

MELHORIA NA AGILIDADE EM MUDANÇA DE SOFTWARE COM ADOÇÃO DE MODELOS DE GESTÃO E ARQUITETURA DE TECNOLOGIA

Rodrigo Monteiro Furquim - Universidade Presbiteriana Mackenzie

Alberto De Medeiros Junior

Resumo

Este artigo aplicado relata a intervenção realizada em uma empresa que atua no setor financeiro há mais de 90 anos, tendo por foco a adoção de modelos ágeis de trabalho e a modernização da arquitetura de software na área de tecnologia da empresa para atendimento de pessoas jurídicas. Procurou-se resolver um problema de agilidade nas entregas de novos serviços financeiros para os clientes. Analisou-se como a área de tecnologia que desenvolve e mantém o aplicativo móvel e o internet banking dos clientes pessoa jurídica pode-se adaptar para se fortalecer. A situação problema caracteriza-se pela demora de tempo para oferecer aos clientes uma experiência digital relacionada aos produtos financeiros que atendam às suas necessidades sendo, melhores ou semelhantes aos que são ofertados por bigtechs ou fintechs. Por ser um modelo de trabalho inovador houve, no início, forte resistência dos gestores, mas após os dois primeiros meses do projeto, apesar de sua complexidade, demonstrou-se o potencial que o modelo poderia alcançar. O problema encontrado pela empresa, a alta concorrência e a possibilidade de utilização de novos métodos de trabalho e novas arquiteturas de software como alavanca na disponibilização de novos serviços e produtos financeiros, culminou nos resultados de agilidade apresentados neste artigo.

Palavras-chave: DevOps, Metodologia Ágil, Dispositivos móveis

Abstract

This applied article reports the intervention carried out in a company that has been operating in the financial sector for more than 90 years, focusing on the adoption of agile work models and the modernization of the software architecture in the technology area of the company to serve legal entities. It was designed to solve a problem of agility in the delivery of new financial services to customers. It was analyzed how the technology area that develops and maintains the mobile application and internet banking for corporate customers can adapt to strengthen itself. The problem situation is characterized by the delay of time to offer customers a digital experience related to financial products that meet their needs, being better or like those offered by bigtechs or fintechs. As it is an innovative work model, there was, at first, strong resistance from managers, but after the first two months of the project, despite its complexity, the potential that the model could achieve was demonstrated. The problem encountered by the company, the high competition, and the possibility of using new work methods and new software architectures as a lever for the availability of new services and financial products, culminated in the agility.

Keywords: DevOps, Agile Methodology, Mobile devices

MELHORIA NA AGILIDADE EM MUDANÇA DE SOFTWARE COM ADOÇÃO DE MODELOS DE GESTÃO E ARQUITETURA DE TECNOLOGIA

RESUMO

Este artigo aplicado relata a intervenção realizada em uma empresa que atua no setor financeiro há mais de 90 anos, tendo por foco a adoção de modelos ágeis de trabalho e a modernização da arquitetura de software na área de tecnologia da empresa para atendimento de pessoas jurídicas. Procurou-se resolver um problema de agilidade nas entregas de novos serviços financeiros para os clientes. Analisou-se como a área de tecnologia que desenvolve e mantém o aplicativo móvel e o internet banking dos clientes pessoa jurídica pode-se adaptar para se fortalecer. A situação problema caracteriza-se pela demora de tempo para oferecer aos clientes uma experiência digital relacionada aos produtos financeiros que atendam às suas necessidades sendo, melhores ou semelhantes aos que são ofertados por *bigtechs* ou *fintechs*. Por ser um modelo de trabalho inovador houve, no início, forte resistência dos gestores, mas após os dois primeiros meses do projeto, apesar de sua complexidade, demonstrou-se o potencial que o modelo poderia alcançar. O problema encontrado pela empresa, a alta concorrência e a possibilidade de utilização de novos métodos de trabalho e novas arquiteturas de software como alavanca na disponibilização de novos serviços e produtos financeiros, culminou nos resultados de agilidade apresentados neste artigo.

Palavras-chave: DevOps, Metodologia Ágil, Dispositivos móveis

ABSTRACT

This applied article reports the intervention carried out in a company that has been operating in the financial sector for more than 90 years, focusing on the adoption of agile work models and the modernization of the software architecture in the technology area of the company to serve legal entities. It was designed to solve a problem of agility in the delivery of new financial services to customers. It was analyzed how the technology area that develops and maintains the mobile application and internet banking for corporate customers can adapt to strengthen itself. The problem situation is characterized by the delay of time to offer customers a digital experience related to financial products that meet their needs, being better or like those offered by *bigtechs* or *fintechs*. As it is an innovative work model, there was, at first, strong resistance from managers, but after the first two months of the project, despite its complexity, the potential that the model could achieve was demonstrated. The problem encountered by the company, the high competition, and the possibility of using new work methods and new software architectures as a lever for the availability of new services and financial products, culminated in the agility results presented in this article.

Keywords: DevOps, Agile Methodology, Mobile devices

1. INTRODUÇÃO

Esse artigo aplicado demonstra como a área de tecnologia do Banco MDN Brasil (nome fictício), voltada a canais de internet *banking* e aplicativos móveis para os clientes “pessoa jurídica”, vem se adaptando a constantes mudanças no ciclo de desenvolvimento de software, visando responder de forma mais rápida as inovações trazidas pelos novos entrantes no mercado financeiro.

Alinhado com ações estratégicas do banco identificou-se o desejo dos clientes de terem ao alcance da sua mão uma experiência cada vez mais digital e produtos cada vez mais personalizados para suas necessidades, onde o banco passa a não concorrer apenas com seus concorrentes tradicionais, mas sim com as grandes empresas de tecnologia, *fintechs* e novos bancos digitais.

No início dos anos 20, vivencia-se um destes momentos, onde o cliente passa a comparar suas experiências digitais de serviços não financeiros com suas experiências em serviços financeiros, onde empresas mais novas que entraram nesse mercado estão sendo capazes de oferecer essa experiência de forma mais satisfatória. Para essa experiência digital a área de tecnologia teve que se adaptar a esses novos tempos e assim responder as necessidades dos clientes de forma mais rápida e inovadora, por se tornar fundamental a implementação da estratégia do negócio. Neste sentido a proposta da resolução do problema identificado foi mudar práticas para o ciclo de desenvolvimento aliada a propostas de arquitetura de softwares mais modernas, assim como adotadas nessas empresas mais novas.

2. CONTEXTO INVESTIGADO

Com mais de 90 anos de história o banco MDN é um dos maiores três bancos privados por valor de mercado no país, mais que R\$ 30 bilhões (MDN, 2020). A instituição está presente em todas as regiões do Brasil por meio de uma ampla estrutura, composta de agências, Postos de Atendimento Bancário (PAB's) e máquinas de autoatendimento, além de escritórios regionais, centros de tecnologia e unidades culturais.

O setor bancário é o segmento com maiores investimentos em tecnologia segundo a Federação Brasileira dos Bancos [Febraban] (2018), o que demonstra que os bancos estão preocupados com a vanguarda tecnológica e passando a oferecer aos clientes maiores comodidade para realizar as transações bancárias.

O sistema financeiro nacional tem muita capilaridade de empresas, por volta de 1.400 instituições que competem em segmentos distintos e nichos específicos. O maior número de empresas concentra-se em mercados de crédito, principalmente em cooperativas de crédito, além do sistema tradicional de competição, os bancos competem também com as *startups*.

Segundo o relatório *Fintech Mining Report* (Sebrae, 2019), as *startups* delimitadas em seu relatório foram selecionadas a partir de um trabalho minucioso de pesquisa e consulta ao banco de dados do Distrito, que é uma holding de empresas de negócios voltadas a inovação e provê informações públicas bem relevantes, assim como Liga Ventures, Associação Brasileiras de Startups, StartSe, DisruptBox e LinkedIn.

O relatório mostra uma distribuição de quantidade de *fintechs* por setor e o crescimento da quantidade de *fintechs* entre 2015 e 2019, porém apesar de existir 553 *fintechs* e uma evolução de 56 a 553, ou seja, quase dez vezes mais nos últimos quatro anos, é demonstrado como os bancos brasileiros têm atuado para alavancar novos negócios e oportunidades em parceria com as *fintechs*, reforçando que não as considera somente como ameaças, mas sim explorando oportunidades. Outro fator importante apontado é a supervisão dos órgãos reguladores, conforme for acontecendo à evolução das *fintechs*. Neste sentido para se evitar um risco sistêmico, as autoridades reguladoras exigem requisitos mínimos de

atendimento, como por exemplo, o Fundo Garantidor de Crédito, ter provisões conservadoras, atender a risco de capital em relação a crédito e liquidez, entre outras, fatores já em cumprimento pelos bancos.

Ao fazer uma breve análise do Banco MDN sob a ótica da Matriz *SWOT* (acrônimo em língua inglesa para *Strengths*, *Weakness*, *Opportunities* e *Threats*), essa ferramenta busca identificar as forças, fraquezas, oportunidades e ameaças que permeiam a empresa e o ambiente em que está inserida. “O objetivo dessa análise é gerar informações importantes para tornar a organização mais consciente de suas possibilidades futuras” (Nogueira, 2014, p.46). Destacam-se os seguintes principais elementos:

- **Força (*Strengths*):** os pontos positivos da organização, são justamente aqueles que fazem do MDN uma opção bancária diferenciada em relação aos seus concorrentes. No MDN, os mesmos serviços são oferecidos em diversos canais, sejam eles digitais ou presenciais por sua rede de agências. Proporciona-se, portanto, flexibilidade ao cliente para escolher onde quer ser atendido, já que esse pode ter acesso ao banco em qualquer lugar, da forma que ele escolher e a qualquer momento. A marca do banco MDN traz uma tradição de mais de 90 anos de existência, atualmente o banco com o maior número de clientes no país.
- **Fraquezas (*Weakness*):** assim como seu tempo de história, o banco MDN traz consigo uma grande quantidade de sistemas legados que trazem dificuldades a inovação e que junto com o tamanho do banco acaba por trazer aumento total do *lead time* de entregas de valor aos clientes, pois espera-se o amadurecimento das propostas dentro do fluxo de trabalho atualmente instituído.
- **Oportunidades (*Opportunities*):** com a tendência de aumento do uso das tecnologias digitais nas tarefas bancárias, as pessoas atualmente “desbancarizadas”, ou seja, que atualmente não possuem conta em banco, estejam aptos a abrir uma conta no banco, de forma digital ou não. Aumenta-se, portanto, o índice de clientes potenciais. A manipulação de *Big Data* para entender a jornada do cliente e as necessidades dos clientes para realizar ofertas contextualizada e eficientes é importante para maiores taxas de retorno e redução de gastos com marketing ineficaz.
- **Ameaças (*Threats*):** a política regulatória do Banco Central e outros órgãos reguladores podem representar um risco. Apesar de ter feito todas as adequações necessárias para cumprir as exigências da instituição, o órgão pode tornar mais flexível suas regras no futuro, o que facilitaria novos entrantes que deveriam cumprir menores exigências. Além do crescimento elevado de *fintechs* nos últimos anos, nota-se um avanço de empresas grandes de tecnologia para o setor financeiro, assim como Google, Amazon, Apple, Alibaba e Facebook entre outras.

2.1 Caracterização do problema

O problema a ser tratado neste artigo aplicado caracteriza-se pela demora de tempo para oferecer aos atuais e aos novos clientes uma experiência digital relacionada aos produtos financeiros que atendam suas necessidades: melhores ou semelhantes aos que são ofertados por *bigtechs* ou *fintechs*. Para isso, deve-se analisar como a área de tecnologia que desenvolve e mantém o aplicativo móvel e o internet *banking* dos clientes pessoa jurídica, denominadas empresas, pode-se adaptar a fim de se fortalecer, entregando uma experiência similar aos usuários à de grandes empresas de tecnologia. Tratar dessa área de tecnologia e propor melhorias significa afetar o dia a dia de mais de 1 milhão de empresas que já utilizam esses canais para se relacionar com o banco.

O impacto da TI no modo estratégico de uma organização baseia-se, conforme modelo de Nolan e McFarlan (2005), em duas questões estratégicas: a primeira é quanto a empresa

depende de custo-benefício, ininterrupto, seguro, para operar sistemas de tecnologia sem problemas (o que é referido como TI "defensivo")? A segunda é: quanto a empresa confia na TI para obter vantagem competitiva, com sistemas que forneçam novos serviços e produtos de valor agregado ou alta capacidade de resposta aos clientes (TI "ofensivo")? Essas questões originaram quatro modos de operação de tecnologia.

- **Modo Suporte:** Mesmo com interrupções repetidas de serviço de até 12 horas, não há consequências graves, o tempo de resposta do usuário pode levar até cinco segundos com transações online, os sistemas internos são quase invisíveis para os fornecedores e clientes, a empresa pode reverter rapidamente para procedimentos manuais cerca de 80% das transações de valor e o trabalho dos sistemas é principalmente de manutenção.
- **Modo Produção:** se os sistemas falharem por um minuto ou mais, haverá uma perda imediata de negócios, a diminuição do tempo de resposta além de um segundo tem sérias consequências para os usuários internos/externos, a maioria das atividades principais de negócios está online, o trabalho dos sistemas é principalmente de manutenção, o trabalho em sistemas fornece pouco apoio estratégico e há diferenciação ou redução drástica de custos.
- **Modo Transição:** novos sistemas prometem grandes transformações em processos e serviços, eles prometem também grandes reduções de custo além de apresentarem desempenho significativo em custos, serviços ou processos em relação aos concorrentes, a TI constitui mais de 50% dos gastos de capital e representa mais de 15% do total das despesas corporativas
- **Modo Estratégico:** se os sistemas falharem por um minuto ou mais, há uma perda imediata de negócios, a diminuição do tempo de resposta além de um segundo tem sérias consequências para os usuários interno/externos, os novos sistemas prometem grandes transformações em processos e serviços, eles prometem grandes reduções de custo e apresentarão avanços significativos no desempenho do processo em relação aos concorrentes.

Com isso, pode-se observar que o diagnóstico é que o banco MDN tem um alinhamento de TI de modo estratégico, pois percebe-se o que existe uma alta dependência de tecnologia com a alta necessidade de informação para o negócio funcionar e assim, cada vez que se melhorar sua capacidade de entrega e a qualidade do aplicativo, oferecendo uma experiência superior ou similar as empresas de tecnologias para que o cliente não migre para o concorrente.

2.2 Questões político-estratégicas

Por uma questão de escolha estratégica deste artigo aplicado que foi implementado na organização, limitou-se a propor melhorias na cadeia de valor, conforme demonstrado na Figura 1, em que a tecnologia tem poder de decisão. Visto que nessa cadeia existem diversas áreas de negócios, estudar e propor melhorias para cada área e convencer o *stakeholder* dessa área poderia estender demais o escopo desse trabalho e não ser conclusivo pelos diversos interesses de cada *stakeholder*.

Dentre as diversas responsabilidades, tendo que destacar as principais de cada área:

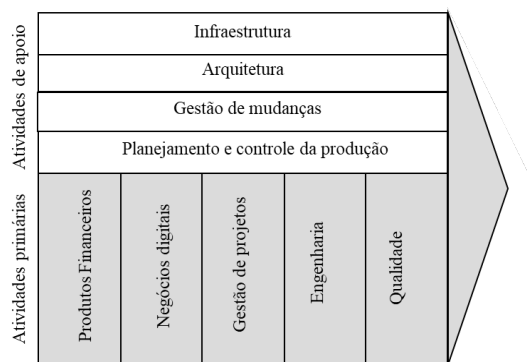


Figura 1 - Cadeia de valor para desenvolvimento em canais PJ.

Fonte: Adaptada de Porter (1991).

- **Produtos Financeiros:** responsável por entender as necessidades dos clientes, pesquisar e interagir com o mercado, modelar um novo produto e acompanhar a captura de valor.
- **Negócios Digitais:** responsável por definir a jornada do cliente, além do canal que o produto será disponibilizado, também fica a cargo de montar um documento que reúne as informações necessárias que permite aos executivos avaliarem se o projeto é viável ou não, chamado de *business case*, conseguir aprovação do projeto e para entrar na lista de projetos a serem executados.
- **Gestão de projetos:** responsável por organizar as entregas dos projetos, conseguir a alocação dos recursos necessários para execução do projeto, monitorar e controlar a execução do projeto.
- **Engenharia:** responsável por desenvolver os projetos, contratar as pessoas, gerenciar e acompanhar os custos. Essa área é composta por 190 profissionais, 82% sendo profissionais contratados do próprio MDN e 18% de empresas de tecnologias parceiras e prestadores de serviços de desenvolvimento de *software* e gestão.
- **Qualidade:** responsável por definir as práticas e políticas de teste e qualidade de software.
- **Planejamento e controle da produção:** responsável por definir a capacidade em horas das comunidades e demonstrar os projetos que devem ser executados por estarem aprovados na lista única de projetos.
- **Gestão de Mudanças:** responsável por disseminar a visão integrada de mudanças nos sistemas, além de autorizar ou desautorizar implantações de infraestrutura e de software conforme o risco associado essas mudanças.
- **Arquitetura:** responsável por definir e direcionar quais padrões, práticas, ferramentas e peças estruturais de software é necessário para atender as necessidades do projeto.
- **Infraestrutura:** responsável por validar a capacidade computacional da empresa, disponibilizar recursos para os ambientes de desenvolvimento, testes e produção com a finalidade de manter o funcionamento adequado dos aplicativos.

As intervenções aqui sugeridas, por conta das alçadas nas tomadas de decisão, são restritas às áreas de tecnologia, apresentada aqui na cadeia de valor como: Gestão de Projetos, Engenharia e Qualidade, ou seja, qualquer problema de produtividade nas áreas de Produtos Financeiros e Negócios Digitais, podem ser sugeridas, mas não serão apresentadas neste contexto.

3. DIAGNÓSTICO DA SITUAÇÃO - PROBLEMA

Para se adotar um procedimento para diagnóstico, faz-se necessário entender o que é desempenho de entrega de software que, segundo Forsgren (2018, pp. 48-49):

[...] é a mensuração do desempenho deve ter duas características, a primeira baseia-se que o desempenho deve ser concentrar em um resultado global, para garantir que as equipes não se enfrentem, como por exemplo, enquanto existem interesses em entregas rápidas de software, também existe o interesse de se manter a estabilidade dos sistemas, se for considerado desempenho de uma área e não de outra, começam as disputas. A segunda, medida do desempenho, deve focar-se nos resultados e não na produção, ou seja, a medida não deve recompensar as pessoas por colocarem uma grande quantidade de atividades que adicionem pouco valor ao software e não ajuda a empresa atingir seus objetivos.

Diante disso, para Forsgren (2018, pp 52-54), foram analisados quatro indicadores de desempenho para entender se existia um indício de problema.

1. **Lead Time:** é o tempo que leva desde que o cliente faz a solicitação até que essa solicitação seja atendida. No contexto de desenvolvimento de software, pode ser classificado em duas etapas: a) o tempo que leva para projetar e validar um produto ou funcionalidade e, b) o tempo que leva para o trabalho ser implementado, testado e entregue.
2. **Batch Size:** é a quantidade de software que se faz necessário implantar para a entrega de um produto (tamanho do lote). A redução do seu tamanho, reduz os tempos de ciclo de entrega de produto e a variabilidade no fluxo, reduz o risco e a sobrecarga e melhora a eficiência. Para se medir o tamanho do lote, utiliza-se a métrica frequência de implantação (*Deployment Frequency*).
3. **Mean Time to Restore (MTTR):** é o tempo necessário para se restabelecer os serviços, medidos pela quantidade de incidentes e o tempo que esses incidentes são resolvidos.
4. **Change Fail Percentage:** é o percentual de falha durante as mudanças de software, sejam elas mudanças de software ou infraestrutura. Esse indicador tem sua importância por demonstrar a degradação do serviço e com isso pode exigir correção, seja por um *hotfix* (correção em produção) ou ainda *rollback* (reversão da versão do software).

De acordo com a pesquisa apresentada por Forsgren (2018), realizada em 2017, as empresas são avaliadas e classificadas como de Alto Desempenho, Médio Desempenho e Baixo Desempenho de acordo com as variações dos quatro indicadores, demonstrados na tabela 1:

Tabela 1 Desempenho na entrega de software.

Pesquisa em 2017	Alto Desempenho	Médio Desempenho	Baixo Desempenho
<i>Deployment Frequency</i>	Por demanda (várias vezes ao dia)	Entre uma vez por semana e uma vez por mês	Entre uma vez por semana e uma vez por mês
<i>Lead Time</i>	Menos de uma hora	Entre uma semana e um mês	Entre uma semana e um mês
<i>MTTR</i>	Menos de uma hora	Menos de um dia	Entre um dia e uma semana
<i>Change Failure Rate</i>	0-15%	0-15%	31-45%

Nota. Adaptado de Forsgren (2018).

A análise dos dois primeiros indicadores apresentados por Forsgren (2018), no contexto atual dessa área de tecnologia, foi o que evidenciou o problema de agilidade, que pode ser olhada pelo tempo médio entre as mudanças e o volume ou quantidade de mudanças que aconteceram no sistema. Nos cinco primeiros meses de 2020 onde existiram implantações de novas funcionalidades, seja no *internet banking* ou aplicativo do dispositivo móvel, onde a demanda já passou por quase toda a cadeia de valor e agora é o tempo de desenvolvimento, testes até a implantação e ficar disponível para o cliente utilizar, teve um número muito alto de tempo entre as mudanças dos sistemas, ficando em média uma mudança a cada 11 dias.

Quando identificada a quantidade de modificações do sistema, seja decorrente de implantações de projetos ou de novas funcionalidades, verifica-se que nos cinco primeiros meses do ano de 2020, aconteceram em média 25 mudanças de software por mês, mostrando equivalência ao indicador *Deployment Frequency* e se analisado apenas com base nesse indicador, segundo Forsgren (2018), poderíamos classificar como Baixo ou Médio Desempenho. Demonstrado na figura 2.

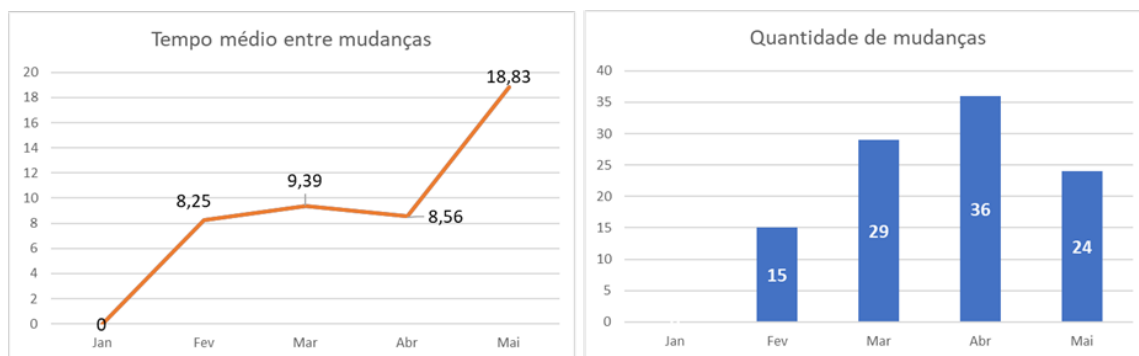


Figura 2 - Quantidade de mudanças e tempo médio no período em 2020.

Fonte: Adaptado pelos autores de Banco MDN (2020).

A análise do *Lead Time*, demonstra um tempo médio ainda maior comparado ao tempo médio entre mudanças, tendo a média de 40 dias para se completar o ciclo todo de desenvolvimento até a implantação, isso se dá devido ao fato de você ter várias equipes que trabalham no desenvolvimento do sistema de forma paralelo em histórias diferentes, então enquanto uma equipe está refinando as histórias, a outra já está em processo de homologação, por exemplo. Se analisado apenas com base nesse indicador, segundo Forsgren (2018), poderíamos classificar como Baixo ou Médio Desempenho, conforme observado na figura 3.

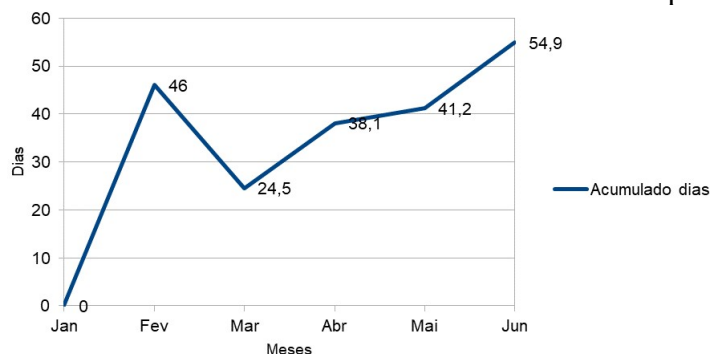


Figura 3 - Lead Time (médio) das entregas das histórias.

Fonte: Adaptado pelos autores de Banco MDN (2020).

Já a análise dos dois últimos indicadores apresentados por Forsgren (2018), evidência que não existe um problema de qualidade, o que demonstra uma estabilidade dos serviços para os clientes e que não é por esse fato que as equipes estão desperdiçando tempo com retrabalhos e cuidados com a operação do sistema e assim não conseguindo fazer novas

entregas. O resultado do indicador *Change Failure Rate* (índice de falhas em mudanças), em seu maior pico em maio, foi de 13% ficando em média em torno de 10% e o resultado do indicador *MTTR* (tempo médio para se reestabelecer um serviço) não passou de uma hora, tendo um caso isolado abril de 2h24min conforme a figura 4. Esses indicadores se comparados aos critérios definidos por Forsgren (2018), caracteriza como empresa de alto desempenho.

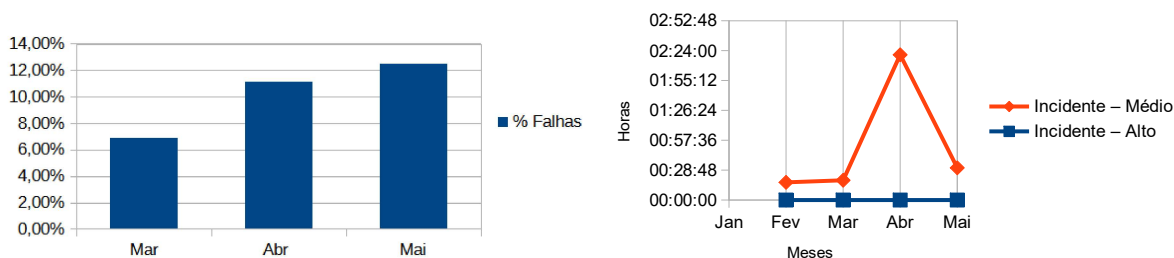


Figura 4 – Índice de mudanças com falhas e tempo para restabelecer o serviço (MTTR).
Fonte: Adaptado pelos autores de Banco MDN (2020).

Em resumo, os indicadores demonstram que existe um problema quando se refere a velocidade de entrega, mas não existe um problema quando a relação é estabilidade ou qualidade dos sistemas.

Em relação às arquiteturas de aplicativos, o banco MDN precisa modernizar os aplicativos para atingir os princípios das agilidades. Para apoiar as empresas que precisam inovar, as equipes de desenvolvimento devem fornecer continuamente serviços de software a uma velocidade maior. Isso foi reconhecido por empresas da internet como Netflix e Amazon, que mudaram suas arquiteturas de software para suportar a necessidade de inovação contínua – essencialmente usá-los para redefinir os mercados em que operam.

A arquitetura do software atualmente do banco MDN, é referida como aplicativo monolítico que, em engenharia de software descreve um único aplicativo de software em camadas no qual a interface de usuário e código de acesso aos dados são combinados em um único programa a partir de uma única plataforma e qualquer alteração no sistema envolve a construção e a implantação de uma nova versão do aplicativo do lado do servidor (Fowler, 2015).

As arquiteturas de software definidas como monolitos, impedem ciclos de desenvolvimento ágil, fazendo com que as equipes de desenvolvimento utilizem um modelo mais tradicional de desenvolvimento, conhecido como *Waterfall*. Com métodos ágeis, as organizações podem iterar rapidamente para apoiar a inovação, mas as equipes também estão mudando as arquiteturas de aplicativos para melhorar flexibilidade de software e ajudar a acelerar a implantação. (Ravichandran, 2016).

3.1 Problema definido

Como resultado, foi identificado um longo tempo para ocorrer uma mudança de software com o objetivo de lançar um novo produto ou nova funcionalidade comparado aos concorrentes, devido a arquitetura do software, podendo ocasionar aos clientes a busca por outras opções de soluções financeiras.

3.2 Objetivos

O objetivo geral é evoluir a arquitetura do software e adequar os processos da cadeia de valor para que sejam mais eficientes com o intuito de realizar entregas mais rápidas do aplicativo móvel aos clientes, sem renunciar à qualidade das entregas sustentando vantagem

competitiva. Há também expectativas complementares ao objetivo geral, que resolvem mais questões para a organização, como:

- Aumentar o nível de automação nos processos da cadeia de valor, fazendo com que o *lead time* das atividades sejam reduzidos.
- Adotar métodos e ferramentas novas de trabalho que tragam agilidade nos processos atuais do desenvolvimento de software, principalmente no gerenciamento do ciclo de vida do software, fazendo uso das práticas de *DevOps*.
- Diminuir custos da operação de tecnologia, quando possível.

4. INTERVENÇÃO PROPOSTA

Considerando-se as causas identificadas como geradoras do problema na fase de diagnóstico e os objetivos pretendidos com a solução do problema evidenciado, a proposta de solução foi desenvolvida buscando alternativas que gerassem valor percebido na cadeia de valor do processo de lançamento de uma nova funcionalidade. Assim, deverá entregar a evolução necessária das plataformas para se atingir tal objetivo.

A partir dos indicadores que demonstram a necessidade de se lançar uma funcionalidade mais rápida aos clientes, a equipe coordenada por um dos autores deste artigo aplicado iniciou um estudo para obter-se cenários possíveis (dentro das limitações de orçamento, escopo e prazo que a área teria à disposição). Diante deste quadro apresentaram-se dois cenários possíveis, detalhados a seguir.

a) Cenário 1.

O primeiro cenário, tratado neste artigo aplicado como “cenário 1”, propõe a evolução tecnológica apenas da frente de *internet banking*, onde a avaliação passou pela decisão de modernizar o aplicativo mais antigo da área e é o mais utilizado pelos clientes atualmente.

Para realizar essa modernização, a equipe coordenada por um dos autores deste artigo aplicado, avaliou a utilização das técnicas de arquitetura de software conhecida como *micro frontend*. Fowler (2019) define como um estilo de arquitetura em que as entregas dos aplicativos de *frontend* são feitos de forma independente e juntos compõem um todo maior.

Alguns dos principais benefícios que a técnica de *micro frontends* oferecem são:

- bases de código menores, mais coesas e sustentáveis;
- organizações mais escalonáveis com equipes independentes e autônomas;
- a capacidade de atualizar, atualizar ou até mesmo reescrever partes do *frontend* de uma forma mais incremental do que era possível anteriormente.

Todos esses benefícios alcançados por essa técnica podem proporcionar uma diminuição no *lead time* das entregas, pois as equipes de desenvolvimento passam a depender apenas de sua autonomia para realizar a entregas de suas partes de software em produção.

Demonstra-se a utilização de *micro frontend* como uma técnica que as organizações estão adotando, apontado no relatório da ThoughtWorks (2020). Esse relatório denominado Radar, é um documento que define as mudanças que são consideradas atualmente relevantes no desenvolvimento de software — tendências que se acredita que as empresas devem prestar atenção e considerar usar em seus projetos. Reflete a opinião peculiar de um grupo de tecnólogos experientes e é baseado no trabalho e experiências da empresa. Embora seja algo interessante, ele não deve ser tomado como uma análise profunda de mercado.

b) Cenário 2.

O segundo cenário, tratado neste artigo aplicado como “cenário 2”, propõe a evolução tecnológica apenas do aplicativo móvel, onde a avaliação passou pela decisão de modernizar

aplicativo mais promissor, pois os clientes estão migrando a utilização do internet *banking* para utilização dessa tecnologia.

Para realizar essa modernização, avaliou-se a utilização das técnicas de modularização do aplicativo, conforme a própria definição no site da Google, responsável pelas diretrizes da plataforma Android, define a modularização dos aplicativos como um processo de separação de componentes lógicos de um projeto de aplicativos em módulos discretos e essa reorganização da funcionalidade do aplicativo nesses componentes discretos oferecem os seguintes benefícios:

- **Desenvolvimento paralelo:** com a separação de componentes lógicos do seu app em módulos, diferentes equipes ou indivíduos da sua organização podem se apropriar de cada módulo e trabalhar neles com menos conflitos de mesclagem ou interrupções em outras equipes. Além disso, se você tiver lógica usada em várias partes do app, poderá usar módulos de biblioteca para promover a reutilização e o encapsulamento de código.
- **Melhoria no tempo de compilação:** quanto mais modular for seu projeto, mais significativa será a melhoria no desempenho da compilação, pois cada módulo terá menos linhas de código para compilação.
- **Personalização da entrega de recursos:** a modularização dos recursos do app como módulos permite fazer implantações sob demanda, condicional e instantânea, sendo possível entregar aos clientes apenas uma parte nova do aplicativo ao invés do aplicativo todo.

4.1 Solução adotada

Depois de algumas discussões e reuniões sobre a exploração dos cenários, se dependesse apenas da vontade da equipe de projetos e os patrocinadores desse trabalho, seria escolhido o cenário 1, mas essa decisão teria o viés de escolha baseado no empirismo dos participantes de tecnologia para realizar o trabalho e pode não ser o melhor caminho a ser escolhido, assim como Kahneman (2012) cita que mesmo a consciência do efeito de vieses pouco serviu para melhorar a qualidade de decisões de negócios.

Diante disso, a equipe coordenada por um dos autores deste artigo aplicado, sugeriu estabelecer alguns critérios para seleção entre os cenários, até com o objetivo de deixar claro e evitar vieses nas escolhas. Como método de decisão escolheu-se basear no conceito de tríplice restrição de projetos conforme os conceitos do PMI (2017), as restrições do projeto podem ser definidas como limitações que comprometem a execução de um trabalho. São situações impostas por alguém ou por um contexto que podem afetar o desempenho e até o resultado de um projeto, normalmente as principais restrições de um projeto são o prazo, o escopo e custos.

Com base nessas definições imagina-se quais seriam as limitações de custo e prazo para realização do projeto e tenta-se atingir a maior completude de escopo possível, sendo o escopo desse trabalho realizar as modernizações do aplicativo do internet *banking* e do aplicativo móvel conforme os cenários 1 e 2.

Considerando-se as prioridades do escopo da modernização dos aplicativos, se tiver que definir prioridades dados as outras limitações de prazo e custo, será priorizado a completude da modernização de uma plataforma em detrimento de outro, desta maneira, se tiver que realizar a modernização 100% de uma plataforma e 0% da outra, essa poderá ser uma escolha, pois para colher os benefícios descritos em cada cenário, faz-se necessário o máximo de completude dele e se ainda precisar, será priorizado primeiro a modernização do aplicativo móvel e depois a modernização do internet banking, pois percebe-se uma mudança do uso dos clientes do internet *banking* para o uso dos aplicativos móvel.

As limitações de prazo foram definidas em acordo com as equipes de tecnologia considerando o que poderia ser realizado para o ano atual 2020. Para as limitações de orçamentos ou custo, imaginou-se que as demandas de negócio não deveriam ser interrompidas para fazer as evoluções tecnológicas necessárias, com isso qual seria o investimento necessário para ser ter apenas uma equipe *Scrum*, conforme definidos na seção 3, que trabalhando até o final do ano de 2020 fosse capaz de modernizar as plataformas.

Os custos foram definidos conforme média histórica de uma equipe *Scrum* trabalhando por seis meses, considerando uma equipe de até 10 pessoas a uma taxa hora de R\$ 200,00 por uma quantidade média de horas produtivas no mês (168h ou 21 dias) durante seis meses de trabalho. Com essas definições as seguintes restrições são demonstradas na Tabela 2.

Tabela 2 - Tríplex restrição da modernização.

Escopo	Prazo	Custo
Maior completude de modernização de uma ou duas plataformas (<i>internet banking</i> e aplicativo móvel)	Seis meses	R\$ 2M

Nota. Adaptado pelos autores (2020).

Baseado nas definições, criou-se o conceito do total de capacidade de entrega, que é uma estimativa da quantidade de software que essas restrições definidas são capazes de entregar baseadas nas horas que terá em desenvolvimento em cada cenário, ou seja, o total da capacidade de entrega da equipe em horas seria a quantidade de pessoas pelo tempo que elas estarão no projeto, conforme Tabela 3.

Tabela 3 - Capacidade total de entrega.

Horas Média/Mês	Pessoas	Meses	Total
168	10	6	10.080

Nota. Adaptado pelos autores (2020).

Isso significa que o investimento atual dará uma capacidade máxima de 10.080h para entregar o escopo escolhido.

O cenário 1, que tem como escopo a modernização do *internet banking*, com 1680 telas aproximadamente, esse levantamento foi feito baseado nos códigos fontes em maio/2020. Para adequar esse volume de telas para modernizar foi estimado um tempo média de criação de cada tela em 10 horas.

O cenário 2 que tem como escopo a modernização aplicativo móvel com 48 módulos, esse levantamento foi realizado em maio/2020, baseada nas funcionalidades disponíveis no aplicativo. Para adequar todas essas funcionalidades foi estimado um tempo média de criação de cada módulo em 245 horas.

Com as definições de escopo e o tempo médio para completar esse escopo, obtém-se o total de horas estimadas para realização do investimento, aplica-se a divisão pela capacidade total disponível (10.080h) e sabe-se o percentual de completude de cada cenário.

Em resumo, a Tabela 4 demonstra o percentual completude de cada cenário.

Tabela 4 - Quadro resumo dos cenários avaliados.

Cenário	Tempo Médio	Total Horas	% Completude
1-Modernização do <i>internet banking</i> (1680 telas)	10h por tela	16.800h	60%
2- Modernização do aplicativo móvel (48 módulos)	245h por módulo	11.520h	87%

Nota. Elaborado pelos autores (2020).

Diante dos números apresentados nos cenários, optou-se para esse ano de 2020 priorizar o cenário 2, que trata da modernização do aplicativo móvel, por atingir a maior completude de escopo dentro das definições de restrições já descritas nesse tópico. Ressalte-se que essas estimativas não consideram os investimentos necessários em *DevOps*, uma vez que esses

investimentos já foram realizados em 2019 em outras frentes de trabalho e já estão disponíveis para uso para qualquer área de tecnologia do banco MDN.

4.2. Resultados esperados

Neste sentido, a realização da modularização do aplicativo móvel para clientes de pessoa jurídica criará valor em toda a cadeia, pois espera-se diminuir o tempo na demora da entrega de novas funcionalidades, gerando sustentabilidade do negócio, captura de valor e vantagem competitiva de mercado.

A partir dos dados analisados, espera-se que, pela adoção da proposta apresentada, na próxima avaliação obtenha-se melhoras nos índices de desempenho nas entregas de software, especialmente os índices de *Deployment Frequency* e *Lead Time*. Ao mesmo tempo, espera-se que a solução adotada resulte em aumento da credibilidade do banco MDN junto aos clientes, que podem ser avaliados com maiores adesões ao uso do aplicativo móvel, junto com demais ações tratados em outras frentes de trabalho no banco MDN.

4.3. Ações

O processo de intervenção iniciou-se tão logo foram compilados os dados obtidos de acordo com os indicadores de velocidade e qualidade. Diante do problema definido, a equipe coordenada por um dos autores desse artigo aplicado juntamente com as demais equipes da engenharia de desenvolvimento do aplicativo móvel envolvidas definiu que alguma estratégia de mudança deveria ser implementada para realizar a modularização do aplicativo móvel e assim diminuir o tempo entre os lançamentos de novas funcionalidades aos clientes sem impactar as entregas em andamento das atuais demandas de negócio.

Nesse sentido e de acordo com French e Bell (1995, p. 65):

[...] intervenção define-se como um conjunto de atividades estruturadas nas quais participa um grupo selecionado de unidades organizacionais ou equipes de trabalho, através do desempenho de uma tarefa ou sequência de tarefas, cujos objetivos se relacionam direta ou indiretamente com o aperfeiçoamento da organização.

A estratégia das mudanças sugeridas aqui, de acordo com a equipe coordenada por um dos autores deste artigo aplicado, baseou-se em três pilares: planejamento, modelo de trabalho, monitoramento e divulgação dos resultados.

4.4. Planejamento

Nesse pilar foi realizada a definição de quais funcionalidades do aplicativo móvel seriam executadas primeiro, de acordo com o nível de utilização, que foi apurada com a ferramenta *Google Analytics*. Definiu-se também o conceito de “entrega”, onde a equipe coordenada por um dos autores desse artigo aplicado foi responsável por modularizar a funcionalidade, aplicar os testes necessários para garantir a qualidade, implantar a funcionalidade em produção e acompanhar o pós implantação.

As funcionalidades foram confirmadas com as demais equipes da engenharia, e qual *squad* se responsabilizaria para a evolução das funcionalidades nos módulos após a modularização dessa funcionalidade. Outro ponto importante do planejamento foi identificar com essas *squads* quais funcionalidades seriam alteradas por demandas de negócio e planejar a modularização nesse mesmo trabalho quando possível. Nesse momento houve descoberta de novas funcionalidades não mapeadas no *Google Analytics* e uma última consideração levada no planejamento foi avaliar o crescimento das equipes que se daria pelos novos *squads* devido a ter novas funcionalidades previstas. Nesse ano de 2020 foi constatada toda a criação das funcio-

nalidades do PIX, novo modelo de pagamentos e transferências, resultando em uma quantidade total de 58 funcionalidades distribuídas por *squad*.

4.5. Modelo de trabalho

Esse pilar teve como propósito ofertar um modelo de trabalho em que as modularizações das funcionalidades do aplicativo móvel acontecessem de forma que não causassem impactos nas demandas de negócios e que fosse capaz de conviver com os modelos ágeis de trabalho das equipes conforme descrito no item 3. Para atingir a efetividade no resultado desse trabalho assumiu-se as premissas:

- Todas as novas funcionalidades no aplicativo móvel das outras equipes da área já seriam desenvolvidas de forma modularizada e seguindo os padrões de arquitetura de software estabelecida pela equipe por um dos autores deste artigo aplicado, dessa forma, evitando que o “estoque” de funcionalidades não modularizadas aumentasse.
- A equipe de modularização deveria garantir a “entrega” do módulo e a passagem de conhecimento, aqui conhecimento considera-se a mudança de arquitetura, correções de bugs encontrados e a criação das esteiras *DevOps* para implantação, para a equipe que recebeu o software pronto.

Encontraram-se no modelo ágil duas cerimônias essenciais para estabelecer esse novo modelo de trabalho, a primeira foi a *planning*, onde a equipe responsável pela funcionalidade juntamente com a equipe de modularização tinha como objetivo, a apresentação da funcionalidade, que acontecia com uma simulação de um caso real de cada funcionalidade a ser modularizada, com as devidas explorações de cenários básicos e alternativos possíveis, uma visão geral de como se testa essas funcionalidades e a quantidade de testes aplicados e se havia bugs registrados para essas funcionalidades. A segunda foi a cerimônia de demonstração, onde a equipe de modularização coordenada um dos autores deste artigo aplicado devolvia para a equipe da funcionalidade, com uma passagem de conhecimento a apresentação da nova arquitetura do módulo, as informações de intercorrências que aconteceram durante o processo de modularização e a demonstração do resultado da esteira *DevOps* para aquele módulo, exemplificado pela figura 5.

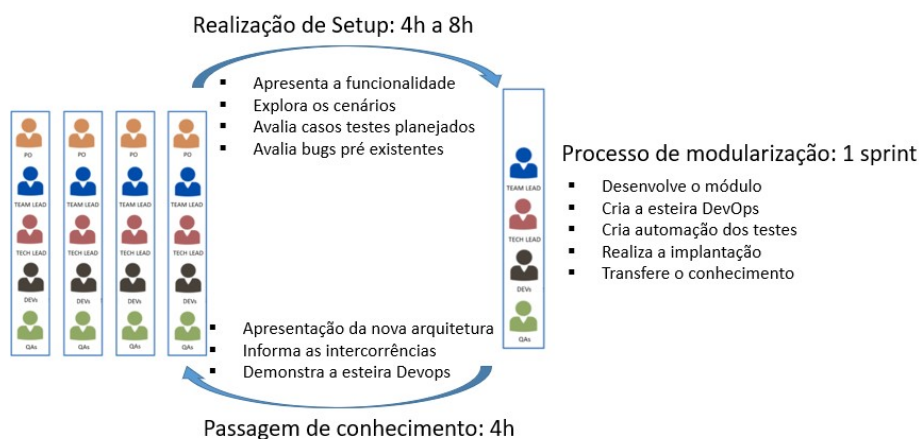


Figura 5 - Modelo de trabalho para modularização.

Fonte: Elaborado pelos autores (2020).

4.6. Monitoramento e divulgação dos resultados

Esse último pilar teve como propósito demonstrar como o resultado seria acompanhado e a divulgação desses resultados a todos os interessados. Os principais itens que serão divulgados são:

1. A evolução das funcionalidades modularizadas já consideradas como “entregas”, com a comparação do planejado em relação ao trabalho realizado, principais pontos de impedimentos com cada responsável por atuação.
2. A velocidade da esteira de *DevOps* para cada módulo.
3. A quantidade de defeitos encontrados e corrigidos durante a modularização.

Até o momento da escrita desse artigo aplicado a evolução das funcionalidades modularizadas já estava em 92% do trabalho realizado, conforme a figura 6.

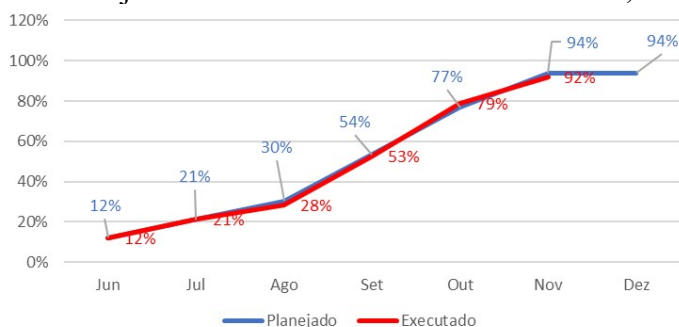


Figura 6 - Evolução das entregas das funcionalidades modularizadas.

Fonte: Adaptado pelos autores de Banco MDN (2020).

Com essa evolução dos módulos já foi possível aferir resultados coerentes no qual se tinha como objetivo, assim como a velocidade de entregas de novas funcionalidades, onde para cada funcionalidade existia uma ocupação de tempo de dois dias em média para a execução dos testes de forma manual, com a criação da esteira *DevOps* e a automação dos testes para os módulos criados, os testes agora de forma automatizada passaram a executar em até uma hora.

Ressalte-se que na automação destes testes, foram encontrados defeitos nas funcionalidades e quando avaliados já se encontravam impactando os clientes. De todos os defeitos encontrados, até esse momento, podem-se aferir 12 defeitos graves que impossibilitava ao cliente de concluir as principais operações no aplicativo móvel. Na tabela 6 é possível identificar os ganhos obtidos até o momento.

Tabela 5 - Resultado preliminares.

Item	App Principal (baseline inicial)	Atualmente
Média do tempo da esteira <i>DevOps</i>	46 minutos	12 minutos
Execução dos testes regressivos	2 dias	1 hora
Quantidade de defeitos resolvidos		12

Nota. Adaptado pelos autores de Banco MDN (2020).

O indicador de quantidade de mudanças (*Deployment Frequency*) foi acompanhado pela evolução da quantidade de mudanças que as equipes conseguiram realizar e pelo tempo médio entre essas mudanças. A quantidade de mudanças teve uma evolução muito significativa pois, desde o mês julho, não teve resultado menor que 60 mudanças por mês. O indicador tempo médio entre mudanças também teve evolução com redução de 18 dias em maio e chegando um pouco mais 7 dias em setembro. Dadas as limitações orçamentárias, pelo escopo desse trabalho, atingiu-se apenas a plataforma do aplicativo móvel, com limitação nas mudanças da plataforma de internet. A figura 7 demonstram a evolução desses resultados.

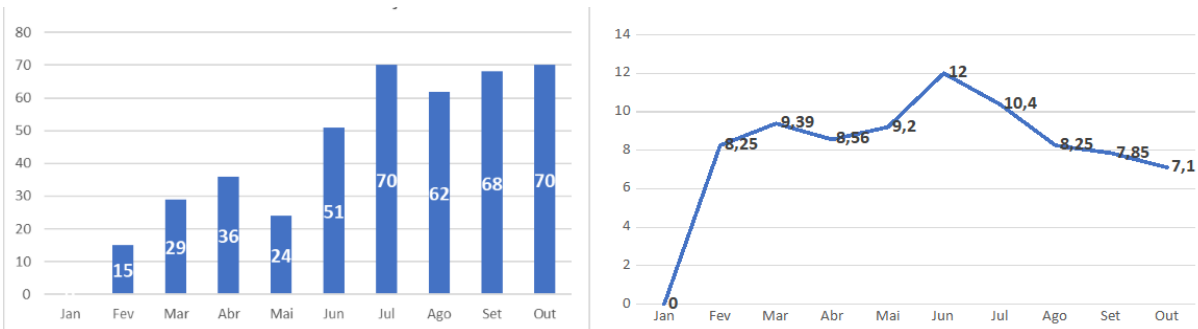


Figura 7 - Quantidade de mudanças e tempo médio entre mudanças.

Fonte: Adaptado pelos autores de Banco MDN (2020).

O indicador de tempo das entregas (*Lead Time*) foi acompanhado pelas histórias terem todo o ciclo de desenvolvimento até a implantação da funcionalidade, onde até junho de 2020 o *lead time* médio era de 41 dias por história. Com os resultados já aferidos foi possível constatar que o *lead time* média atual está em torno de 29 dias, conforme demonstra a figura 8.

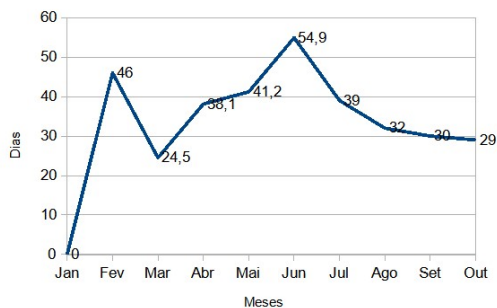


Figura 8 - Tempo médio *lead time* das histórias.

Fonte: Adaptado pelos autores de Banco MDN (2020).

O resultado do tempo de esperado para entrega de novas funcionalidades, visto pelo indicador de tempo médio entre mudanças, mostra um número onde há evidente evolução do quadrante proposto por Forsgren (2018), onde ele relata que um alto desempenho deve ser tempo médio entre mudanças menor que uma semana até várias vezes ao dia e o resultado desse indicador ficou em 7 dias para o aplicativo móvel. Evidentemente, trabalhos complementares a esse devem ser realizados porque a evolução tecnológica é um processo constante, inclusive a evolução da plataforma do *internet banking*.

5. CONTRIBUIÇÃO TECNOLÓGICA-SOCIAL

A busca por velocidade para entrega de produtos que satisfaçam as necessidades dos clientes em um cenário de maior competição foi fator determinante para a criação de uma nova solução de arquitetura de software e proposta de modelo de trabalho.

Por ser uma proposta inovadora na empresa, no início sofreu uma forte resistência, mas após o período dos estudos realizados nos dois primeiros meses do projeto, apesar de sua complexidade, conseguiu-se, então, demonstrar o potencial que esse método poderia alcançar. Anteriormente os dados, que eram produzidos pela empresa e disponibilizados não possuíam uma forma de cobrança pré-estabelecida, isto é, eram fornecidos de forma gratuita pelo site institucional da empresa.

A nova arquitetura também exigiu um novo modelo de trabalho que considerou um diagnóstico de mercado, bem como escolhas referentes às estratégias competitivas, as condições idealizadas para a criação de valor e construção de vantagens competitivas para o desenvolvimento desta nova arquitetura e desse modelo de trabalho.

Nesse sentido, o problema encontrado pelo MDN, o aumento da concorrência e a possibilidade de utilização da tecnologia como alavanca na disponibilização de produtos e serviços financeiros, culminou na criação desse artigo aplicado.

O grande desafio deste trabalho foi verificar quais práticas de gestão e tecnologias podem ser utilizadas em projetos de modernização de plataforma tecnológica e quais vantagens e desvantagens elas trazem.

Um resultado, não esperado previamente, desse projeto foi a criação e a disseminação da cultura de evoluir a plataforma de forma gradual e constante, possibilitando a disponibilização do uso dos serviços financeiros pelos clientes de forma incremental, como foi o caso do lançamento do PIX.

Isto demonstra a importância da implementação deste projeto relatado, que foi o primeiro de muitos projetos baseados em arquitetura mais modernas de software e utilização de modelos ágeis de trabalho.

REFERÊNCIAS E FONTES DE CONSULTAS

- Federação Brasileira dos Bancos (2018). *Inovação e Competição: novos caminhos para redução dos spreads bancários?* Recuperado em 15/04/2020 de <https://cmsportal.febraban.org.br/Arquivos/documentos/PDF/Inova%C3%A7%C3%A3o%20e%20Competi%C3%A7%C3%A3o%20-%20Novos%20Caminhos%20para%20Redu%C3%A7%C3%A3o%20dos%20Spreads%20Banc%C3%A1rios.PDF>.
- Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: building and scaling high performing technology organizations*. Portland: IT Revolution Press.
- Fowler, M. (2015) *Microservices, a definition of this new architectural term*. junho 2015. Recuperado em 29/06/2020.
- Fowler, M. (2019) *Micro Frontends*. junho 2019. Recuperado em 29/06/2020 de <https://martinfowler.com/articles/micro-frontends.html>
- French, W. L., & Bell, C. (1995). *Organization development: Behavioral science interventions for organization improvement*. Englewood Cliffs: Prentice Hall.
- MDN (2020). *Relações com investidores*. Recuperado em 29/03/20 de <https://www.itau.com.br/relacoes-com-investidores/Download.aspx?Arquivo=1GN/sgNJ8k0SadTcb+bNeA==&linguagem=pt>
- Nogueira, C. S. (2014) *Planejamento Estratégico*. São Paulo: Pearson.
- Nolan, R. L., & McFarlan, F. W. (2005) Information technology and the board of directors. *Harvard Business Review*, 83(10), pp. 96-106.
- PMI. Guia PMBOK (2017): *Um Guia para o Conjunto de Conhecimentos em Gerenciamento de Projetos*, 6th ed, Pennsylvania: PMI.
- Porter, M. E. (1991) *Vantagem competitiva: criando e sustentando um desempenho superior*. Rio de Janeiro: Campus
- Ravichandran, A., Taylor, K., & Waterhouse, P. (2016). *DevOps for digital leaders: reignite business with a modern DevOps-enabled software factory*. Berkeley: Apress CA Press, Distributed to the Book trade worldwide by Springer.
- Sebrae (2019). *Fintech Mining Report* Recuperado em 07/09/2020 de https://respostas.sebrae.com.br/wp-content/uploads/2020/03/Fintech_Mining_Report_compressed.pdf.
- ThoughtWorks (2020). *TECHNOLOGY RADAR* Recuperado em 01/10/2020 de <https://www.thoughtworks.com/radar/techniques/micro-frontends>