Discussão sobre o uso de metodologias ágeis no desenvolvimento de produtos industriais e digitais

Discussion on the use of agile framework for the development of industrial and digital products.

JOTA, Gustavo; Bacharel; Universidade de Brasília - UnB

jotagustavo@me.com

MAYNARDES, Ana Claudia; Doutora; Depto. de Design - Universidade de Brasília - UnB

anacmay@gmail.com

Resumo

Metodologias ágeis em produtos digitais (software) são principalmente usadas no desenvolvimento de produtos, sejam eles serviços, software ou modelos de negócios inovadores e disruptivos. Estas metodologias possuem como principais características a velocidade, adaptabilidade e a falta de um modelo de referência baseado na modelagem prévia de etapas, processos e ferramentas: resultando em um modelo evolutivo e empírico de desenvolvimento de ambos produto, cliente e modelo de negócios, conforme defendido por BLANK (2017). Este artigo se propõe a discutir e confrontar as metodologias prescritivas baseadas em 'stage gates' defendidas por Cooper (2001) que são a base de metodologias de desenvolvimento integrado de produtos, *versus* as metodologias ágeis a fim de se avaliar a factibilidade de futura adaptação ou unificação de metodologias e ferramentas visando atender a projetos com características de alta complexidade tanto projetuais como de negócio.

Palavras Chave: processo de desenvolvimento integrado de produto; metodologias ágeis; empreendedorismo.

Abstract

Agile frameworks (or methodologies) in digital products (software) are mainly used for developing products, being these services, software or innovative and disruptive business models. These frameworks present as their key characteristics the speed, adaptability and absence of a reference model based on historical modeling of phases, processes and tools: resulting in an evolutionary and empirical model that develops concurrently both product, customer and business model, as defended by BLANK (2017). This article proposes to discuss and confront the prescriptive methodologies based on stage gates defended by Cooper (2001) that are the basis of integrated product development methodologies versus agile frameworks in order to evaluate the feasibility of future adaptation or unification of methodologies /frameworks and tools to fit projects with high complexity characteristics, both project and business.

Keywords: integrated product development process; agile framework; entrepreneurship

1 Introdução

É observada nas duas últimas décadas de 2000 e 2010 a aceleração da obsolescência e canibalização de diversas categorias de produtos industriais e mesmo digitais (software). Segundo LORENZ (1986) este processo iniciou-se na década de 1980 quando os japoneses lançaram a obsolescência programada, que nestas últimas duas décadas tem se acentuado devido aos lançamentos inovadores e disruptivos de tecnologia que introduzem produtos (digitais e físicos ou ambos) que competem diretamente com categoria de produtos outrora estabelecidos. Como resultado, as pressões sobre a correta seleção de metodologia, ferramentas e abordagens ao projeto tornam-se ainda mais cruciais para o sucesso de novos produtos. Atualmente duas principais correntes de pensamentos guiam os modelos/metodologias de desenvolvimento: os modelos prescritivos *versus* adaptativos.

Os modelos de desenvolvimento prescritivos têm por característica descrever como o processo de desenvolvimento a ser iniciado deve ocorrer, sugerindo os melhores caminhos, práticas e mesmo ferramentas. Após o ciclo de desenvolvimento é realizada a avaliação do projeto, e realizada se necessária a atualização e/ou revisão do modelo. Rozenfeld et al (2006) propõe a divisão de seu modelo de referência (metodologia) em três macro-fases: Pré-Desenvolvimento, Desenvolvimento e Pós-Desenvolvimento. Nestas, existem fases separadas por 'stage gates' de avaliação, conforme proposto por Cooper (2001), que reforça o caráter de detalhamento prescritivo do projeto do produto anterior à sua execução.

Já os modelos de desenvolvimento adaptativos, segundo Sutherland & Schwaber (2017) são abordagens ou estruturas (do inglês framework) dentro do qual pessoas tratam e resolvem problemas complexos e adaptativos. E dentro do mesmo podem ser usadas conforme a necessidade, vários processos, ferramentas ou técnicas, iterações, mudança de escopo de produto e outros com grande flexibilidade e priorização, resultando, portanto, em um modelo adaptativo.

Este estudo tem como objetivo discutir e confrontar as diferenças metodológicas entre as duas abordagens de desenvolvimento prescritivas *versus* adaptativas que utilizam do design em sua matriz de envolvidos no projeto. Como objetivo secundário iniciar a identificação de fatores que auxiliem na seleção de qual abordagem utilizar, ou mesmo na modificação das abordagens visando o sucesso comercial de novos produtos.

2 A importância de novos produtos e sua definição

Cooper (2017) apresenta a importância do desenvolvimento de novos produtos nas empresas atuais, apontando os potenciais riscos ou ganhos financeiros, inclusive trazendo estudos conduzidos por instituições financeiras que apontam esta importância sob a ótica de investidores (capital), conforme:

O desenvolvimento de novos produtos é um dos mais arriscados, contudo importantes, empreendimentos nas modernas corporações. (...) Os riscos são elevados inclusive com grande gasto de recursos financeiros em desastrosos novos produtos, mas os ganhos também podem ser elevados.

Uma pesquisa do Deutsche Bank realizada em 2011 em mais de mil empresas apontou que aquelas que gastam significativamente mais em pesquisa & desenvolvimento que seus competidores são melhor avaliadas pelos investidores (COOPER, 2017, p.13, tradução nossa).

O autor prossegue apresentando uma definição estendida do que são novos produtos, ampliando a tradicional visão de que produtos são apenas bens manufaturados em escala. Importante ressaltar que o presente artigo utilizará daqui por diante esta definição ampliada de 'produto' e/ou 'novo produto' para produtos físicos e intangíveis (a exemplo de produtos bancários ou mesmo de software como serviço / SaaS):

Produto: Primeiro, um "produto" é tudo aquilo que é direcionado para o mercado, seja para venda, uso ou consumo. Inclui produtos físicos, assim como software (programa de computador) e serviço, inclusive combinações de serviços, hardware (dispositivo físico) e/ou software. (COOPER, 2017, p.13, *tradução nossa*).

Finalmente o autor aponta que a definição 'novos produtos' para as empresas tende a sofrer diferença de interpretação oriunda de fatores internos a cada empresa e da dinâmica do mercado no qual estão inseridas. Propondo, por conseguinte esta classificação: [1] Novos para o mundo; [2] novas linhas de produtos; [3] adição nas linhas de produtos; [4] melhorias e revisões em produtos existentes; [5] reposicionamento; [6] redução de custos. Adicionalmente apresenta um gráfico em que distribui os 'tipos de novos produtos', de acordo com sua proposta classificação, em um gráfico tridimensional onde temos novidade para a empresa no eixo Y, novidade para o mercado no eixo X e a quantidade de lançamentos no eixo Z (pesquisa conduzida na América do Norte).

Figura 1 - Classificação de novos produtos

Fonte: COOPER, 2017, p.26.

3 Modelos de Desenvolvimento Prescritivos

Os modelos prescritivos têm como principal característica o detalhamento das atividades do projeto orientadas a partir de um modelo de referência (ou metodologia). Raz et al (1998) propõem o WBS (work breakdown structure ou EDT - estrutura desdobramento do trabalho) como a ferramenta para detalhar e planejar o escopo do projeto anteriormente à sua aprovação e execução (sizing ou dimensionamento). Modelos de desenvolvimento prescritivos propõem uma estrutura na qual as atividades são divididas em fases (evolutivas), separadas por avaliações com sequencia linear. Este artigo não detalhará ou discorrerá sobre o WBS e seu papel dentro de modelos prescritivos, apenas aponta o uso desta ferramenta visto que é prática comum o uso dos conhecimentos do PMI (Project Management Institute)¹ junto a modelos de referência prescritivos.

É interessante discorrer sobre os modelos ou estruturas² tradicionalmente utilizadas na bibliografia de desenvolvimento de produto para posterior análise comparativa entre os modelos Prescritivos e Adaptativos.

3.1 Modelo Unificado do Processo de Desenvolvimento do Produto (PDP)

O modelo apresentado por Rozenfeld et al (2006) é voltado principalmente para empresas de manufatura e bens de consumo duráveis e de capital. Aborda desde o "pré-desenvolvimento" até o "pós-desenvolvimento", apresentando como o modelo de referência se encaixa em uma estrutura empresarial multidisciplinar (inclusive trata das estruturas organizacionais discutidas no PMI) e seus impactos no projeto. Tem como característica de modelo prescritivo a busca pelo detalhamento e controle do processo de desenvolvimento.

A dificuldade de descrever como é o processo de desenvolvimento tem reflexos importantes na forma como ele pode ser gerenciado. Como prever, planejar e controlar o trabalho das pessoas uma vez que nem todos tem uma linguagem comum e uma visão mínima do andamento do projeto geral e da contribuição que é esperada para a empresa? Lembre-se de que, quando uma linha de produção para, todos ficam sabendo, pois a paralisação os afeta de maneira imediata. Ao contrário, um problema em um processo de desenvolvimento pode aparecer meses ou até mesmo anos após a sua ocorrência. O primeiro passo para o gerenciamento eficiente do processo de desenvolvimento de produtos, portanto, é torna-lo "visível" a todos os atores envolvidos. (ROZENFELD, 2006, p. 38).

Esta afirmação discorre tanto sobre a necessidade de controle no projeto, como no impacto futuro de problemas oriundos do modelo nos produtos no mercado (em operação, serviço ou produção), mas, principalmente, declara a filosofia do modelo prescritivo.

Como mencionado acima, o PDP proposto por Rozenfeld et al. (2006) é estruturado em

¹ Nota: O WBS ou EDTs é a decomposição hierárquica de todo o escopo de trabalho (atividades relacionadas ao processo de desenvolvimento) necessários para que as entregas previstas no escopo do projeto sejam entregues (o que o projeto realiza), viabilizando a contabilidade e rastreamento de custos, prazos, indicadores de qualidade dentre outros.

² Nota: "Estrutura" pode ser encontrada na língua Inglesa como "Framework" especialmente na literatura de software ágil ou cascata. Neste artigo será usado o termo "Estrutura" seguido de "Framework" para melhorar a compreensão e posteriores pesquisas no tema.

três macro-fases: pré-desenvolvimento, desenvolvimento e pós-desenvolvimento. No pré-desenvolvimento encontram-se as atividades que relacionam o planejamento estratégico da empresa com o modelo de referência proposto e o seu planejamento. O Desenvolvimento concentra as etapas "Projeto Informacional", "Projeto Conceitual", "Projeto Detalhado" e "Preparação para a Produção do Produto". A última macro-fase, de pós-desenvolvimento, discorre sobre o "Lançamento do Produto", acompanhamento, suporte, atualização /re-projetos até a descontinuação do produto do mercado.

Dentro destas etapas se distribuem a aplicação de diversas ferramentas utilizadas pelos profissionais envolvidos no processo de desenvolvimento de produto. Baxter (1998) e Rozenfeld et al. (2006) compartilham inclusive da utilização de diversas ferramentas vastamente conhecidas, como por exemplo de Análise FOFA (SWOT), Brainwriting, Brainstorming, QFD e Kano. Importante ressaltar que este estudo se delimita a discutir e confrontar os modelos, e não na discussão do uso e aplicação correta das ferramentas supracitadas ou outras apresentadas pelos autores em suas respectivas publicações.

Nesta estrutura sequencial de desenvolvimento são utilizados os 'stage gates' propostos por Cooper (2017) que têm a função: aprovação, veto, controle, revisão e mesmo iniciar 'iterações' (reiniciar dentro da fase ou fase anterior) garantindo o correto desenvolvimento das funcionalidades aprovadas no plano de produto (ou escopo do produto). Esta etapa de 'aprovação', de acordo com norma vigente será sempre representada por losangos, e pode ser encontrada na literatura como 'avaliação', 'gate de avaliação', 'gateway' e outros.

Figura 2 - Visão geral da representação do modelo de referência

Fonte: ROZENFELD et al., 2006, p. 44.

Será utilizado este modelo de referência para esboçar o modelo prescritivo, visto que o mesmo é baseado em modelos de diversos autores como Cooper (2017), Baxter (1998) e outros. Não é escopo deste artigo confrontar diferentes modelos de referência de design e engenharia.

3.1.1 Definição de práticas de projeto e a Engenharia Simultânea

Yazdani & Holmes (1999) conduziram um estudo na indústria automotiva e aeroespacial (respectivamente em seis e duas empresas), na Europa, Japão e Estados Unidos (década de 1990) com a finalidade de identificar as principais práticas de projeto (independente do modelo de referência) usando cinco indicadores de performance: (i) forças para introduzir novo produto (NPI) com gerenciamento de performance e forças dominantes; (ii) risco; (iii) tempo; (iv) custo e (v) qualidade. O resultado foi a identificação de quatro abordagens de projeto:

- Modelo Sequencial (Sequential Model)
- Modelo centralizado no projeto (Design Centered Model - ou DFX - design para variável x)
- Definição Simultânea (Concurrent Definition)
- Modelo Dinâmico (Dynamic Model)

Avaliando os indicadores de performance, conforme figura 2, é possível notar a baixa performance da abordagem sequencial especialmente em cenários com acelerado ritmo de lançamento de produtos, controle de custos e de alta competitividade: projetos lentos, de alto custo e qualidade baixa. No outro lado do espectro temos as abordagens dinâmicas e simultâneas como alternativas mais arriscadas em projetos de maior inovação (breakthrough), mas com melhores resultados em tempo e qualidade (projeto e direcionador NPI).

Figura 2 - Comparação das diferentes abordagens de projeto

Table I. Comparison of the identified models				
	Sequential model	Design centered model	Concurrent definition model	Dynamic definition model
Project risk level: incremental	Low	Low	Low	Low
Project risk level: breakthrough	Low	Medium	High	High
Top NPI driver	Cost /quality	Cost /quality	Quality /time	Time
Time	Slow	Medium	Medium	Fast
Cost	High	Medium	Medium /low	Low
Design quality	Low	High	Medium	High
Time, critical path from project launch to volume production; cost, measured in terms of project costs throughout the process; quality of design, measured in terms of cost of engineering changes and also the stage at which change was necessitated				

Fonte: YAZDANY & HOLMES, 1999, p. 35.

Rozenfeld et al (2006), apontam a "Engenharia Simultânea" como uma filosofia de projeto que visa aumentar a velocidade e qualidade do projeto, propondo sua utilização sobre o modelo de referência do PDP Unificado. A figura 3 ilustra a compressão de tempo entre a abordagem simultânea e sequencial do projeto.

Figura 3 - Diferença entre Engenharia Sequencial e Engenharia Simultânea

Fonte: YAZDANY & HOLMES, 1999, p. 26.

Liker et al. (1999) apontam que a Toyota emprega das melhores práticas de projeto utilizando menor tempo para o lançamento de produtos e quantidade de engenheiros quando comparada com montadoras Norte Americanas. Inclusive que não apenas o 'lean manufacturing' (também conhecida como Sistema Toyota de Produção) é responsável pelo sucesso da montadora Japonesa, mas que o design do produto e sistema de desenvolvimento (abordagem) são também fatores chave. Identificaram por meio de entrevistas o uso de Engenharia Simultânea Set-Based como diferencial na abordagem de desenvolvimento de seus automóveis³.

³ Foram realizados seis meses de entrevistas no Japão com gerentes e engenheiros nos diversos ramos do desenvolvimento, incluindo designers de estilo, engenheiros de carroceria, *powertrain* (conjunto motor e câmbio), validação de montagem, engenharia de produção e prototipagem, afiliados e fornecedores da Toyota.

Figura 4 - Engenharia Simultânea Set-Based (Toyota)

Fonte: LIKER et al., 1999, p3.

A Engenharia Simultânea, inclusive defendida por Rozenfeld et al. (2006) tem como principal característica o desenvolvimento em um formato de funil, do qual, a partir de um grande conjunto de soluções (que possibilitem alternativas de soluções quando combinados), serão selecionadas ou eliminadas durante o progresso do projeto através, principalmente, de resultados de testes realizados nos diversos protótipos funcionais. Assim, o resultado são produtos de maior qualidade e confiabilidade lançados em menor espaço de tempo. Importante ressaltar que não é foco discutir qual o modelo dentre as duas abordagens de engenharia simultânea é mais eficiente, e sim neste tópico apresentar que a abordagem simultânea é mais veloz e efetiva que uma abordagem sequencial tradicional.

4 Modelos de desenvolvimento adaptativos

Cusumano (1997) discorre sobre as práticas na indústria de software, especificamente em estudos realizados na Microsoft na década de 1990, que desde aquele momento demonstrava iniciativas de desenvolvimento de produtos diferentes das encontradas em outras áreas de conhecimento tradicionais, mas comuns a outras empresas expoentes da mesma categoria como Yahoo e Netscape que possuíam problemas de negócio semelhantes (inclusive a diversas outras): aumento da competitividade resultando em produtos mais complexos; evolução tecnológica trazendo obsolescência programada; aceleração de lançamentos de produtos no mercado e outros.

Muitos gerentes de projetos de software preferem equipes de produtos pequenas de uma dúzia ou menos de pessoas. Mantendo assim a cultura dos primeiros dias de programação em mainframes (grande computadores) e depois PCs (computadores pessoais), quando duas ou três pessoas poderiam construir um novo produto. As versões iniciais do MS-DOS, Word e Excel da Microsoft no início da década de 1980 tinham equipes de programação de meia dúzia a dez desenvolvedores e consistiam apenas em algumas dezenas de milhares de linhas de código. A primeira versão do Windows NT, introduzida em 1993, consistiu em cerca de 4,5 milhões de linhas de código e contava com uma equipe de desenvolvimento (gerentes de programas, trabalho com base em especificações de produtos, gerenciamento de projetos, desenvolvedores e testadores) de cerca de 450 pessoas no seu ápice. O Windows 95, introduzido em 1995, consistiu em 11 milhões de linhas de código e exigiu um número semelhante de pessoas que trabalharam por aproximadamente três anos. Uma equipe de cerca de 300 pessoas está construindo os principais componentes do navegador Internet Explorer da Microsoft, introduzido em 1996, e várias centenas de pessoas estão envolvidas na criação de vários recursos complementares, como o Microsoft Outlook (correio eletrônico) para Internet. (CUSUMANO, 1997, p.10, *tradução nossa*)

O referido autor identificou que aquelas empresas compartilhavam de algumas práticas: desenvolvimento iterativo, projeto simultâneo, vasta gama de protótipos, entregas incrementais, intensa troca de informações, divisão da arquitetura do produto em times, contato com os usuários, dentre outras, permitindo desenvolver complexos produtos em curto espaço de tempo, mas com grande ênfase da aderência ao mercado. Um contraste com a prática comum do desenvolvimento tradicional "Waterfall" (ou cascata) comum às empresas de software nas décadas de 1970 e 1980, caracterizado por desenvolver a partir de uma especificação estabilizada, prosseguindo de forma linear até o final, para finalmente obter o produto ao final do processo.

Schaber et al (2001) unificaram o que é hoje conhecido como o desenvolvimento ágil, propõem o uso de diversas práticas tais como "extreme programming", "pragmatic programming", "agile modeling" e outros (não discutidos em detalhes aqui). Esta metodologia (também conhecida como metodologia ágil) usa de vários "frameworks" (estruturas de projeto), sem restrição, contudo os mais conhecidos atualmente são: Adaptive software development (ASD); Agile modeling; Agile unified process (AUP); Disciplined agile delivery; Dynamic systems development method (DSM); Extreme programming (XP); Feature-driven development (FDD); Lean software development; Kanban; Rapid application development (RAD); Scrum; Scrumban.

Destes "frameworks" o mais conhecido é o Scrum, popularizado por Schaber & Shuterland (1995) que argumentam que este traz maior velocidade e flexibilidade no desenvolvimento de produtos mais complexos e disruptivos.

Figura 5 - Esquema do processo Scrum

Fonte: MITCHEL, 2015. Wikipedia.

Nomeado a partir do jogo de Rugby, o Scrum é conduzido por times multifuncionais que conduzem o processo de desenvolvimento iterativamente (avanzando e retornando se necessário), amadurecendo durante a evolução do desenvolvimento e com liberdade (empírica) de modificar (replanejar) o escopo conforme o grau de conhecimento cresce no tempo. Contudo este 'framework' está em constante evolução visto que uma de suas premissas básicas é obter um 'produto pronto' ao final de cada ciclo (sprint), independente, inclusive, daquela versão solucionar o problema de negócio ou atividade para o qual inicialmente foi planejado (job to be done).

Blank (2011) propõe em uma visão mais ampla e radical, intitulado de "Lean Launchpad", a posicionando como metodologia para empreendedorismo, muito utilizada em "startups", que são empresas tradicionalmente de software com potencial de escalar soluções em um curto espaço de tempo, a exemplo de Facebook (EUA), Nubank (Brasil), 99 Taxi (Brasil), Pag Seguro (Brasil) e outras - tendo estas últimas três atingido a valoração de um bilhão de dólares em 2018 de acordo com Fonseca (2018). O "Lean Launchpad", portanto, é uma metodologia para ambos [a] testar e desenvolver modelos de negócio baseados (iterativamente) no questionamento e aprendizado com os clientes; [b] desenvolver o mercado (clientes); e [c] desenvolver o produto. Este framework unifica o "Business Model Canvas" proposto por Osterwalder (2014), "Customer Development" apresentado por Steven (2013) em conjunto com o Desenvolvimento Ágil (a exemplo do Scrum ou Kanban). Em resumo os projetos que usam este modelo não estão desenvolvendo um produto ou modelo de negócios, e sim buscando um modelo de negócios sustentável e escalável (o produto /software é apenas um meio para obter este fim).

Adapte sua oferta em um modelo de negócio sustentável que resulte em valor para sua organização. Para tal, use o Business Model Canvas, uma ferramenta que descreve como sua organização cria, entrega e captura valor. O Business Model Canvas e Value Proposition Canvas conectam facilmente como um plug-in permitindo que você adentre nos detalhes de como está criando valor para os clientes. (OSTERWALDER et al., 2010, p. 16, *tradução nossa*).

Figura 6 - Diagrama do "Business Model Canvas" usado no "Lean Launchpad"

Fonte: OSTERWALDER, 2010, p. 94.

Figura 7 - Diagrama do "Business Model Canvas" usado no "Lean Launchpad"

Fonte: OSTERWALDER, 2010, p. 94.

Customer Development é um framework em quatro etapas para descobrir e validar se a equipe de desenvolvimento identificou o mercado correto para o produto, desenvolver apenas as funcionalidades que solucionam problemas reais dos clientes, testar formas para adquirir e converter potenciais clientes em clientes pagantes, e dedicar os recursos necessários para crescer o negócio em escala. (COOPER & VLASKOVITS, 2010, p. 9)

Sobre o "Customer Development" proposto por Blank inicialmente nos anos de 1990, Vlaskovits & Cooper (2010) afirmam que em um formato abstrato, o que se propõe é questionar as premissas do "corebusiness" (atividade principal), aplicando metodologias de engenharia e científicas no empreendimento (mesmo que não seja necessariamente um projeto científico).

Figura 8 - Esquema dos 4 processos do "Customer Development" parte do 'Lean Launchpad'

Fonte: COOPER & VLASKIVITS, 2010, p. 9.

5 Conclusão

A revisão bibliográfica sobre as metodologias tradicionais prescritivas demonstra a busca por maior velocidade e eficiência no desenvolvimento, e ainda sucesso comercial no lançamento de novos produtos. Para tal, utilizam de vasta gama de ferramentas de design e engenharia de produtos, de filosofias como a Engenharia Simultânea e abordagens especializadas DFXs (design ou projeto otimizado para determinado objetivo) para atingir o propósito de velocidade com assertividade. Contudo estas metodologias prescritivas são orientadas a solucionar projetos com natureza fundamentalmente de alvo fixo (mercado estabelecido e tradicional). Já as metodologias ágeis, por terem natureza adaptativa, têm demonstrado resultados mais favoráveis a projetos inovadores e disruptivos (especialmente produtos digitais, mas não apenas estes), e mesmo contando a baixa taxa de sucesso principalmente da 'Startups', estas metodologias são vistas como ideais para projetos com alvos móveis (mercado dinâmico ou em construção).

Importante ressaltar que as diversas ferramentas, práticas, metodologias, ou seja o conjunto de conhecimento utilizados pelos profissionais envolvidos no projeto, não apenas designers, surgiram e evoluíram para aplicação em empreendimentos (projetos) tradicionais e pouco dinâmicos, sendo que algumas delas completarão 100 anos (a exemplo da escola Bauhaus). O que induz a seguinte reflexão: Está o conhecimento tradicional do Design desalinhado com projetos de produto inseridos em um contexto dinâmico?

O papel do Design, englobando aqui as ferramentas, metodologia e abordagem (conhecimento), terá se convertido em coadjuvante, ou está a caminhar para a obsolescência em

projetos de produto que necessitem de velocidade com adaptação vislumbrando o sucesso comercial em empreendimentos de alto risco e incertezas - mesmo que para falhar rapidamente como a própria metodologia ágil prega, visando assim minimizar os investimentos e a exposição financeira ao risco?

O distanciamento do conhecimento de Design do desenvolvimento de software (e empresas Startup) leva a ponderar inclusive sobre a eventual obsolescência da profissão ou redução da quantidade de recursos necessários para levar a cabo um projeto de produto. O avanço da tecnologia e necessidade de velocidade farão alguns designers avaliarem a aceleração do desenvolvimento de produto quando da substituição da prancheta de desenho pelas ferramentas CAD-3D, inclusive na redução drástica de envolvidos em um projeto.

É importante ponderar na adequação e atualização do modelo mental do Design englobando as abordagens, metodologias, nomenclatura, ferramentas e sobretudo velocidade e automação das mesmas, especialmente neste momento em que a IA (inteligência artificial) torna-se um pilar para a produtividade e retroalimentação (insights) nos processos de desenvolvimento de produtos e serviços, inclusive ágeis segundo Davenport (2016).

6 Referências

BECK, K.; BEEDLE, M.; BENNEKUM, A.; FOWLER, M.; MELLOR, S.; THOMAS, S. Ken.; SHUTERLAND, J.; GREENING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B. **The Manifesto for Agile Software Development**. Disponível em: < <https://www.agilealliance.org/>>. Acesso em Março de 2018.

BLANK, S. **Lean Launchpad: We have a moral obligation**. Disponível em: <<https://steveblank.com/category/lean-launchpad/>>. Acesso em Março de 2018.

BLANK, S. **The four steps to the epiphany: Sucessfull strategies for products that win**. Palo Alto: K&S Ranch, 2013.

BROWN, T. **Design Thinking: Uma metodologia poderosa para decretar o fim das velhas ideias**. Rio de Janeiro: Elsevier, 2010.

COOPER, B. VLASKOVITS, P. **The Entrepreneur's Guide to Customer Development: A "Cheat sheet" to The Four Steps to the Epiphany**. Reston: Cooper-Vlaskovits, 2010.

COOPER, R. **Winning at New Products: Creating Value Through Innovation**. New York: Basic Books, 2017.

CUSUMANO, M. **How Microsoft makes large teams work line small teams**. Cambridge: MIT Sloam Management Review, 1997.

DAVENPORT, T. **Big Data no Trabalho: Derrubando Mitos e Descobrendo Oportunidades**. Rio de Janeiro, 2014.

FONSECA, M. **Revista Exame**. Disponível em: <<https://exame.abril.com.br/pme/por-que-o-brasil-vive-uma-avalanche-de-unicornios/>>. Acesso em Março de 2018.

GAGNÉ, D.; RINGUETTE, S. **BPMN Quick Guide - Second Edition**. Disponível em: <<http://www.bpmnquickguide.com/view-bpmn-quick-guide/>>. Acesso em: Janeiro de 2018.

- LIKER, J. K.; WARD, A. C.; SOBEK, D. **Toyota's Principles of Set-Based Concurrent Engineering**. Massachusetts: MIT Sloan Management Review, 1999.
- LORENZ, C. **The Design Dimension**. Oxford: Basil Blackwell, 1986.
- OSTERWALDER, A. **The business model ontology: A proposition in a design science approach**. Lausanne: Escola Estudos Comerciais da Universidade de Lausanne, 2004.
- OSTERWALDER, A.; PIGNEUR, Y.; BERNARDA, G.; SMITH, A. **Value Proposition Design**. New Jersey: John Wiley & Sons, 2014.
- PMI. **PMBOK. A Guide to the Project Management Body of Knowledge: Fifth Edition**. Newton Square: Project Management Institute, 2013.
- RAZ, T.; GLOBERSON, S. **Effective Sizing and Content Definitions of Work Packages**. Tel Aviv: PMI, 1998.
- ROZENFELD, H.; FORCELLINI, F.; AMARAL, D.; TOLEDO, D.; SILVA, S.; ALIPRANDINI, D.; SCALICE, R. **Gestão de Desenvolvimento de Produtos: Uma referência para a Melhoria do Processo**. São Paulo: Saraiva, 2006.
- SUTHERLAND, J.; SCHWABER, K. **Guia do Scrum: Um guia definitivo para o Scrum - As regras do jogo**. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>>. Acesso em: Novembro de 2017.
- YAZDANI, B.; HOLMES, C. **Four Models of Design Definition: Sequential, Design Centered, Concurrent and Dynamic**. Coventry: Journal of Engineering Design, Vol 10, No 1, 1999.