



A combinação de três metodologias para o gerenciamento ágil de projeto: da ideia ao desenvolvimento de um software.

The combination of three methodologies for agile project management: from idea to software development.

ALMEIDA, David¹; RIBEIRO, Victor
davidwsalmeida@gmail.com¹; victorvidigal@gmail.com

¹Especialista em Gestão e Gerenciamento de Projetos

Informações do Artigo

Palavras-chave:
 Desenvolvimento de
 Software
 Metodologias Ágeis
 Incertezas em Projetos

Keywords:
 Software Development
 Agile Methodologies
 Uncertainties in Projects

Resumo:

A acelerada evolução tecnológica no setor de Tecnologia de Informação e Comunicação provocam instabilidades e mudanças, fazendo com que os profissionais de TI adotem metodologias que permitam reduzir as incertezas enquanto desenvolvem produtos inovadores. Diante desse cenário, a empresa americana de consultoria Gartner propôs a utilização de uma combinação de três métodos para o desenvolvimento de software: Design Thinking, Lean Startup e Agile. O objetivo deste trabalho foi analisar a utilização dessas três metodologias no desenvolvimento de software verificando a prática de uma Startup brasileira chamada Lemobs. Os achados demonstram que a adoção das metodologias permitem que as equipes de desenvolvimento possam desenvolver os softwares em curtos ciclos, realizando entregas intermediárias para validar as premissas adotadas na fase de ideação. Assim, utilizam das metodologias combinadas se apresentam como uma estratégia bem-sucedida para o desenvolvimento de software em um ambiente de incerteza.

Abstract

The accelerated technological evolution in the Information and Communication Technology sector causes instabilities and changes, forcing IT professionals to adopt methodologies that reduce uncertainty while developing innovative products. Given this scenario, the American consulting company Gartner proposed the use of a combination of three methods for software development: Design Thinking, Lean Startup and Agile. The objective of this work was to analyze the use of these three methodologies in software development by verifying the practice of a Brazilian Startup called Lemobs. The findings demonstrate that the adoption of methodologies allows development teams to develop software in short cycles, carrying out intermediate deliveries to validate the assumptions adopted in the ideation phase. Thus, using combined methodologies presents itself as a successful strategy for software development in an uncertain environment.

1. Introdução

A crescente digitalização do mundo físico está impulsionando o crescimento da produção de *softwares* que requerem maior desempenho no desenvolvimento de produtos e excelência operacional para gerenciar a maior complexidade no contexto do projeto, do produto e do próprio sistema [1]. Como consequência, organizações engajadas em construir *softwares* estão substituindo métodos tradicionais de gerenciamento de projetos por metodologias mais flexíveis [2].

Os processos de gerenciamento de projetos são orientados pela definição de escopo do produto. O escopo do produto pode ser entendido como o conjunto de funções e recursos que caracterizarão o produto, resultado das entregas do projeto [3].

Em um ambiente de inovação, há uma grande incerteza quanto a definição do escopo do produto. De fato, a incerteza é um aspecto inerente ao processo de inovação que só se torna possível quando a organização avança em uma direção desconhecida [4] na intenção de validar uma hipótese.

No que diz respeito ao desenvolvimento de software as incertezas e o desconhecido tangenciam as especificações do produto, a viabilidade técnica da inovação, utilidade, funcionalidade ou a qualidade do que deve ser produzido [5].

Para tal, o desenvolvimento de software deve ser feito de forma incremental, em estágios, com participação e replanejamento contínuos do usuário, e com programação do projeto conforme o custo em cada estágio [6]. Assim, surgem as metodologias de gerenciamento ágil, como forma de entregar valor à organização conforme as descobertas ou respostas às incertezas são sanadas.

O gerenciamento ágil de projetos é uma abordagem por meio de uma série de etapas que envolve a entrega de valor de forma iterativa e incremental ao longo do ciclo de vida do projeto [7].

O recente estudo realizado pela consultoria Gartner [8] combinou uma

metodologia de desenvolvimento ágil com outros dois métodos para a resolução de problemas e redução de incertezas nos negócios, permitindo que os profissionais envolvidos no desenvolvimento dos softwares possam promover a inovação digital em suas organizações.

Diversos estudos [9;10;11;12;13] foram realizados com a mesma proposta, a combinação dessas três metodologias. Para Schneider [9], o *Design Thinking* permite explorar problemas e oportunidades, o Lean nos permite testar nossas crenças e aprender com isso, o que nos leva a construir as coisas certas e *Agile* (ou metodologia de gerenciamento de projetos Ágil) é uma maneira de construir as coisas da maneira certa.

A startup carioca Lemobs desenvolve soluções tecnológicas para as cidades inteligentes. Utiliza a combinação das metodologias propostas para o desenvolvimento dos *softwares* para os seus clientes.

O presente trabalho apresenta uma análise da combinação de metodologias sugeridas pela consultoria Gartner adotadas pela *startup* Lemobs para a construção de produtos de *software*. Além desta seção introdutória, este artigo é composto de uma seção de apresentação da startup pesquisada, seguida de uma seção com a definição das metodologias e, por fim, uma seção com as considerações finais.

2. A Startup Lemobs

A Lemobs faz parte do ambiente de inovação da Universidade Federal do Rio de Janeiro através da Incubadora de empresas da COPPE/UFRJ e classificada como uma *Govtech*, por ter como característica o fornecimento de serviços tecnológicos para Governo.

O estudo “As startups *Govtech* e o futuro do governo no Brasil” [14] desenvolvido pela *Corporación Andina de Fomento* (CAF) ligada ao Banco de Desenvolvimento da

América Latina, a Lemobs é citada como uma das 80 *Govtechs* de destaque no mercado brasileiro.

Tem como foco o desenvolvimento de *softwares* para o gerenciamento das cidades, utilizando Tecnologias de Informação e Comunicação (TIC) para a melhorias dos serviços fornecidos pela administração pública municipal.

Em um cenário de alta complexidade e incerteza e evolução tecnológica acelerada, as equipes da Lemobs adotam a combinação das metodologias *Design Thinking*, *Lean Startup* e *Agile* para o desenvolvimento de seus *softwares*.

3. Metodologias para o desenvolvimento de Software

Uma metodologia de desenvolvimento de *software* compreende o conjunto de processos necessários para a produção do *software*. Para Summerville [5] existem muitos processos de desenvolvimento de *software* diferentes, mas todos devem incluir quatro atividades fundamentais:

1. Especificação de *software* - a funcionalidade do *software* e as restrições para o seu funcionamento devem ser definidas.
2. Projeto e implementação de *software* - O *software* deve ser produzido para atender às especificações.
3. Validação de *software* - o *software* deve ser validado para garantir que atenda às demandas do cliente.
4. Evolução do *software* - o *software* deve evoluir para atender às necessidades de mudança dos clientes.

Para o desenvolvimento de um *software* os diversos atores envolvidos utilizam um conjunto de metodologias de acordo com cada etapa da produção para dar suporte ao aprendizado das equipes de forma iterativa e focada no usuário/cliente.

A combinação de três metodologias foi recomendada no relatório da consultoria

Gartner [8] com foco na descoberta do que o cliente realmente precisa, agilidade no desenvolvimento e inovação incremental em cenários de total incerteza. A Figura 3 disponível no Anexo apresenta o gráfico extraído do documento demonstrando a combinação das três metodologias iterativas e experimentais para dar suporte aos esforços de produção de produtos digitais inovadores.

A abordagem proposta por Gartner possibilita organizar as metodologias de acordo com as etapas do projeto, sendo a primeira parte a jornada de descoberta dos problemas dos clientes e a segunda, a construção e implantação das soluções dos problemas identificados. Nas próximas sessões, apresentaremos uma breve explicação dessas duas etapas e as metodologias envolvidas neste contexto.

3.1. *Customer PROBLEM*: a jornada de descoberta dos problemas do cliente

Entender os problemas que os clientes (público-alvo) enfrentam é um caminho necessário para a compreensão das especificações do produto e, por sua vez, a definição do escopo do projeto. Os problemas dos clientes são dificuldades que podem variar de intensidade ou frequência nas suas rotinas e que atrapalham a conclusão de seus objetivos. Diversos desses problemas podem ser resolvidos através das TICs, gerando oportunidades de mercado de acordo com o público-alvo com a intensidade e frequência dos distúrbios causados.

O primeiro passo para a descoberta dessas dificuldades é entender o perfil desse cliente, o que eles querem fazer e o que atrapalha para que eles façam de forma satisfatória. Após esse entendimento, é necessário uma abordagem criativa e focada na resolução do problema para encontrar propostas de soluções efetivas. A metodologia de *Design Thinking* adota processos para a geração de ideias para a resolução dessas dificuldades contando com uma equipe multidisciplinar.

3.1.1. Design Thinking

O *Design Thinking* alcança soluções inovadoras por meio da compreensão do contexto em que um projeto de inovação é desenvolvido. Brown [15], define o *Design Thinking* como uma ferramenta que visa a compreensão, visualização e descrição de problemas complexos para, através da criatividade, auxiliar no desenvolvimento de soluções para tais problemas

O *Design Thinking* apresenta um fluxo de atividades iterativas que envolvem o processo de pesquisa (Inspiração), geração de ideias e protótipos (Ideação) e execução da solução (Implementação), que contribuem para que a equipe envolvida no projeto compreenda quais são os recursos e habilidades necessárias para o desenvolvimento de uma inovação nos produtos, nos processos e nos modelos de negócio [16]. A Figura 4 disponível no Anexo apresenta as três etapas do *Design Thinking* e o conjunto de atividades, técnicas e informações condizentes com cada uma delas.

A Inspiração, etapa de pesquisa e descoberta de oportunidade, é composta por pesquisa detalhada sobre o contexto e observação do público-alvo para compreender suas reais preocupações, aspirações e desejos. Nesta etapa, a equipe procura desenhar uma persona, um ser fictício que represente o usuário com as características principais encontradas no público-alvo, para ajudar na compreensão e estimular a empatia dos colaboradores.

Nesta fase, algumas perguntas essenciais são realizadas aos envolvidos, em geral utilizando o framework 5W2H [18], sigla para as perguntas: *what?* (o que?), *who?* (quem?), *where?* (onde?), *when?* (quando?), *why?* (por quê?), *how?* (como?) e *how much?* (quanto?).

As respostas dessas perguntas permitem que a equipe tenha um entendimento do fluxo dos processos realizados pelos clientes facilitando a identificação das oportunidades de melhorias e dos processos que serão

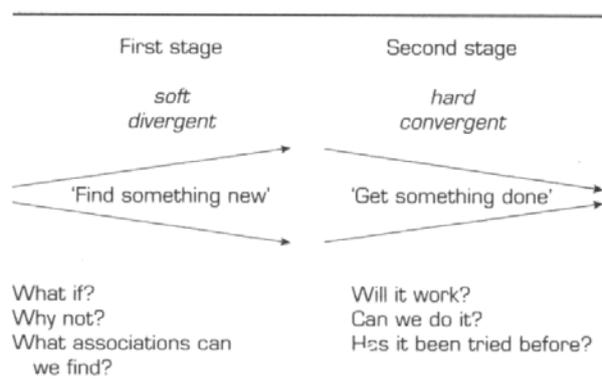
afetados pelas propostas de soluções. Para facilitar a visualização e comunicação, cria-se um modelo dos processos [19], representação gráfica com diversas informações sobre recursos, pessoas, insumos e resultados, que poderá ser atualizado ao longo do projeto.

A fase de Ideação consiste no conjunto de atividades de criação de ideias e elaboração de protótipos para visualização e validação das ideias. A criação de ideias pode ser realizada através de reuniões com a equipe utilizando técnicas como o *Brainstorm*. Barker [20] afirma que a utilização da técnica de *Brainstorm* fornece novas e poderosas maneiras de pensar, criando associações entre as ideias, e estímulos aos pensamentos criativos do grupo.

O pensamento criativo pode ser dividido em dois estágios. O primeiro estágio envolve o pensamento divergente, com o propósito de encontrar algo novo, os participantes são estimulados a questionar suas suposições, buscando outras percepções do objeto ou fazendo aplicações aleatórias com ideias não relacionadas. O segundo estágio envolve o pensamento convergente, julgando o conjunto de ideias que surgiram na fase divergente, fazendo conexões e desenvolvendo-as como algo executável, utilizando lógica, classificação, medição e análise comparativa com os objetivos. O objetivo é obter algo que possa ser realizado. A Figura 1 representa os dois estágios do pensamento criativo [20].

A multidisciplinariedade da equipe estimula o pensamento criativo coletivo e traz para a discussão novos olhares e percepções, importantes temas para o pensamento divergente. Paletz e Schunn [21] identificaram que equipes multidisciplinares com diversidade de conhecimento (níveis de especialização diferentes) podem ser mais inovadores no longo prazo. É interessante que a diversidade de conhecimento seja estendida como uma composição de equipe plural em termos de gênero, etnia e cultura.

Figura 1 - Os 2 estágios do pensamento criativo.



Fonte: extraído de Barker et al. [20]

Após a criação de ideias, é necessário validar junto ao cliente com modelos visuais. As equipes desenvolvedoras da Lemobs utilizam a prototipação como uma etapa importante para validação das regras e escopo do código que será desenvolvido. Os protótipos permitem que as equipes possam refletir sobre a usabilidade do produto, testar os conceitos e reduzir a incerteza do que tem que ser feito. Também apresentam ao cliente para que possam validar se a hipótese de solução conseguirá realizar o trabalho a ser feito.

Com o objetivo de facilitar o desenvolvimento de novas soluções e aproveitar o trabalho já realizado com a construção de elementos do sistema, a empresa Lemobs criou um *Design System* denominado *Hexagon*. Trata-se de um catálogo de padrões de componentes de *design* do sistema que podem ser reutilizados. É um documento composto por um guia de padrões, biblioteca de componentes e biblioteca de padrões que comunicam as regras de cores e componentes em cada tipo de tela dos sistemas, economizando tempo e esforço das equipes e mantendo a identidade visual criada para cada marca. Para Kalerme [22], “quando os desenvolvedores não precisam perder tempo se perguntando sobre o *design* dos componentes, eles podem se concentrar mais na usabilidade e funcionalidade”.

Com as ideias e hipóteses validadas, alcança-se o conceito do que será a solução.

Neste momento, as equipes avançam para as etapas de construção do produto e futura implementação. Vale lembrar que essas etapas são iterativas e podem ser repetidas até que as ideias e o conceito da solução estejam claros para todos os envolvidos, principalmente para os clientes.

3.2. Customer SOLUTION: Construção da solução que resolvem os problemas

A definição do conceito da solução dará origem às especificações dos softwares da Lemobs. É provável que essas especificações representem um conjunto de atividades que a equipe deverá executar à longo prazo (mais de 6 meses). Neste momento, as equipes devem organizar as tarefas para que possam definir o escopo mais imediato a ser entregue ao cliente e o que será entregue ao longo do tempo. A metodologia Lean Startup adota uma abordagem para a definição do escopo inicial e um ciclo contínuo de aprendizado.

3.2.1. Lean Startup

A metodologia *Lean Startup* está fundamentada no ciclo de *feedback* construir-medir-aprender. Trata-se de um fluxo fundamental para inovação que consiste em transformar ideias em produtos, medir como os clientes reagem e, então, aprender com o *feedback* e voltar a construir de forma incremental o produto [23].

Para entrar na fase de construção o mais rápido possível, é importante definir o que deve ser entregue primeiro que agregue valor ao cliente para que ele possa iniciar a sua utilização. A versão do produto que permite uma volta completa do ciclo construir-medir-aprender com esforço mínimo e tempo mínimo de desenvolvimento foi denominada por Reis [23] como o produto mínimo viável (PMV). Provavelmente faltarão muitos itens essenciais no PMV. É importante lembrar que é necessário medir o impacto da utilização do PMV para a aprendizagem ser realmente efetiva.

Ao realizar os ciclos construir-medir-aprender os membros da equipe devem avaliar se a hipótese inicial de solução está sendo confirmada com o uso do PMV. Em

caso negativo, deve-se avaliar se a equipe deve manter a estratégia (perseverar) ou realizar uma grande mudança (*pivot*) e retornar para fase de ideação e elaborar novas hipóteses.

Um dos principais benefícios dessa metodologia é o rápido aprendizado, permitindo que a equipe possa avaliar rapidamente que uma ideia não está sendo efetiva e possa tentar outra ideia sem que os impactos de tempo e esforço empreendidos para desenvolver um produto completo sejam extremos [24].

No caso de um teste de hipótese bem-sucedido, a equipe deve organizar novas tarefas para incrementar de forma contínua novas funcionalidades ao produto, realizando novos ciclos construir-medir-aprender. A metodologia ágil (*Agile*) propõe uma organização das tarefas para execução em pequenos ciclos, facilitando a adaptação às mudanças oriundas do *feedback* do cliente.

3.2.2. *Agile*

Modificar um produto tradicional que já foi disponibilizado para o mercado é uma tarefa muito complexa para as empresas fornecedoras. A indústria automobilística, por exemplo, ao identificar um problema em um dos seus produtos precisa realizar um chamamento público aos proprietários para que levem seus veículos até as oficinas e, assim, possam corrigir o problema. As despesas com a divulgação do chamamento e com a manutenção são facilmente mensuráveis, mas o incômodo gerado aos proprietários pode gerar grandes insatisfações e torná-los verdadeiros detratores da marca.

Os *softwares* modernos se favorecem da conexão via internet com a capacidade de atualizar as suas versões sem grandes impactos aos clientes. As versões com correções e com melhorias incrementais são atualizadas sem que a maioria dos usuários perceba.

Nesse contexto, a metodologia de gerenciamento ágil adapta-se de forma

favorável por permitir que as equipes possam realizar esforços em curtos períodos para incrementar funcionalidades ou correções ao código-fonte e disponibilizar uma versão nova do produto com facilidade.

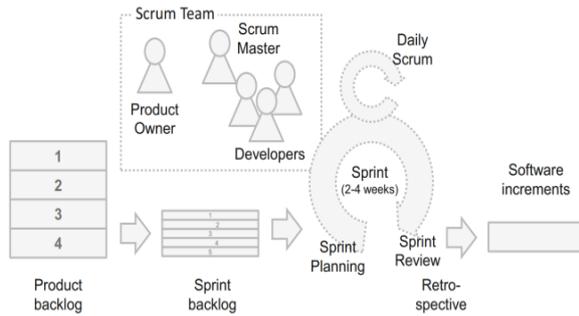
Em 2001, um grupo de engenheiros de software se reuniram para discutir sobre os aspectos comuns aos chamados Métodos Ágeis. Assim, criaram um conjunto de valores e princípios (a lista de princípios do Manifesto Ágil está disponível em: <http://agilemanifesto.org/iso/ptbr/principles.html>) que constituem a essência do desenvolvimento de software ágil. Os quatro valores do Manifesto Ágil são:

- 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

Os engenheiros de software priorizam os valores da primeira parte das sentenças (em destaque) que a segunda, mas ambas são valores importantes para os Métodos Ágeis [25].

O Scrum é o Método Ágil mais popular no Brasil, adotado por 73,8% das organizações que utilizam esta metodologia [26]. Trata-se de um *framework* que especifica (1) certas funções na equipe de desenvolvimento, estabelece o (2) modo de trabalho iterativo que gira em torno dos ciclos de desenvolvimento, e define diferentes (3) artefatos que os desenvolvedores usam para organização das tarefas [15]. A Figura 4 apresenta os elementos do Scrum e a organização das tarefas da equipe a cada ciclo de desenvolvimento.

Figura 2 - Framework de desenvolvimento Scrum.



Fonte: extraído de Brown [15].

Alguns dos elementos ilustrados na Figura 4 são detalhados nos próximos itens deste trabalho:

3.2.2.1. Equipe

A equipe Scrum é composta por um grupo de até dez profissionais e é obrigatoriamente composta por dois profissionais com funções específicas chamados de *Scrum Master* (SM) e o *Product Owner* (PO).

O *Scrum Master* (SM) assume o papel de um facilitador responsável por manter os processos Scrum e eliminar impedimentos que possam atrapalhar a equipe de trabalhar de forma eficiente.

O *Product Owner* (PO) representa o cliente dentro da equipe e garante que os requisitos transmitidos pelos clientes serão satisfeitos, definindo as metas da equipe a cada ciclo de desenvolvimento.

Todos os outros membros da equipe pertencem à equipe de desenvolvimento e fazem o trabalho de desenvolvimento de software. Eles analisam requisitos e protótipos, desenvolvem, e validam o software. As equipes de desenvolvimento Scrum são multifuncionais, ou seja, espera-se que todos os membros da equipe tenham o conjunto de habilidades necessárias para realizar todas essas tarefas de desenvolvimento de software. Consequentemente, não há funções na equipe de desenvolvimento, como desenvolvedores de interface de usuário, testadores ou outros especialistas.

3.2.2.2. Ciclos de Desenvolvimento

As equipes Scrum buscam um modo de trabalho iterativo e dividem o projeto de desenvolvimento em ciclos de desenvolvimento curtos, os chamados Sprints de desenvolvimento. As Sprints têm uma duração específica de 2 semanas e resultam em uma nova versão do sistema para o cliente.

Cada Sprint começa com uma reunião de planejamento para decidir qual as funcionalidades serão desenvolvidas. A equipe divide o trabalho em subtarefas e define as responsabilidades de cada membro. Todos os membros da equipe se reúnem diariamente por cerca de 15 minutos para informam sobre suas realizações, descrever o trabalho atual e levantar questões a serem abordadas pela equipe. Cada Sprint termina com uma reunião de revisão de Sprint para apresentar os resultados e discutir possíveis melhorias para os processos de trabalho.

3.2.2.3. Organização das Tarefas

A equipe organiza suas tarefas de desenvolvimento usando uma lista de Backlog do produto. No caso da equipe Lemobs, utiliza-se uma ferramenta de gerenciamento ágil de projetos denominado Trello (ferramenta de gerenciamento ágil de projetos, disponível em: <https://trello.com/>). O Backlog contém uma lista de funcionalidades definida pelo PO a partir dos requisitos levantados junto ao cliente. A cada Sprint, a equipe se reúne para planejar (Sprint Planning) o conjunto de itens do Backlog que será priorizado e organiza esses itens em tarefas que serão executadas ao longo do ciclo, formando assim o Sprint Backlog.

4. Considerações Finais

O objetivo deste artigo foi apresentar os métodos utilizados para o desenvolvimento de software adotados pela *startup* Lemobs de acordo com a proposta de combinação de três metodologias realizada pela consultoria Gartner.

A combinação dessas metodologias permite que os esforços para a construção de um sistema sejam mais assertivos uma vez que envolvem diversos ciclos de aprendizado e consultas aos usuários que utilizarão o sistema.

A adoção dessas metodologias combinadas uma abordagem organizada em duas etapas de desenvolvimento do projeto, a primeira referente à jornada de descoberta dos problemas dos clientes e, a segunda, relativa à construção e implantação das soluções dos problemas identificados.

O curto ciclo de desenvolvimento permite disponibilizar produtos intermediários, permitindo validar as hipóteses e reavaliar as diversas premissas adotadas no planejamento do projeto realizada em um cenário de muitas incertezas.

Para os cenários de grande incerteza que comumente as *Startups* e empresas de desenvolvimento de *software* estão inseridas, é necessário adotar metodologias que permitem o desenvolvimento do projeto sem que todas as respostas e informações sejam disponibilizadas e a descoberta desses dados sejam realizados durante a execução do projeto.

Na opinião do autor, a utilização da combinação das metodologias propostas por Gartner permite que as incertezas sejam reduzidas e aumenta a chance de sucesso na construção de um produto de *software* inovador.

Este trabalho não coletou as opiniões dos participantes do projeto, portanto, para futuros estudos, recomendamos realizar levantamento em campo com as equipes desenvolvedoras envolvidas e os clientes que serão usuários das soluções tecnológicas.

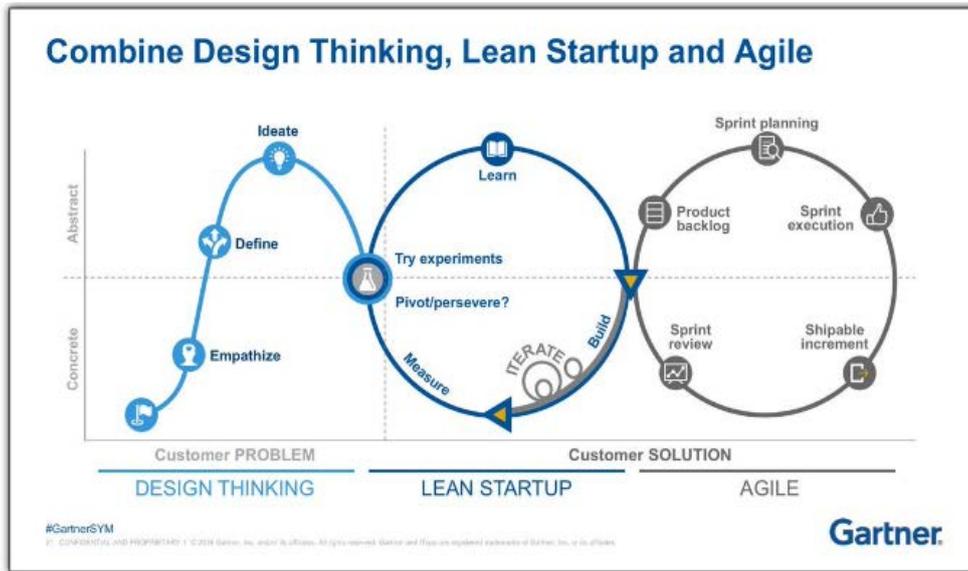
Referências

- [1] TIBAZARWA, A. “Strategic Integration for Hardware and Software Convergence Complexity”, IEEE Eng. Manag. Rev., p. 1–1, 2021, doi: 10.1109/EMR.2021.3089475.
- [2] CONFORTO, E. C. SALUM, F. AMARAL, D. C. SILVA, S. L. da e ALMEIDA, L. F. M. de “Can Agile Project Management be Adopted by Industries Other than Software Development?”, Proj. Manag. J., vol. 45, no 3, p. 21–34, jun. 2014, doi: 10.1002/pmj.21410.
- [3] PMI. Project Management Institute “PMBOK® Guide”, São Paulo. <https://pmisp.org.br/pmbok-guide/> (acessado set. 16, 2021).
- [4] JALONEN, H. “The uncertainty of innovation: A systematic review of the literature”, J. Manag. Res., p. 1–47, 2011.
- [5] CARVALHO A., “TMP08 - engenharia de software 9 edicao”, Acessado: ago. 17, 2021. [Online]. Disponível em: https://www.academia.edu/31449888/TMP08_engenharia_de_software_9_edicao
- [6] MILLS, H. D. “Software Development”, IEEE Trans. Softw. Eng., vol. SE-2, no 4, p. 265–273, dez. 1976, doi: 10.1109/TSE.1976.233831.
- [7] TAYLOR, P. e SHARROCK, E. “Agile project management”, Make Your Business Agile, Routledge, 2021.
- [8] GARTNER, “Enterprise Architects Combine Design Thinking, Lean Startup and Agile to Drive Digital Innovation”, Gartner. Disponível em: <https://www.gartner.com/en/documents/3941917/enterprise-architects-combine-design-thinking-lean-start> (acessado ago. 17, 2021).
- [9] SCHNEIDER, J. “Understanding Design Thinking, Lean, and Agile”, p. 76.
- [10] DOBRIGKEIT, F., DE PAULA, D., UFLACKER, E. M. “InnoDev: A Software Development Methodology Integrating Design Thinking, Scrum and Lean Startup”, Design Thinking Research, Springer International Publishing, 2019, p. 199–227. doi: 10.1007/978-3-319-97082-0_11.

- [11] DOBRIGKEIT, F., DE PAULA, D., CARROLL, N. “*InnoDev Workshop: A One Day Introduction to Combining Design Thinking, Lean Startup and Agile Software Development*”, 32nd Conference on Software Engineering Education and Training (CSEET), nov. 2020, p. 1–10. doi: 10.1109/CSEET49119.2020.9206184.
- [12] DE PAULA, D. e ARAÚJO, C. “*Pet Empires: Combining Design Thinking, Lean Startup and Agile to Learn from Failure and Develop a Successful Game in an Undergraduate Environment*”, HCI International 2016. Posters’ Extended Abstracts, Cham, 2016, p. 30–34. doi: 10.1007/978-3-319-40548-3_5.
- [13] COCCHI, N., DOSI, C. e VIGNOLI, M. “*The Hybrid Model Matrix Enhancing Stage-Gate with Design Thinking, Lean Startup, and Agile*”, Res. Technol. Manag., vol. 64, no 5, p. 18–30, set. 2021, doi: 10.1080/08956308.2021.1942645.
- [14] DOMINGUEZ, G. “*As startups govtech e o futuro do governo no Brasil*”. CAF y BraziLab, 2020. Acessado: ago. 15, 2021. [Online]. Disponível em: <https://scioteca.caf.com/handle/123456789/1653>
- [15] BROWN, T. “*Design Thinking: Uma metodologia poderosa para decretar o fim das velhas ideias*”. Alta Books, 2020.
- [16] MACEDO, M., MIGUEL, P. e FILHO, N. “*A Caracterização do Design Thinking como um Modelo de Inovação*”, RAI Rev. Adm. E Inov., vol. 12, no 3, p. 157–182, jul. 2015, doi: 10.11606/rai.v12i3.101357.
- [17] BROWN, T. “*Design Thinking*”, Harvard Business Review, jun. 01, 2008. Acessado: ago. 17, 2021. [Online]. Disponível em: <https://hbr.org/2008/06/design-thinking>
- [18] KLOCK, A., GASPARINI, I. e PIMENTA, M. “*5W2H Framework: a guide to design, develop and evaluate the user-centered gamification*”, Proceedings of the 15th Brazilian Symposium on Human Factors in Computing Systems, New York, NY, USA, out. 2016, p. 1–10. doi: 10.1145/3033701.3033715.
- [19] ABPMP “*CBOK: Guia para o Gerenciamento de Processos de Negócio. Corpo Comum do Conhecimento*”, BPM CBOK V3., Association of Business Process Management Professionals, 2013
- [20] BARKER, A., FORSYTH, P., SMITH, J. e BARKER, A. “*30 minutes*”. London: Kogan Page, 2003.
- [21] PALETZ, S. e SCHUNN, C. “*A Social-Cognitive Framework of Multidisciplinary Team Innovation*”, Top. Cogn. Sci., vol. 2, no 1, p. 73–95, 2010, doi: 10.1111/j.1756-8765.2009.01029.x.
- [22] KALERMO, T. “*Building a Design System*”. Metropolia University of Applied Sciences. Acessado: ago. 17, 2021. [Online]. Disponível em: https://www.theseus.fi/bitstream/handle/10024/503045/Kalermo_Tatu.pdf?sequence=2&isAllowed=y
- [23] REIS, E. “*A startup enxuta*”. Sextante, 2016.
- [24] EISENMANN, T., RIES, E. E DILLARD, S., “*Hypothesis-Driven Entrepreneurship: The Lean Startup*”.
- [25] FOWLER, M. e ABSTRACT, G. H. “*Agile EAP*”. 2002.
- [26] MELO, C. SANTOS, V. A.; CORBUCCI, H.; KATAYAMA, E.; GOLDMAN, A.; KON, F. “*Métodos ágeis no Brasil: estado da prática em times e organizações*”, p. 9. Relatório Técnico RT-MAC-2012-03. Departamento de Ciência da Computação. IME---USP. Maio, 2012.
- [27] SCHMIDT, C. “*Agile Software Development*”, Agile Software Development Teams, Cham: Springer International Publishing, 2016, p. 7–35. doi: 10.1007/978-3-319-26057-0_2.

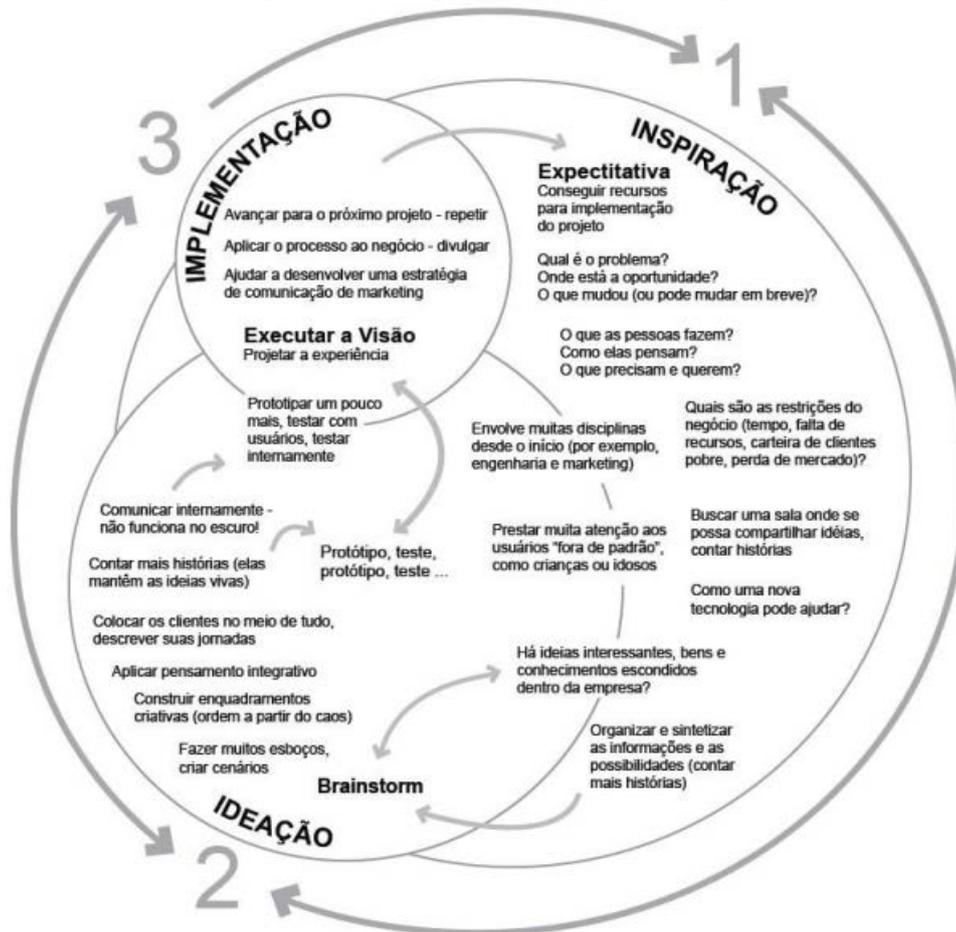
1. Anexos e Apêndices

Figura 3 - Metodologias combinadas propostas por Gartner [8].



Fonte: Gartner, 2018 [8].

Figura 4 - Três processos do Design Thinking.



Fonte: extraído de Macedo et al [16] adaptado de Brown, [17].