

TRANSFORMAÇÃO DIGITAL EM MULTI-X

Prof. Renato Vertemati

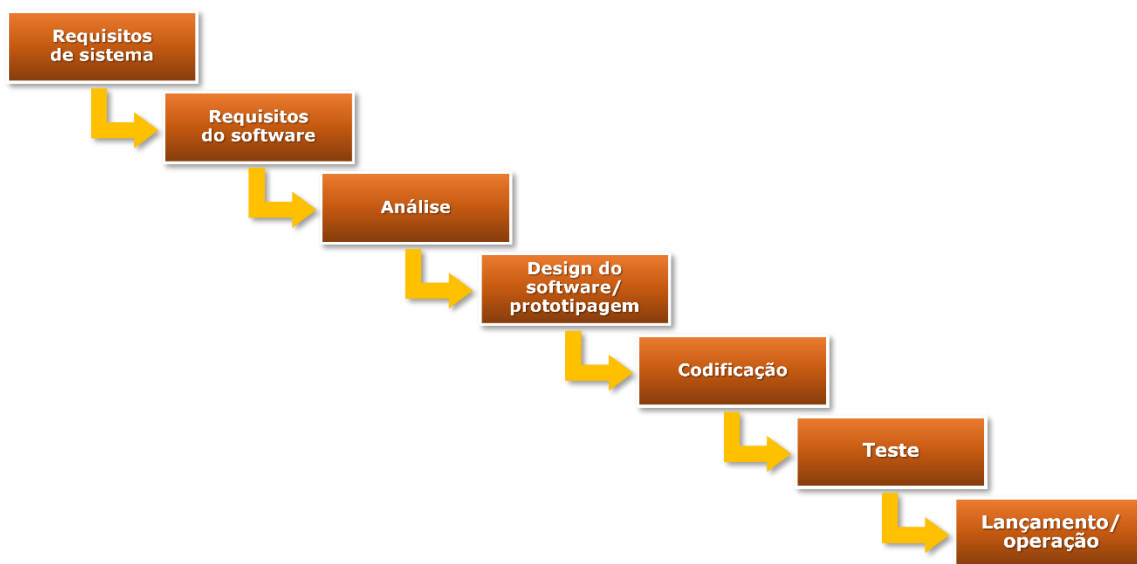
INTRODUÇÃO

Muito se ouve sobre gestão e cultura ágil. É uma forma de conduzir projetos com alta produtividade e que está conquistando cada vez mais empresas, transformando culturas e demandando uma adaptação rápida das pessoas. Seja para entregas complexas, seja somente para ajustar o dia a dia e torná-lo mais produtivo, é importante dominarmos o assunto.

TEMA 1 – A CULTURA ÁGIL: HISTÓRIA

Todo modo novo de pensar é desafiador. E antes de assimilar novas formas de conduzir nossas ações, é interessante conhecer a lógica e a história por trás.

Tudo começou em 1970, quando Winston Royce, renomado cientista de computação estadunidense, publicou o paper “*Managing the development of large software systems*”, ou “Gerenciando o desenvolvimento de sistemas de *software* complexos”. Neste documento, Royce critica o processo sequencial de desenvolvimento que, mais tarde, recebeu o apelido de *modelo waterfall*, ou cascata, devido à sua representação visual.



Mas, por algum motivo, em vez de ficar conhecido como um crítico do método tradicional, Royce foi intitulado como o criador do modelo. Muitos artigos na internet reforçam este curioso e equivocado crédito.

Saiba mais

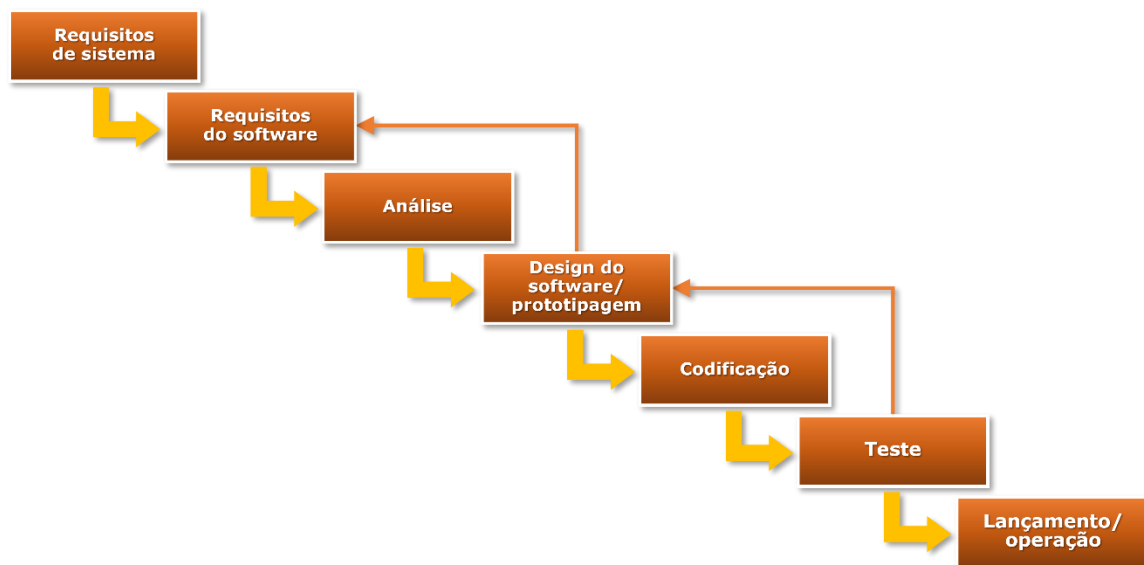
Disponível em: <<http://agile.pub/assuntos-diversos/a-grande-mentira-do-waterfall/>>. Acesso em: 7 jul. 2022.

O motivo de Royce figurar na história como um precursor do modelo ágil foi o teor de suas críticas ao modelo sequencial:

- o design do programa vem antes;
- documente o design;
- faça 2x;
- planeje, controle, teste;
- envolva o usuário.

Destes 5 tópicos, os 3 últimos estão intimamente ligados ao manifesto ágil, criado 30 anos depois e que veremos a seguir, principalmente “Envolva o usuário”.

O modelo de Royce pode ser chamado de uma evolução baseada em iterações, que podem ser interpretados como “loops” na história do desenvolvimento para estressar toda a cadeia do desenvolvimento quantas vezes for necessário e viável para se obter o melhor resultado:



Observe que, conforme o resultado dos testes, o fluxo retorna para o *design* do *software* que, por sua vez, pode retornar até o início, quando os primeiros requisitos são definidos.

Atualizando os conceitos, Royce desenhou de certa forma o fluxo de uma *Sprint*, termo amplamente utilizado hoje no universo ágil.

1.1 O conceito de Iteração

- A divisão da linha do tempo em blocos menores. Essa é uma ideia central do ágil. Um sistema nunca é desenvolvido de uma vez só, como um só bloco. Em vez disso, seu escopo é fragmentado em entregas que possam,

simultaneamente, ter um valor isolado para o cliente, e sua entrega não necessita de um tempo muito extenso para a entrega isolada.

- Cada bloco é desenvolvido rapidamente para que possa ser testado e aprimorado antes da entrega. Royce escreveu *faça 2 vezes*. Para que uma iteração tenha tempo de ser avaliada e, caso necessária, toda refeita, é preciso que seu tempo de desenvolvimento seja o menor possível. Caso contrário, não haverá tempo para estressar o seu resultado e refiná-lo até o estado ideal.
- Toda iteração tem início, meio e fim. Como dito anteriormente, é preciso haver valor para o cliente em cada iteração. Valor não pode ser construído com entregas parciais.
- Iterações também são conhecidas como *Sprints* no *framework* de desenvolvimento Scrum.

Dessa forma, como se crê entre os seguidores do método ágil, o valor gerado em cada Sprint, somado às demais, resulta em muito mais valor para o cliente que um *software* pensado e desenvolvido de uma só vez. Ou seja, podemos dizer que “a soma das partes é sempre maior que o todo”.

TEMA 2 – O MANIFESTO ÁGIL

Pedra fundamental de todo desdobramento, o manifesto foi produto de 17 mentes diferenciadas que compartilhavam de uma mesma forma de pensar sobre o desenvolvimento de sistemas. Estes desenvolvedores e engenheiros de sistemas se reuniram em 2001 e escreveram o manifesto que foi traduzido para 68 idiomas e atualmente é a origem de dezenas de novas metodologias.

Abaixo, o nome dos 17 envolvidos na criação do manifesto:

- Arie van Bennekum;
- Dave Thomas;
- Jeff Sutherland;
- Steve Mellor;
- Robert C. Martin;
- Brian Marick;
- Jon Kern;
- Ron Jeffries;
- Andrew Hunt;

-
- Jim High;
 - Martin Fowler;
 - Alistair Cockburn;
 - James Grenning;
 - Kent Beck;
 - Mike Beedle;
 - Ken Schwaber;
 - Ward Cunningham.

O manifesto prega valores que, bem interpretados, podem servir de base não só para o desenvolvimento de sistemas, mas para a vida!

- Indivíduos e interações ACIMA de processos e ferramentas. O método ágil prega o contato frequente entre os membros de uma equipe como forma de aprimorar o resultado final. A colaboração não só entre desenvolvedores, mas também entre as outras especialidades que compõem o produto final: *Product Owners*, *designers*, testadores etc.
- *Software* em funcionamento ACIMA de documentação abrangente. Contrário a muitos gerentes de projeto, o texto sugere que a energia do time é melhor empregada na qualidade das discussões e do produto entregue do que a documentação nos mínimos detalhes. Observemos que não é sobre ausência de documentação que o tema fala, mas sim na racionalização dos recursos empregados nesta parte do projeto que raramente tem conexão direta ao valor da solução. Trocando em miúdos, de nada vale uma documentação enorme e detalhada sobre uma solução de má qualidade.
- Colaboração com o cliente ACIMA de negociação de contratos. Contrato é importante, principalmente quando algo dá errado. Mas um verdadeiro parceiro não se garante por contrato, e sim por conversas produtivas em um ambiente colaborativo. Cliente e fornecedor, na visão ágil, devem ser parceiros durante todo o projeto.
- Responder a mudanças ACIMA de seguir um plano. Outro pilar importante do pensamento ágil. Tendo uma visão clara de onde se quer chegar e quais problemas devem ser resolvidos, o restante é melhor resolvido com flexibilidade do que com rigidez para seguir um plano predefinido. As melhores soluções podem vir na etapa final de um projeto, e a equipe a ele

dedicada deve ter jogo de cintura para assimilar mudanças de rota a qualquer instante.

2.1 Os 12 princípios do manifesto ágil

A seguir, vejamos a estruturação do manifesto em seus 12 princípios.

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

Este princípio prega sobre o que toda entrega profissional deve ter, independentemente de ser um *software*: valor. Não é à toa que este é o primeiro princípio. Todo o manifesto ágil surgiu ao redor do cliente.

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

Pensando na gestão de recursos humanos, é um princípio mais difícil de conduzir. Estar disposto a alterar tudo o que foi feito na última milha de um projeto não é para qualquer um, mas pode garantir a competitividade de um projeto assim que seja lançado no mercado. O princípio seguinte tende a minimizar os efeitos de uma mudança tardia nos requisitos.

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

A entrega frequente pode trabalhar como um antídoto para o estresse potencial de mudanças frequentes no escopo. Se o ciclo de entrega de um módulo de *software* é curto, o tempo não age contra a qualidade do *software* (tecnologia deprecia a cada minuto). A política de entregas em ciclos curtos também é um excelente recurso para redução da ansiedade que um investimento pode gerar.

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

Times multifuncionais, discussões mais ricas e trabalho mais inteligente. Deste princípio se originaram algumas funções em times ágeis que têm a missão de consolidar conhecimento e facilitar a comunicação, como o *Product Owner*.

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

Um mantra da gestão moderna. Excelência em produto parte de profissionais excelentes. Todo potencial de um profissional pode se desenvolver se o ambiente o permitir.

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

Seja em um trabalho presencial, seja via videoconferência, o contato frequente sempre é sadio em qualquer equipe. Para isso, é preciso ter uma cultura de comunicação aberta, cordialidade e altruísmo. Enganam-se os que pensam que times de alto desempenho são compostos por pessoas focadas 100% do tempo em suas próprias tarefas.

- 7) *Software* funcionando é a medida primária de progresso.

Podemos trocar “*software* funcionando” por “entrega de valor”. Esse princípio é uma vacina contra a mediocridade que existe em diversos mercados, com entregas de baixo ou nenhum valor “empacotadas” em discursos eficientes e embalagens chamativas. Também são chamadas de *métricas de vaidade* e, definitivamente, não mudam o mundo para melhor.

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

Por ritmo constante somado à sustentabilidade, devemos entender que a carga de trabalho deve ser controlada diariamente. Picos eventuais acontecem e não prejudicam o andamento de um projeto, desde que de fato eventuais. Não há ambiente de trabalho saudável nem portando entrega de qualidade numa rotina de sobrecarga contínua.

- 9) Contínua atenção à excelência técnica e bom *design* aumenta a agilidade.

Fazer bem-feito pode tomar um pouco mais de tempo, mas certamente é mais rápido do que refazer algo ruim. Além da perda de tempo, os *bugs* e problemas de *design* geram estresse entre fornecedor e cliente, desmotivam o time e colocam muitos projetos em descrédito.

- 10) Simplicidade — a arte de maximizar a quantidade de trabalho não realizado — é essencial.

O texto pode não ser tão intuitivo, mas trata-se de valorizar o tempo economizado por trabalhos que ninguém quer fazer — uma semana, por exemplo,

passada resolvendo um débito técnico de um projeto. *Features* sem valor também integram esse grupo de “trabalhos que ninguém deveria realizar”.

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

Gestores, futuros gestores, prestem muita atenção neste princípio que deve soar como um mantra de equipes de alto desempenho. Este princípio deve estar em sua mente desde a contratação. Times maduros, talentosos e conscientes de suas missões não precisam de constantes “puxões de orelha” ou acertos de rota. Líderes centralizadores nunca serão bons líderes de times ágeis.

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

Melhoria contínua. Todo processo tende a se tornar ineficiente com o passar do tempo. É saudável que as estruturas sejam provocadas de tempos em tempos para manutenção da motivação e da competitividade.

TEMA 3 – MODERNIZAÇÃO DO ÁGIL E A ADAPTAÇÃO AO MUNDO DOS NEGÓCIOS

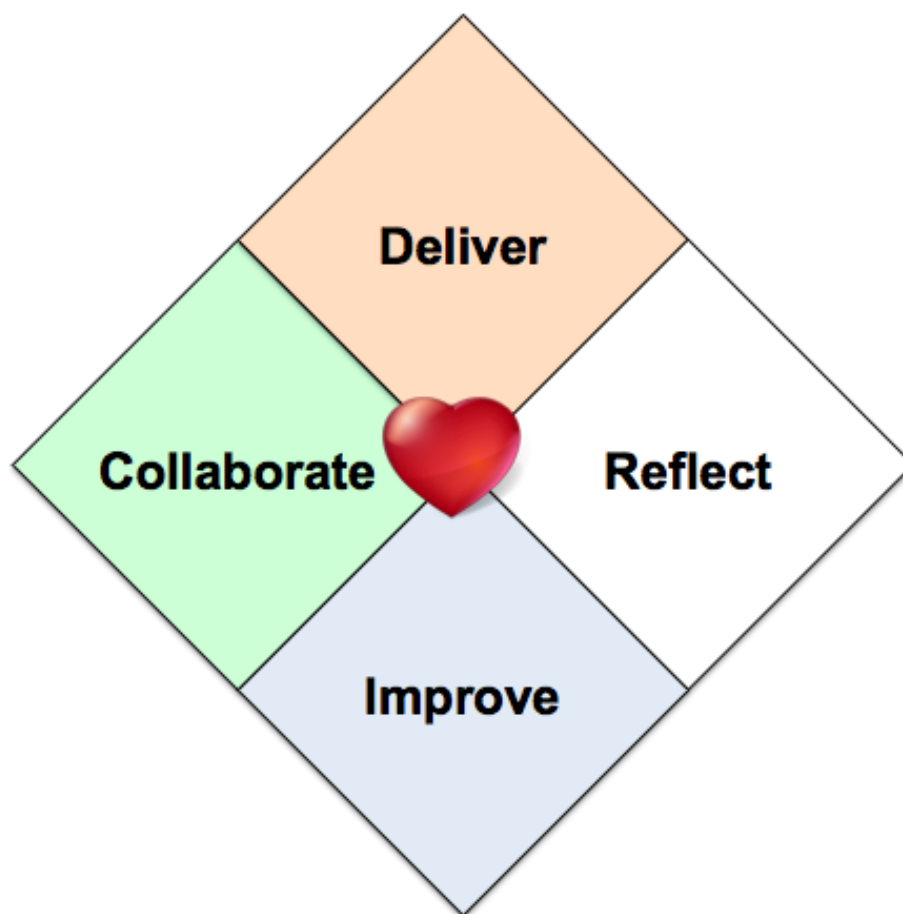
O próprio manifesto pode ser interpretado como a principal razão do surgimento de desdobramentos mais atuais, por pregar a melhoria contínua. Afinal, mais de 20 anos se passaram e não há como negar que temos um mundo mais complexo tecnologicamente e também a necessidade de poder aplicar bons princípios de gestão de projetos em empresas que não entregam *software* como produto final.

Sendo assim, temos hoje uma miríade de desdobramentos em diversos formatos, dos mais formais *frameworks* como o Scrum até formatos adaptados a corporações específicas como o Spotify Model e o TPS (*Toyota Production System*).

Trouxemos nesta etapa duas releituras bastante interessantes do manifesto ágil e que podem ser carregadas para dentro de sua empresa, independentemente do produto ou serviço ofertado. Não são *frameworks* complexos e formais, mas sim formas mais leves e simples de gerenciar projetos de forma ágil. Adicionalmente, veremos uma metodologia que muitos conhecem pelo visual ou por um famoso *software*, mas que vale repassar para que seja corretamente aplicada.

3.1 Heart of agile

A proposta do “Coração do Ágil” é reduzir o manifesto a uma essência mais simples e “democrática”. Seu criador dispensa apresentações: Alistair Cockburn é um dos 17 signatários do Manifesto Ágil e seu propósito com o *Heart of Agile* foi o de levar a essência do manifesto a times de todo tamanho e finalidade. São 4 “palavras fortes” que sintetizam a aplicação da essência ágil nas empresas e que têm um propósito mais educacional que gerencial, como é o caso do manifesto em si. Podemos entender que esta iniciativa do Dr. Cockburn é mais voltada ao mercado de treinamento corporativo que de gestão de projetos em si.



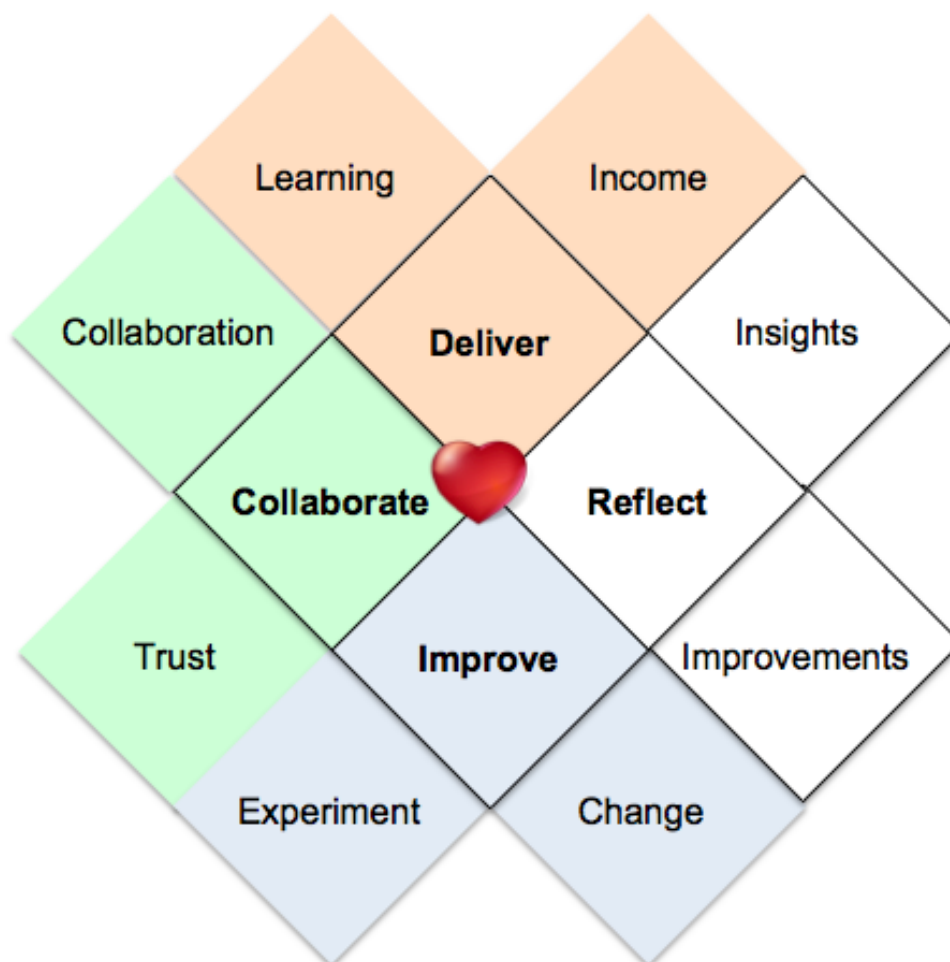
Fonte: <http://heartofagile.com/>

- Colaborar: a meta é ter um time que se comunica bem e frequentemente, com indivíduos que confiam uns nos outros. Colaboração é consequência.
- Entregar: inicie em pequenas entregas que geram aprendizado para entregas maiores e de maior valor, a cada “rodada”.
- Refletir: “quem só trabalha não tem tempo para ganhar dinheiro”. Esse ditado popular, apesar de bastante duvidoso, é uma forma de explicar essa

palavra do coração do ágil. É preciso de pausas rotineiras para reflexão sobre o que já foi entregue e os próximos movimentos. Só com reflexão e análise é possível aprimorar as entregas de um time de tempos em tempos. Do contrário, a rotina criará a dinâmica do “mais do mesmo” eternamente.

- Melhorar: aplicar os aprendizados, experimentar, mudar sempre que necessário. A quarta palavra pode funcionar como uma síntese objetiva do propósito do coração.

Consultores costumam trabalhar este método pensando nos possíveis desdobramentos de cada uma das 4 palavras, o que torna o modelo um pouco mais rico. A imagem abaixo, retirada do próprio site heartofagile.com, ilustra um primeiro nível de desdobramento:



Fonte: <http://heartofagile.com/>

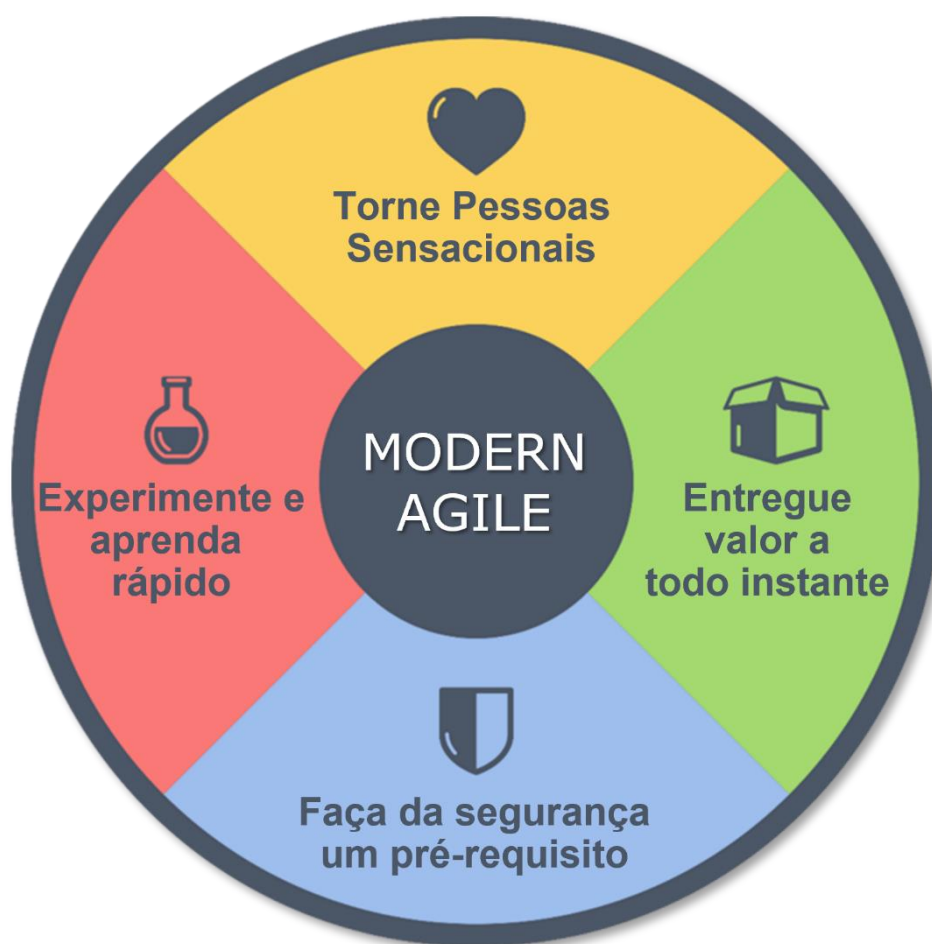
3.2 Modern Agile

Criado em 2016 por Joshua Kerievsky, nome também bastante conhecido na cultura ágil e autor de diversos livros. Trabalha desde os anos 90 focado em

métodos de desenvolvimento mais “leves” e ágeis. Diferente do *Heart of Agile*, o *Modern Agile* tem uma forma de distribuição mais livre e próxima dos desenvolvedores. Em seu site oficial¹, o visitante pode baixar materiais para imprimir, aprender todos os princípios do método e colocá-los em prática. É um método que recomendo a iniciantes que não precisam (ainda, pelo menos) de um método mais robusto para desenvolvimentos complexos. Seja pela acessibilidade e suporte disponíveis, seja pela facilidade de implantação.

O método, assim como o *Heart of Agile*, tem seu resumo visual no formato de uma roda de quatro setores:

Figura 1 – Método



Fonte: <http://heartofagile.com/>

- Torne as pessoas sensacionais: é o objetivo mais amplo do *Modern Agile*. Pessoas, aqui, são todos os envolvidos no ecossistema do projeto: quem pensa, desenvolve, vende, compra ou usa. É necessário ouvi-los,

¹ Disponível em: <<https://modernagile.org/>>. Acesso em: 11 jul. 2022.

compreendê-los, entender os problemas e dores e aí pensar em formas de tornar todos sensacionais. Vejo mais como um “espírito” que tarefa, dada a enorme complexidade que se esconde atrás de uma frase como essa. Gestão de pessoas, requisitos de *software*, pesquisas, conversas informais de corredor, tudo cabe aqui.

- Faça da segurança um pré-requisito: pode parecer, mas esse tópico não fala de segurança de informação. Como no primeiro tópico, ele também é bastante amplo. Segurança de uma equipe pode envolver respeito (ao tempo e condições de trabalho), informações pessoais, vida financeira, saúde e relacionamentos interpessoais. É como em um ambiente de trabalho perigoso: antes de enviar um funcionário para executar tarefas, é necessário equipá-lo com o básico da segurança para que ele possa desempenhar suas funções da melhor forma possível.
- Experimente e aprenda rápido: um tópico que funciona como condicional dos dois primeiros. Não há como tornar pessoas sensacionais nem fazer da segurança um pré-requisito se não estivermos aprendendo com o trabalho. E não há como aprender sem experimentar. O ambiente de projeto determinado por esta filosofia de gestão é aquele onde experimentar, falhar e retomar atividades aprimoradas é altamente recomendado. Mas não esqueçamos de uma condição para que este tópico não signifique atrasos: a velocidade do ciclo da experimentação, aprendizado, retomada e melhoria deve ser sempre alta.
- Entregue valor a todo instante: eu poderia traduzir este tópico com uma frase que adoro: “antes feito que perfeito”. Mas isso limitaria o significado deste setor da roda do *Modern Agile*. O “feito” significa valor. E “antes” é antes mesmo, contínua e frequentemente. É nesse aspecto que se encontra uma das pedras fundamentais do pensamento ágil: dividir grandes blocos de entrega em peças menores, cada qual com valor para os clientes.

3.3 Kanban

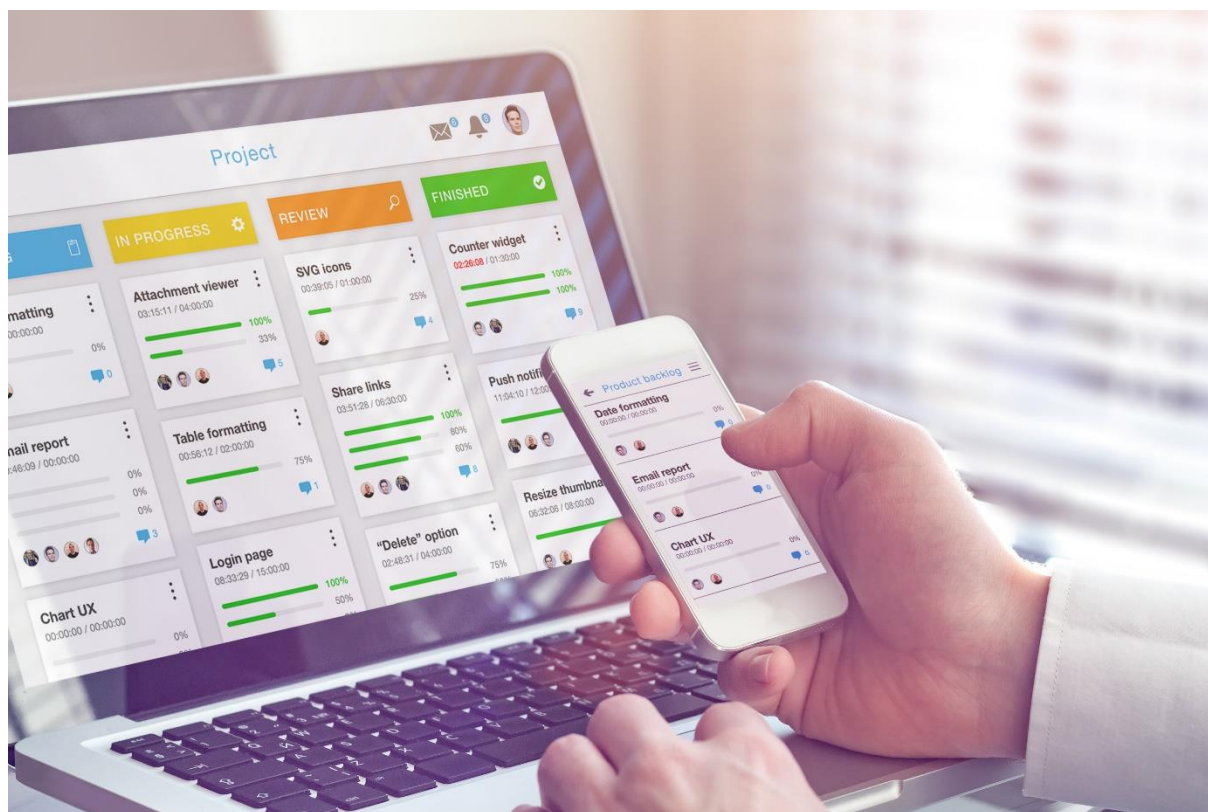
Originado de um termo japonês que significa “tabuleta” ou um “painel visual”. Sua criação antecede o manifesto ágil em diversas décadas, mas devido aos seus aspectos básicos, foi introduzido ao ágil. Com sua utilização crescente, o Kanban originou *softwares* famosos utilizados em diversas empresas, como

Trello, Jira, Pipefy e Pipedrive, que atendem plenamente desde departamentos de marketing e vendas até oficinas mecânicas.

O *Kanban board* básico é composto de colunas, que significam etapas de trabalho, e cartões (*cards*), que são as tarefas a serem executadas. Essas etapas têm um ponto de partida: sempre há uma etapa de entrada, onde as tarefas são criadas, uma etapa para quando as tarefas estão em andamento e uma final para entrega. Toda tarefa tem um responsável pela execução e, preferencialmente, um prazo para conclusão. Em sistemas como o Trello®, os *cards* podem ser alimentados com as mais diversas informações, como arquivos, imagens, informações pré-selecionáveis etc.

Abaixo, temos um exemplo de um sistema de *Kanban*.

Figura 2 – Sistema de *Kanban*



Créditos: NicoElNino/Shutterstock.

O *Kanban*, devido à sua simplicidade, pode ser utilizado em ambientes onde não há sistemas de computador para acesso dos colaboradores. Foi assim, inclusive, que ele surgiu:

Figura 3 – *Kanban*



Créditos: Olya Maximenko/Shutterstock

Para tirar máximo proveito do *Kanban*, é recomendável que se sigam alguns princípios básicos:

- Colaboradores precisam ter a visualização das próximas tarefas a qualquer momento. Dessa forma, conseguem organizar a própria fila de entregas (lembramos que equipes auto-organizáveis é um dos 12 princípios ágeis).
- Controlar/limitar a quantidade de tarefas na fila, por colaborador. Dessa forma, teremos uma segurança adicional para que a equipe não se sobrecarregue.
- Acompanhar as tarefas que estiverem em vias de serem entregues, para avaliar a fila das próximas tarefas a serem executadas. Dessa forma, garante-se a entrega contínua.
- *Cards* não “dão ré”. Refações e mudanças de rota são sempre novas tarefas. Dessa forma, as métricas de produtividade não são prejudicadas (uma das principais métricas do *Kanban* é o tempo que cada *card* fica em uma etapa”).

TEMA 4 – A ADAPTAÇÃO DO ÁGIL AO DESIGN/UX

A metodologia ágil permaneceu por alguns anos restrita ao mundo do desenvolvimento, mas aos poucos foi sendo adaptada para a fase de *design* dos produtos e serviços. O motivo é simples: velocidade e qualidade de entrega são benefícios extremos também para esta fase, individualmente. Nem o melhor desenvolvimento pode salvar um mau *design*. Assim como um bom *design* não resistirá a maus desenvolvedores.

Como em toda adaptação, existem algumas novas formas de se aproveitar o conteúdo e os ensinamentos não do manifesto original isoladamente, mas de desdobramentos que juntos provaram ter mais valor à disciplina de UX. Vejamos a seguir quais são:

4.1 Agile UX

Adaptação direta do ágil à UX, trabalhando alguns dos seus princípios:

- Times multifuncionais: *designers* e desenvolvedores, principalmente. A colaboração entre designers e desenvolvedores gera propostas muito mais consistentes e factíveis de serem construídas. Interações entre os membros do time são mais importantes do que os processos e ferramentas.
- Envolver o consumidor no processo é extremamente importante para colher *feedback* o quanto antes.
- Construir o *software* é mais importante do que produzir documentação super extensa.
- Iterações bem definidas e em intervalos curtos: o grande segredo do ágil e uma grande ruptura para boa parte das equipes de *design*, que historicamente trabalham o produto todo numa mesma etapa.
- Saber se adaptar à mudança ao invés de seguir um plano rígido. Mais um desafio de gestão de equipes de design, que frequentemente se entendem detentoras de algumas “verdades” difíceis de serem alteradas.

4.2 Lean

O *Lean* não é exatamente um dos desdobramentos do manifesto ágil, mas compreende princípios dos quais muitos coincidem com o ágil. A palavra-chave é *otimização pela eliminação dos excessos*.

Por este motivo, vem sendo muito usado no mundo dos negócios cujo sucesso está intimamente ligado à otimização: de recursos, custos e principalmente receita. Afinal, como o enxadrista Garry Kasparov já disse: “ao final do dia, o que importa é o dinheiro”.

O *Lean* também inspirou o empreendedor de tecnologia Eric Ries a escrever o livro *The Lean Startup*, que virou febre em gestão de negócios a partir de seu lançamento, em 2011.

Saiba mais

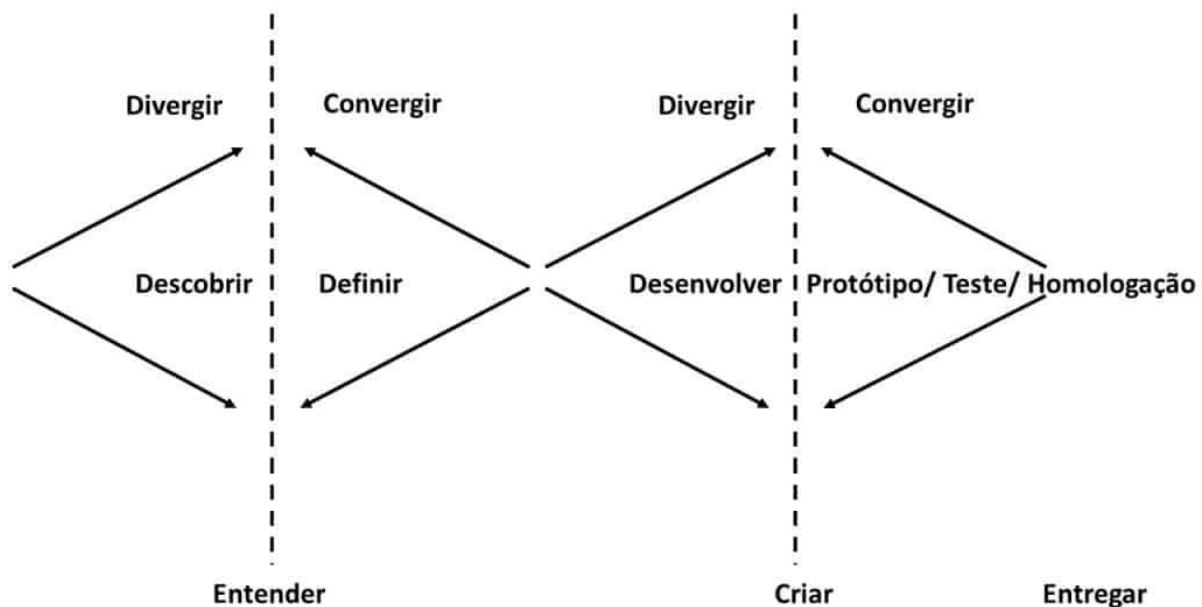
Saiba mais sobre Garry Kasparov. Disponível em: https://en.wikipedia.org/wiki/Garry_Kasparov. Acesso em: 7 jul. 2022.

Na Lean UX, deve-se:

- construir os MVPs (*Minimum Viable Products*) para resolver os problemas do usuário;
- prototipar rapidamente e colaborativamente;
- validar hipóteses com os usuários o quanto antes;
- usar métricas e outros insights para validar as ideias e refiná-las com o passar do tempo.

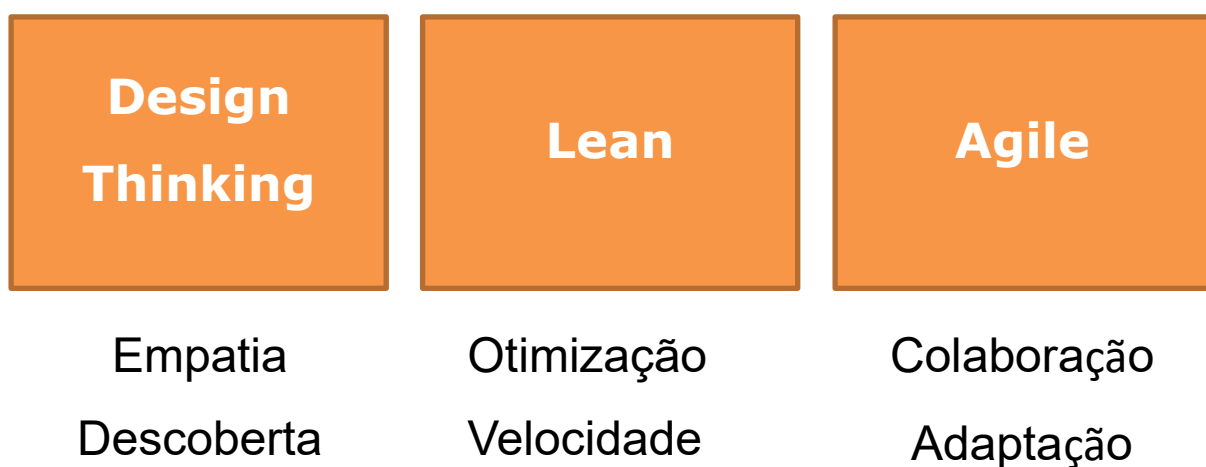
4.2 Design thinking

O DT entra como um complemento ao casamento Agile UX + Lean UX. É um excelente método para as etapas iniciais dos projetos, onde ficam as descobertas dos problemas dos usuários e os levantamentos das hipóteses de solução. O primeiro do “duplo diamante” é onde fica a maior contribuição para a lógica combinada do ágil + UX.

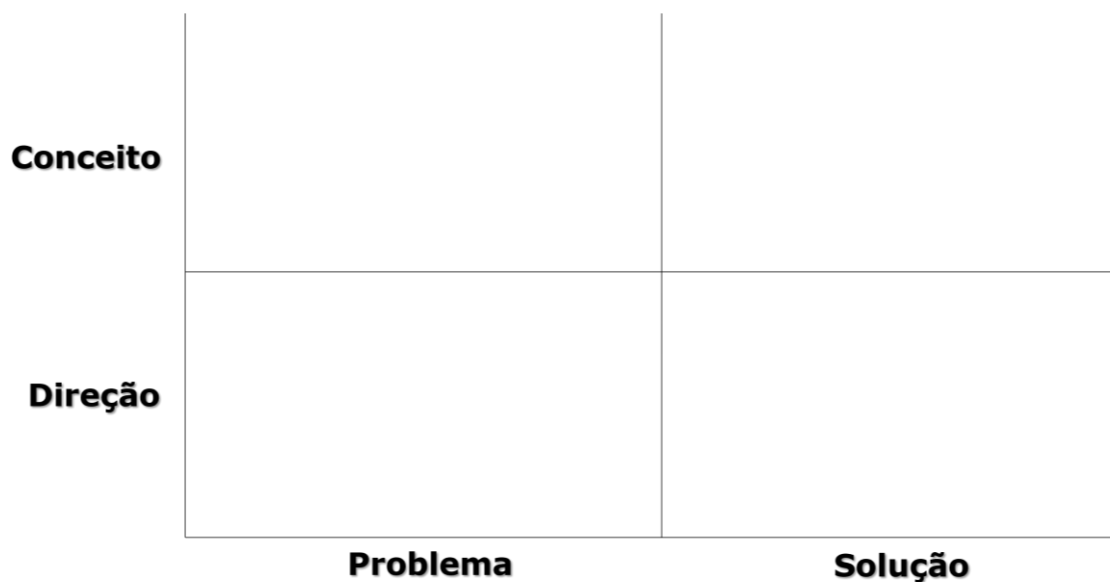


TEMA 5 – O MÉTODO COMBINADO

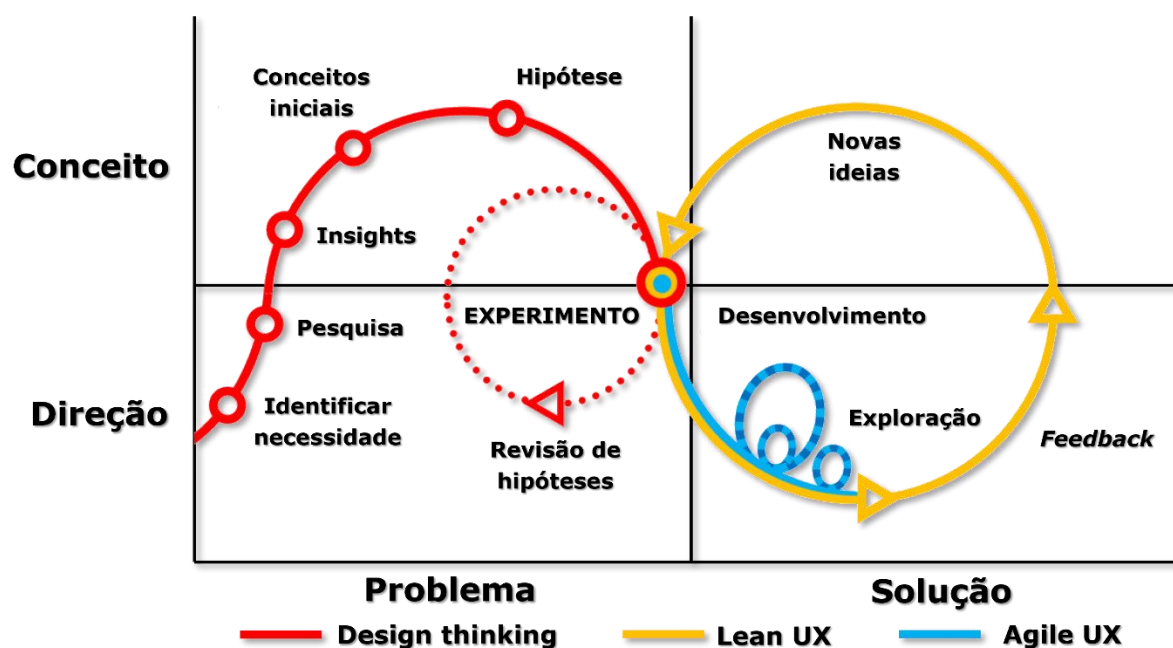
Podemos resumir a contribuição de cada um dos métodos do Tópico 4 no esquema a seguir:



Mas não basta entender o que cada método traz como contribuição para o *design*. O melhor para se aplicar corretamente cada um dos três métodos é entender como eles se encaixam na cronologia de um projeto. Para isso, vamos trabalhar nesta matriz que tem como eixo horizontal as duas grandes fases do projeto (definição do problema e desenvolvimento da solução) e no eixo vertical, os focos que se alternam até a entrega final: as direções a serem tomadas e os conceitos a serem testados.



Agora podemos ver, sobre esta matriz, a curva do projeto combinada com as 3 metodologias.



Fonte: dave.landis@lithespeed.com

Assim fica fácil compreender onde entra cada método e em que fases eles mais colaboram. Agora, é estudar cada um e começar a botar em prática o passo a passo:

- entender o problema a ser resolvido;
- explorar hipóteses de solução;
- especificar e construir o MVP;

-
- prototipar rapidamente e colaborativamente;
 - validar as hipóteses com os usuários;
 - refinar as ideias, aprimorar a solução;
 - desenvolver:
 - times multifuncionais;
 - reuniões cara a cara, rápidas e frequentes;
 - envolver o cliente no processo de *feedback*;
 - adaptação às mudanças.

Assim como em diversas áreas do conhecimento, a solução de problemas pode ser encontrada capturando o que há de bom em diversas fontes de conhecimento. Com o *design* ágil, não é diferente: ao realizar a fusão do *Design Thinking*, o *Lean* e o *Agile* em si, podemos trabalhar com um sistema extremamente robusto e que garante entregas rápidas e com máximo de qualidade.

REFERÊNCIAS

PORTMAN, H. A Project Manager's Guide to 42 Agile Methodologies. **The Digital Project Manager**. Disponível em: <<https://thedigitalprojectmanager.com/agile-methodologies/>>. Acesso em: 11 jul. 2022.

SILVA JÚNIOR, C. O Conceito do The Heart Of Agile. **LinkedIn**, 29 maio 2020. Disponível em: <<https://www.linkedin.com/pulse/o-conceito-do-heart-agile-carlos-silva-j%C3%BAnior-/?originalSubdomain=pt>>. Acesso em: 11 jul. 2022.

TEIXEIRA, F. A diferença entre Lean UX e Agile UX (e a importância dos protótipos no design). **Medium**, 3 maio 2015. Disponível em: <<https://brasil.uxdesign.cc/a-diferen%C3%A7a-entre-lean-ux-e-agile-ux-e-a-import%C3%A2ncia-dos-prot%C3%B3tipos-no-design-85607e43a882>>. Acesso em: 11 jul. 2022.

TOP 7 Agile Frameworks. **Premieragile.com**. Disponível em: <<https://premieragile.com/types-of-agile-frameworks/>>. Acesso em: 11 jul. 2022.