis em grande escala e em todos os setores.

Scaled Agile Framework demonstra com seus valores e princípios que é possível aplicar o ágil em escala em qualquer empresa. Porém como demonstra o roteiro de implementação exige que esse processo seja realizado com cautela, para não afetar de maneira negativa os colaboradores e nem os clientes.

Nesta aula conversamos sobre o *SAFe - Scaled Agile Framework* e sua proposta de transformar empresas com o modo ágil de trabalhar e aplicando isso em escala, nos mais diversos ambientes e realidades organizacionais.

Nos encontramos na próxima aula para conhecermos mais a fundo essa realidade de metodologias ágeis.

Feature Driven Development - FDD

Olá, aluno, tudo bem? Nesta aula vamos conhecer melhor o *Feature Driven Development* (FDD), vamos descobrir juntos seus princípios e valores e como essa metodologia vem mudando a forma de desenvolvimento de software nos últimos 23 anos.

O *Feature Driven Development* (FDD), em português, Desenvolvimento Orientado a Funcionalidade, revolucionou o mercado de desenvolvimento de software, pois leva consigo a proposta de aplicar práticas ágeis em projetos grandes e complexos.

O FDD é umas das metodologias criadas antes mesmo do manifesto ágil existir, essa metodologia foi desenvolvida em 1997 durante a construção de um grande projeto na linguagem de programação JAVA para um banco em Singapura, nasceu com o contexto de análise e modelagem orientada a objetos do *Peter Coad* e com a experiência de gerenciamento de projetos de Jeff de Luca, mas só foi exposta para o restante do mundo em 1999 no livro "*Java Modeling in Color with UML*".

Após isso o FDD começou a ser amplamente utilizado em grandes projetos e que possuem alta complexidade de negócio para serem desenvolvidos. Diferentes de outras metodologias ágeis o FDD já nasce dentro de uma necessidade de desenvolvimento de um software.

Por ser uma metodologia ágil o *Feature Driven Development* possui características como:

- Desenvolvimento iterativo;
- Não exige um modelo específico de modelagem;
- Adaptativo;
- Suportar mudanças de requisitos e de necessidades do mercado;

Videoaula 1

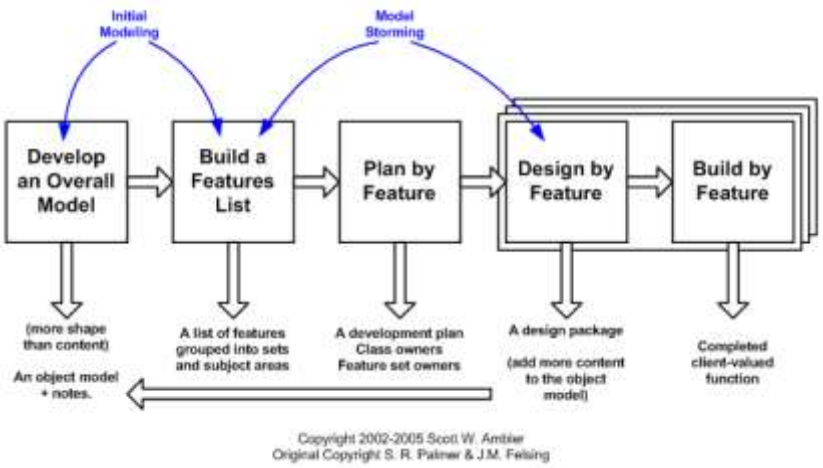
Utilize o QR Code para assistir!

Assista o vídeo para entender melhor o assunto.

Fases

Assim como outras metodologias, o FDD possui 2 fases que, quando aplicadas corretamente, auxiliam no bom desenvolvimento desta metodologia, são eles:

Figura 5. Fluxo total FDD



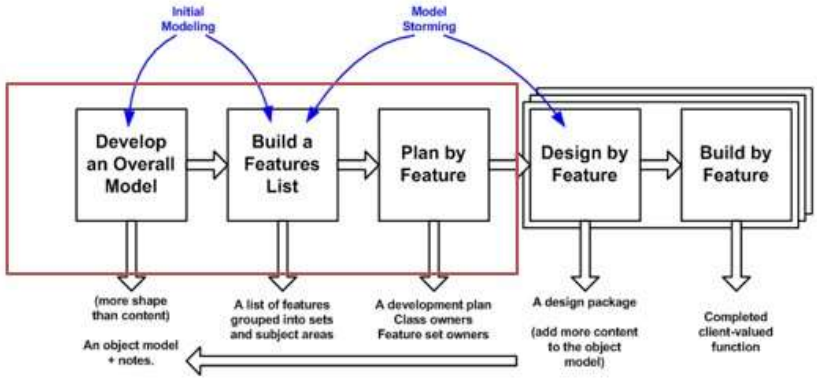
Fonte: Scott W. Ambler Original Copyright S. R. palmer & J.M. Felsing.

Na Figura 5 pode-se observar de modo gráfico o fluxo dos 5 principais processos do FDD e como eles interagem entre si, sempre focando nas funcionalidades que o time precisa desenvolver para entregar o software com sucesso.

No FDD existem somente 2 fases, sendo elas: planejamento e construção. Cada fase possui um fluxo de trabalho que quando seguido favorece o desenvolvimento, esse fluxo consiste no "ETVX":

- **Entrada**: define os critérios de entrada para as fases;
- **Tarefa**: lista as tarefas que precisam ser realizadas em cada fase;
- **Verificação**: específica os tipos de validação para cada fase (testes);
- **Saída**: especifica os critérios para concluir a fase.

Figura 6. Fluxo total FDD e suas Fases



Fonte: Scott W. Ambler Original Copyright S. R. palmer & J.M. Felsing (Adaptado)

Na Figura 6 observa-se em destaque as duas fases do FDD sendo dentro do retângulo vermelho a fase de planejamento e fora do retângulo vermelho a fase de construção, a fase de planejamento possui 3 processos e a fase de construção 2, sendo que para cada funcionalidade os processos da fase de construção são realizados novamente.

Videoaula 2

Utilize o QR Code para assistir!

Assista o vídeo para entender melhor o assunto.

Processos FDD

Desenvolver um modelo compreensível (*Develop an overall model*)

Neste processo é onde se reúne o time de especialistas de negócio e os programadores e então é realizado um estudo dirigido por domínio onde os especialistas de negócio apresentam as regras de negócio e a visão global do produto que precisa ser desenvolvido.

Após isso são criados pequenos grupos para que se possa discutir e criar um modelo compreensível do que precisa ser desenvolvido e de acordo com o que foi apresentado anteriormente pelo time de negócios. Depois, cada grupo apresenta a sua proposta de modelo e, é eleito um modelo compreensível do que vai ser realizado para solucionar a necessidade de negócio existente.

Vale lembrar que esse processo é realizado somente no início do desenvolvimento do projeto.

Construir uma lista de funcionalidades (*Build a features list*)

Esse processo consiste em reunir o time de desenvolvido onde alguns já participaram do processo anterior e expor o modelo escolhido para que desse modo seja criada a de funcionalidades de acordo com cada área de negócio que precisa ser entregue no final do desenvolvimento do software.

Planejar por funcionalidade (*Plan By Feature*)

Durante a execução desse processo é construído o time de planejamento, com isso esse time de planejamento é responsável por identificar as dependências do projeto, carga horária de desenvolvimento e complexidade das funcionalidades. Depois disso, são destinadas as funcionalidades para cada programador. Um programador pode ser “dono” de mais de uma funcionalidade.

Projetar por funcionalidade (*Design by feature*)

Neste ponto o líder técnico de cada time se reúne com os programadores que são “donos” das funcionalidades para então detalhar o que precisa ser realizado para então desenvolver uma funcionalidade.

No caso de *features* muito complexas pode-se buscar auxílio de especialistas de negócio, nesse processo não podem haver dúvidas do que precisa ser realizado na funcionalidade em questão.

Quanto maior o nível de abstração da funcionalidade, mais simples será aplicar o último processo, dessa forma podendo detalhar a construção de classes e arquitetura de desenvolvimento.

Construir por funcionalidade (*Build by feature*)

Esse processo consiste em observar o que foi desenhado na fase anterior e implementar no código, desenvolver a funcionalidade, olhando para o planejamento construído no processo anterior.

Cada desenvolvedor é responsável por avaliar aquilo que está desenvolvendo e também por testar e aplicar melhorias em seu código. Após a realização desse fluxo então pode-se entregar a funcionalidade.

Juntamente com os processos e fases a metodologia FDD possui também papeis onde cada integrante do time pode desempenhar um ou mais papeis de acordo com a necessidade e a funcionalidade que a equipe deve desenvolver.

Videoaula 3

Utilize o QR Code para assistir!

Assista o vídeo para entender melhor o assunto.

Papéis

Arquiteto chefe

É um integrante do time que possui mais experiência com a organização e modelagem de objetos, deve servir como guia para os outros desenvolvedores, em outras metodologias é considerado o líder técnico do time.

Programador chefe

É responsável por organizar o time com suas tarefas e tratar como serão implementadas as funcionalidades, a ordem de entrega, além de fazer a integração com o gerente de projetos e o gerente de desenvolvimento.

Gerente de projeto

Realiza o gerenciamento de todo o projeto, recursos, tempo, risco e organiza juntamente com o programador chefe e o gerente de desenvolvimento ordem de entregas para cada funcionalidade

Gerente de desenvolvimento

É responsável por gerir o time de desenvolvedores, auxiliando cada um na sua necessidade para que tudo ocorra bem durante a construção do projeto, além de auxiliar o programador chefe e o gerente de projetos.

Especialista de domínio

Este é o responsável por prover para toda a equipe as informações necessárias sobre para área de domínio, é aquele que deve possuir conhecimento do negócio para explicar para o time o que precisa ser construído, seu conhecimento é essencial para que todos os processos do FDD ocorram da maneira correta.

Programador

Este é o responsável por todo o fluxo de codificação do software, ele é quem diagrama, testa e documenta todas as funcionalidades que lhe foram atribuídas pelo programador chefe.

Esses são os papéis que o FDD apresenta dentro da sua metodologia, claro que assim com outros métodos ágeis é possível adaptar tanto os processos, quanto os papéis para assim entregar valor para os clientes.

Indicação de Vídeo:

FDD - Feature Driven Development. Disponível em: https://www.youtube.com/watch?v=pzdbmE5fcnM. Acesso em: 14 mar. 2022.

Como as funcionalidades são listadas e realizadas iterações para entregar um pacote de funcionalidades é muito importante estabelecer regras e tempos para poder organizar melhor o fluxo de desenvolvimento, bem como alinhar com o cliente como irá funcionar as iterações.

Iterações devem possuir uma cadência entre 2 até 4 semanas, desse modo se uma funcionalidade demora mais que esse tempo para ser desenvolvida, isso indica que essa funcionalidade deve ser quebrada em mais partes para assim facilitar a entregar e mitigar riscos futuros.

Desenvolver grandes funcionalidades e entregar todas de uma vez é um sinal de alerta, se isso ocorre dentro de uma empresa é porque a aplicação do FDD não está sendo efetiva, podendo acarretar grandes melhorias futuras e dificuldade na manutenção do software.

Algumas vantagens da aplicação do FDD:

- Entregas frequentes;
- Suporte para grandes projetos;
- Planejamento e Modelo adaptáveis;
- Entrega valor para o cliente de modo visível;
- Permite a manutenção rápida por meio de funcionalidades novas;
- Fases com validação, assim garante qualidade.

É possível aplicar o FDD em conjunto com outras metodologias, quando isso ocorre torna-se mais fácil a sua aplicação, o FDD não é a salvação para o projeto, ou seja, aplicar ele não garante que os problemas não ocorrerão, mas aumentam as chances de fazer entregas com qualidade e de possuir um desenvolvimento organizado e bem praticado.

Chegamos ao fim de mais uma aula, espero que os conhecimentos obtidos sobre o *Feature Driven Development* possam lhe auxiliar em sua caminhada de descobertas no mundo ágil do desenvolvimento de software.

Encerramento

Chegamos ao final da nossa terceira unidade onde você teve contato com os primeiros conceitos e ferramentas do *FDD* e *SAFe*. Com certeza muito do que foi mostrado aqui você já tinha uma pequena noção. Espero que você, aluno, tenha aprendido um pouco mais sobre metodologias ágeis.

Na aula 1, conhecemos a metodologia *SAFe*, ao qual a proposta é desenvolver o ágil em escala dentro de grandes organizações e assim podendo ser aplicado dentro de qualquer empresa que deseja melhorar o seu desempenho, seja ela de tecnologia ou não.

Na aula 2, foi discutido e apresentado a metodologia *FDD* um método ágil fortemente utilizado no mercado de tecnologia, também conhecemos suas fases, processos e papéis e como esses podem auxiliar no desenvolvimento de um software.

Juntando as duas aulas obtemos descobertas interessantes sobre esses métodos que estão sendo aplicados no mercado de desenvolvimento de softwares. Caro aluno, reflita um pouco como os conhecimentos desta unidade podem te ajudar nas suas atividades e como o material apresentado pode ser aplicado, caso você já aplique alguma das metodologias desta unidade, pense naquilo que você pode melhorar. Até a próxima.

Referências

PRESSMAN, Roger S. **Engenharia de Software**. 1. ed. São Paulo: Person, 2011.

SCALED AGILE. **INC**. Disponível em: https://www.scaledagileframework.com/. Acesso em: 22 de out. de 2021.

FDD. Disponível em: http://www.featuredrivendevelopment.com. Acesso em: 22 de out. de 2021.


UniFil
EDUCAÇÃO A DISTÂNCIA
UNIFIL.BR