



**UNIVERSIDADE DO SUL DE SANTA CATARINA
FEDERICO GUILLERMO MAIER RUTENBERG**

**DESENVOLVIMENTO DE SOFTWARE EM ESCALA UTILIZANDO
PRÁTICAS ÁGEIS:
UM ESTUDO DE CASO EM UMA INSTITUIÇÃO FINANCEIRA**

Palhoça
2017

FEDERICO GUILLERMO MAIER RUTENBERG

**DESENVOLVIMENTO DE SOFTWARE EM ESCALA UTILIZANDO
PRÁTICAS ÁGEIS:
UM ESTUDO DE CASO EM UMA INSTITUIÇÃO FINANCEIRA**

Relatório apresentado ao Curso **Tecnólogo em Gestão da Tecnologia da Informação**, da Universidade do Sul de Santa Catarina, como requisito parcial à aprovação na unidade de aprendizagem de Estudo de Caso.

Orientador: Prof. Horácio Dutra Mello

Palhoça

2017

FEDERICO GUILLERMO MAIER RUTENBERG

**DESENVOLVIMENTO DE SOFTWARE EM ESCALA UTILIZANDO
PRÁTICAS ÁGEIS:
UM ESTUDO DE CASO EM UMA INSTITUIÇÃO FINANCEIRA**

Este trabalho de pesquisa na modalidade de Estudo de Caso foi julgado adequado à obtenção do grau de Tecnólogo em Gestão da Tecnologia da Informação e aprovado, em sua forma final, pelo Curso Superior de Tecnologia em Gestão da Tecnologia da Informação, da Universidade do Sul de Santa Catarina.

Palhoça, 6 de dezembro de 2017.

Prof. e orientador (Horário Dutra Mello),
Universidade do Sul de Santa Catarina

RESUMO

Com a demanda latente por velocidade na resolução de problemas e aumento da complexidade dos produtos de software desenvolvido, as empresas se veem na necessidade de escalar o desenvolvimento de software. Isto é, múltiplos times, trabalhando de forma integrada em um mesmo produto. O efeito da abordagem de desenvolvimento em escala traz outras preocupações e restrições que o desenvolvimento tradicional não precisava preocupar-se. Dentre eles, cabe citar o aumento dos esforços de integração, o gerenciamento de dependências e a preocupação na gestão de débitos técnicos. O presente trabalho, através de um estudo de caso exploratório, tem por finalidade apresentar um cenário de desenvolvimento em escala em um ambiente bastante complexo, o desenvolvimento de um produto digital mobile para uma grande instituição financeira brasileira, cujo foco são os clientes pessoa jurídica dessa instituição. Como resultado desse estudo de caso, temos o aprendizado validado de boas práticas de desenvolvimento em escala, que visaram resolver os problemas encontrados no decorrer da execução do projeto.

Palavras-chave: Métodos Ágeis. Desenvolvimento de Software. Desenvolvimento em Escala. Agilidade.

SUMÁRIO

1 INTRODUÇÃO	6
2 DESENVOLVIMENTO DE SOFTWARE EM ESCALA UTILIZANDO PRÁTICAS ÁGEIS	7
3 OBJETIVOS	11
3.1 OBJETIVO GERAL	11
3.2 OBJETIVOS ESPECÍFICOS	11
4 PROCEDIMENTOS METODOLÓGICOS	12
4.1 CAMPO DE ESTUDO	12
4.2 INSTRUMENTOS DE COLETA DE DADOS	12
5 APRESENTAÇÃO E ANÁLISE DA REALIDADE OBSERVADA	13
5.1 CONTEXTO DAS ORGANIZAÇÕES ENVOLVIDAS	13
5.2 CONSULTORIA ESPECIALIZADA EM DESENVOLVIMENTO DE PRODUTOS DIGTAIS	13
5.3 DESCRIÇÃO DO PRODUTO DIGITAL DESENVOLVIDO	14
5.4 DESCRIÇÃO E ANÁLISE DA REALIDADE OBSERVADA	15
6 PROPOSTA DE SOLUÇÃO DA SITUAÇÃO-PROBLEMA	17
6.1 PROPOSTA DE MELHORIA PARA A REALIDADE ESTUDADA	17
6.2 RESULTADOS ESPERADOS	18
6.3 VIABILIDADE DA PROPOSTA	19
7 CONSIDERAÇÕES FINAIS	20
REFERÊNCIAS	21

1 INTRODUÇÃO

O presente trabalho trata-se de um estudo de caso contendo propostas de organização, de pessoas e processo, a respeito do desenvolvimento ágil de software. O objetivo é apresentar propostas para a resolução de problemas principais que impactam o desenvolvimento em escala, como organização de equipes, redução de dependências, priorização de funcionalidades, entre outros.

O desenvolvimento em escala vem sendo bastante discutido na comunidade ágil de uma forma geral, tanto pela academia como pelas empresas e organizações de um modo geral. Alguns frameworks já foram propostos para resolver o problema, mas a aplicação desses ainda carece de alguns elementos que o cenário do estudo de caso não comportaria. Sendo assim, o aprendizado validado, através das hipóteses formuladas nesse estudo de caso, terão um valor inestimável para o desenvolvimento do tema e das pessoas que desenvolvem software em escala, um cenário cada vez mais comum nas organizações. As hipóteses serão formuladas através de uma pesquisa exploratória, onde o autor é parte importante no processo de organização do desenvolvimento do projeto e tem autonomia para propor soluções que venham a resolver os problemas identificados.

Para realização do trabalho, será apresentada uma contextualização teórica sobre desenvolvimento ágil de software, seguida da exploração do ambiente pelo autor, através da observação e participação direta no processo de organização, levantamento dos principais problemas identificados, propondo-se soluções de melhoria para os mesmos, relacionando seus resultados esperados e fazendo uma análise da viabilidade de implementá-los. Com isso, espera-se como resultado final desse trabalho um conjunto de soluções analisadas, conforme o contexto, para resolver os problemas encontrados durante a análise do cenário.

2 DESENVOLVIMENTO DE SOFTWARE EM ESCALA UTILIZANDO PRÁTICAS ÁGEIS

A indústria de software é uma das que mais movimenta a economia em todo o mundo. De acordo com o relatório da Gartner (2016), estima-se que os gastos com software em todo do mundo chegarão a US\$ 357 bilhões, representando um crescimento de 7,2% em relação a 2016. Muito desse investimento em software justifica-se por atualmente é praticamente impossível imaginar a vida sem software. Com a explosão da internet no início dos anos 2000 e o ‘boom’ das tecnologias móveis, software está por toda a parte, nas mais diversas atividades, desde programas embarcados em aparelhos de TVs e automóveis até softwares de complexidade mais elevadas, capazes de examinar e diagnosticar pacientes com determinados tipos de doenças. Conforme afirmam Schwaber e Sutherland (2012, p. 1), “software não está mais somente no quarto dos fundos. Software está em todo lugar, em operações cada vez mais críticas em nossa sociedade”. Esses mesmos autores afirmam ainda que sem software a possibilidade de sobrevivência nos negócios torna-se bastante limitada, quase impossível.

O desenvolvimento de software é uma atividade de extrema complexidade. De la Riva (2016, p. 9) afirma que “desenvolvimento de software é uma área complexa, tratada por seres humanos, e os erros são frequentes e devem continuar sendo”. A tecnologia está em constante evolução, o cenário de negócios está cada vez mais dinâmico, com isso a necessidade de resposta rápida às mudanças é vital para a perenidade dos negócios em si. Com isso, diversas tentativas de acelerar essa resposta às mudanças em termos de métodos ou boas práticas de desenvolvimento de software.

No início da indústria de software, nos anos 70, as práticas definidas foram inspiradas no taylorismo e no fordismo. Ou seja, o paradigma industrial foi migrado sem adaptações para a indústria de software. Conforme afirma Verheyen (2013), durante muito tempo a indústria de software foi dominada pelos paradigmas da visão e crenças industriais. Um paradigma baseado no trabalho repetitivo, no comando e controle, em planejamento a longo prazo e na documentação excessiva que não permitia aos desenvolvedores a possibilidade de inovar e fazer as coisas de um modo diferente. Aos desenvolvedores em tempos remotos, só era dada a oportunidade de executar as tarefas. Porém, desenvolvimento de software é uma atividade legitimamente provida do conhecimento. A necessidade de se adaptar às mudanças é inerente a essa indústria. Sendo assim, era impossível obter resultados mais condizentes ao investimento cada vez maior se a abordagem permanecia a mesma.

Dado esse cenário a indústria de software passava por uma crise profunda, devido a taxas de sucesso cada vez menores. Segundo o Relatório do Caos (Chaos Report) do Standish Group, apenas 10 a 20 por cento dos projetos de software eram entregues com sucesso. Um número mais alarmante, também revelado por esse mesmo relatório revelava que apenas 20% do software desenvolvido era de fato utilizado pelos clientes. Foi então que na década de 90, novas maneiras de desenvolver software começaram a ser testadas e disseminadas. As lideranças da indústria de software ao redor do mundo dispuseram-se a romper os paradigmas e modelos antigos e utilizar novos meios de entregar software. Era o surgimento do que chamamos hoje de Métodos Ágeis.

O grande marco desse movimento foi a publicação do Manifesto para o Desenvolvimento Ágil de Software (Beck, Beedle, et.al., 2001) que declarava:

“Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

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

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.”

Sendo assim, os métodos e práticas de desenvolvimento de software que se basearam no Manifesto Ágil, passaram a ser conhecidos no mercado como Métodos Ágeis. Verheyen (2013) afirma que as principais características comuns aos métodos ágeis são:

a) Orientação a pessoas: a contínua colaboração das pessoas potencializará a criatividade necessária para o desenvolvimento de software e estimulará a inteligência coletiva e a auto-organização;

b) Facilitação: a liderança-servidora emerge para facilitar as atividades do time de desenvolvimento ao contrário da abordagem comando-e-controle taylorista;

c) Processo iterativo e incremental: os produtos são criados “pedaço por pedaço”, integrados e a integração toda é revisitada constantemente para garantir a integridade;

d) Medida de sucesso: o foco da definição de sucesso dos métodos ágeis está em entregar valor continuamente para os clientes e usuários.

e) Mudança: a mudança constante através da inspeção e adaptação contínuas é uma fonte de inovação e melhoria.

Dentre os Métodos Ágeis mais experimentados na ascensão do movimento, podemos destacar a Programação Extrema (Extreme Programming), Crystal, o mais popular de todos os métodos, o Scrum. Segundo o relatório da Scrum Alliance (2016), cerca de 80% dos projetos de software desenvolvidos atualmente utilizam o Scrum.

O Guia do Scrum (2016) afirma que Scrum é um framework simples e adaptativo para desenvolvimento de produtos complexos. A popularidade do Scrum deu-se sobretudo pela simplicidade de ser aplicado. Scrum não é um acrônimo, refere-se a uma jogada do rúgbi de mesmo nome, onde todos os jogadores se reúnem e juntos tentam conquistar a posse da bola. O Scrum baseia-se no empirismo, no conceito de time-box (todas as atividades têm um limite máximo de tempo) e no desenvolvimento iterativo e incremental. Para isso o Scrum define eventos, papéis e artefatos, explicados pelo Guia do Scrum (2016) e relacionados abaixo:

Papéis: Dono do Produto, Scrum Master e Time de Desenvolvimento.

Artefatos: Backlog do Produto, Backlog do Sprint e Incremento de Software.

Eventos: Planejamento do Sprint, Reunião Diária, Sprint, Revisão do Sprint e Retrospectiva do Sprint.

As premissas do Scrum é aproveitar o máximo dos papéis estabelecidos para que através dos artefatos e eventos o time entregue um incremento de software a cada iteração, chamada no Scrum de Sprint. Sprint é um período de 1 a 4 semanas, onde o time trabalha de forma coesa e com um objetivo em comum, que é desenvolver um incremento de software utilizável nesse período de tempo. Um time Scrum tem a premissa de ser pequeno, tendo de 3 a 9 pessoas. Times ágeis trazem essa característica de serem pequenos para facilitar a comunicação.

Porém, conforme supracitado, a necessidade de se adaptar às mudanças cada vez numa maior velocidade é um requisito essencial para o desenvolvimento de software moderno. Os produtos finais da entrega estão cada vez maiores (em termos de base de código) e complexos. Sendo assim, nem sempre um time de desenvolvimento é suficiente para atender essa velocidade e os requisitos de complexidade. O desenvolvimento de software envolvendo múltiplos times sob uma mesma base de código é conhecido como desenvolvimento em escala. Sendo assim, no desenvolvimento em escala, qual a maneira mais eficaz de organizar vários times de desenvolvimento e sincronizar seu trabalho em curtos períodos de tempo? Quais as melhores práticas de desenvolvimento a serem aplicadas em múltiplos times? Como times acostumados a trabalhar de forma independente adaptam-se a um trabalho em escala?

O presente estudo pretende explorar, através de experiência prática do autor, a vivência do contexto de agilidade em escala de forma prática, o que também permitiu a construção em ciclos contínuos de inspeção e adaptação. O interesse pelo tema surgiu da necessidade de aplicação prática de conceitos de desenvolvimento ágil de software em escala, sem a pretensão de forjar aos times de desenvolvimento frameworks ou métodos de trabalho rígidos pré-estabelecidos. A decisão de elencar as boas práticas utilizadas no processo de formação de times para desenvolvimento em escala em uma empresa conhecida pelo tradicionalismo, deu-se em virtude de uma necessidade latente do mercado em disseminar bons cases a respeito do tema.

3 OBJETIVOS

3.1 OBJETIVO GERAL

Propor boas práticas, através de aprendizado validado, de desenvolvimento de software em escala utilizando Métodos Ágeis.

3.2 OBJETIVOS ESPECÍFICOS

- Caracterizar o desenvolvimento de software em escala.
- Descrever os principais problemas vivenciados por times que desenvolvem software em escala
- Elaborar propostas para a resolução dos problemas, dado o cenário encontrado.

4 PROCEDIMENTOS METODOLÓGICOS

4.1 CAMPO DE ESTUDO

O presente estudo de caso tem caráter exploratório, conforme os objetivos definidos explicitam. O objeto desse estudo é a prestação de serviços de uma consultoria de desenvolvimento de software para uma instituição financeira de grande porte da cidade de São Paulo, SP. O público-alvo da pesquisa é um grupo de aproximadamente 40 pessoas, incluindo funcionários de ambas as partes, que trabalham de forma integrada para o desenvolvimento de um produto único, no caso um aplicativo que faz isso de tecnologias móveis cuja função principal é prover serviços de *mobile banking* para os clientes de pessoa jurídica da instituição.

4.2 INSTRUMENTOS DE COLETA DE DADOS

O quadro 1 abaixo descreve os instrumentos de coleta de dados que serão utilizados no presente estudo de caso.

Quadro 1 – Instrumento de coleta de dados

Instrumento de coleta de dados	Universo pesquisado	Finalidade do Instrumento
Observação direta ou dos participantes	O autor do presente trabalho é parte integrante da equipe de desenvolvimento do produto que é o objeto de estudo do presente trabalho. Sendo assim, tem condições de observar diretamente as relações humanas estabelecidas entre os envolvidos no projeto, tal como as práticas adotadas e os resultados obtidos.	Validar o aprendizado das práticas experimentadas.
Documentos	Serão analisados os relatórios de status, apresentados aos patrocinadores do projeto, tal como os registros internos do time referentes à lições aprendidas.	Acompanhar a evolução do histórico da aprendizagem validada e registrar as lições aprendidas para uso futuro.

Fonte: O próprio autor (2017).

5 APRESENTAÇÃO E ANÁLISE DA REALIDADE OBSERVADA

O presente estudo de caso, refere-se ao desenvolvimento de um aplicativo utilizando para dispositivos móveis, cuja finalidade é fornecer serviços de *banking* para pessoas jurídicas clientes da instituição financeira. Envolve 02 agentes principais: a consultoria que fornece o time de desenvolvimento e a instituição financeira, à qual pertence os Donos de Produto, os responsáveis por fornecer os requisitos de negócio do que precisam ser desenvolvidos, além da infraestrutura para a validação do desenvolvimento.

5.1 CONTEXTO DAS ORGANIZAÇÕES ENVOLVIDAS

Dentro desse contexto, será descrito o contexto das duas organizações envolvidas.

– Consultoria de Desenvolvimento de Produtos Digitais: é uma empresa de médio porte, com a matriz localizada no estado do Rio de Janeiro e com filiais em São Paulo e Minas Gerais. Possui aproximadamente 400 funcionários e é especializada no desenvolvimento de produtos digitais *mobile*, isto é, os desenvolvidos para serem utilizados por dispositivos móveis (*smartphones, tablets, gadgets*, entre outros). A empresa também é reconhecida pelo uso efetivo de Metodologias Ágeis para o Desenvolvimento de Software, dentre os quais se destaca: uso do *framework* Scrum, Lean UX e cultura DevOps. A empresa possui o histórico de desenvolvimento de mais de 200 aplicativos desenvolvidos desde a concepção até a implantação em produção.

– Instituição Financeira: por sua vez, a instituição financeira é a maior instituição privada do país. Possui mais de 10.000 funcionários, com um departamento de Tecnologia da Informação ultrapassando 4.000 funcionários. Em seu portfólio, atende os mais diversos segmentos de clientes em termos de serviços financeiros. Por seus movimentos recentes, podemos classificar que essa instituição está em franco processo de transformação digital e esvaziamento de agências em troca de fornecer os serviços de forma digital.

5.2 CONSULTORIA ESPECIALIZADA EM DESENVOLVIMENTO DE PRODUTOS DIGITAIS

A opção de contratar uma consultoria especializada para o desenvolvimento de produtos digitais ao invés de internalizar o desenvolvimento com recursos próprios, é comum no segmento. Dentre os principais motivos, podemos destacar:

- Concentrar-se nas atividades *core* do negócio: a atividade core de uma instituição financeira é fornecer serviços financeiros, dos mais diversificados aos seus públicos. Não é desenvolver aplicativos. Sendo assim, a terceirização do desenvolvimento permite que a empresa contratante foque em oferecer os melhores requisitos de negócio para atender as necessidades de seus clientes, enquanto a consultoria especializada encarrega-se de atender os requisitos técnicos.

- Ganhar velocidade e qualidade nas entregas: contratando uma consultoria especializada nas tecnologias que a empresa necessita, diminui-se o tempo de aprendizagem e domínio da tecnologia pois a atividade core da consultoria é desenvolver aplicativos. Além disso, contratando especialistas de um parceiro reconhecido pelo mercado, a empresa certifica-se de usufruir o que há de mais atual em termos das tecnologias de desenvolvimento. Além disso, os especialistas apresentam uma maior produtividade nas entregas, além de uma efetividade maior em termos de qualidade.

- Reduzir o Custo Total da Operação: após o desenvolvimento do aplicativo, a empresa contratante não é obrigada a manter a equipe completa pois o serviço já foi entregue. Deve certificar-se que a passagem de conhecimento e a manutenção/suporte do aplicativo ocorra de forma adequada. Ao contrário de uma equipe própria, a empresa não ficará com excesso de pessoas após a entrega do aplicativo, reduzindo assim, o Custo Total da Operação.

Pelo lado da consultoria, ter um cliente desse porte é de extrema importância e muito interessante para colocar no seu portfólio de produtos entregues. Além de ser uma referência no mercado, a complexidade e a escala com que o aplicativo é utilizado tornam ainda mais atrativo o desenvolvimento e o fortalecimento da relação de parceria.

5.3 DESCRIÇÃO DO PRODUTO DESENVOLVIDO

O produto digital desenvolvido é um aplicativo do segmento de *Mobile Banking* focado no atendimento de pessoas jurídicas, isto é, empresas. O aplicativo atende as plataformas Android (Google) e iOS (Apple). Em relação ao público-alvo do aplicativo, trata-se do mais diversificado possível atende desde o microempreendedor até empresas de grande porte. O aplicativo possui diversas funcionalidades comuns características do segmento, dentre as quais, cabe destacar: consulta de saldo e extrato, pagamentos e transferências. O aplicativo possui mais de 500 mil usuários ativos, o que confirma a necessidade de respostas rápidas na utilização.

5.4 DESCRIÇÃO E ANÁLISE DA REALIDADE OBSERVADA

A consultoria de desenvolvimento fornece times multifuncionais, isto é, com todos os papéis necessários para colocar o produto em produção. Nesse projeto em específico, somente alguns papéis característicos da empresa contratante não foram atendidos, principalmente a respeito da infraestrutura e algumas tecnologias de *backend* e banco de dados características do setor bancário. Além dos desenvolvedores, a consultoria fornece um especialista em Métodos Ágeis, cuja principal responsabilidade é garantir que o processo está ocorrendo da forma mais ágil possível. Uma célula média é composta por 09 pessoas:

- 02 desenvolvedores Android
- 02 desenvolvedores iOS
- 01 analista de qualidade
- 01 desenvolvedor de API
- 01 especialista em Experiência do Usuário
- 01 especialista em Design de Interface
- 01 especialista em métodos ágeis

O modelo de contratação é através de horas bilhetáveis da consultoria a um valor fixo para os profissionais, independentemente do nível de senioridade ou papel exercido no time. Foram contratadas 04 células com a composição acima, formando um time de 36 pessoas somente pelo lado da consultoria.

Além disso, conforme supracitado, completam uma célula de desenvolvimento os Donos do Produto, os profissionais da empresa contratante responsáveis por definir os requisitos de negócio que precisam ser desenvolvidos. Os mesmos também têm como incumbência validar se o desenvolvimento realizado atende os requisitos estabelecidos.

Sendo assim, temos um time formado por cerca de 40 profissionais para desenvolvimento de um conjunto de funcionalidades para o aplicativo de pessoas jurídicas. Os Donos do Produto haviam definido com suas áreas de negócio uma série de funcionalidades que precisam ser desenvolvidas em um ano. Porém, essa estimativa de início pareceu fora da realidade. Os times não foram envolvidos nas estimativas e, apesar de serem especialistas em desenvolvimento das tecnologias do aplicativo, os times ainda não atuaram juntos, principalmente em um modelo de desenvolvimento em escala.

Um aspecto relevante no desenvolvimento e na escolha das funcionalidades a serem desenvolvidas é que não havia uma estratégia clara de produtização definida para a priorização do que deveria ser desenvolvido e quando. A priorização em busca da maximização do retorno do investimento é uma das premissas de Métodos Ágeis, pois aumenta a percepção de valor entregue. Os Donos do Produto também tinham visões diferentes entre si a respeito da priorização.

A organização da forma de trabalho integrada do time ainda não estava clara. Tinha-se uma ideia de organização inicial baseada em componentização (time de Android, time de iOS, time de testes, etc), o que geraria mais dependência entre os times de desenvolvimento. Dependências estas também que não estavam por completo mapeadas, uma vez que nenhum time havia desenvolvido uma funcionalidade de ponta a ponta do produto, utilizando a nova estrutura proposta.

Por último, apesar de ser uma consultoria especializada, a mesma não possui um “pool” de desenvolvedores disponíveis para ser integrados aos clientes sempre que precisarem. Os desenvolvedores são contratados à medida que são desalocados de outros clientes ou que são contratados. O prazo médio de contratação de desenvolvedores é de cerca de 60 dias, com uma margem de 15 dias para mais ou para menos. A carência de profissionais qualificados no mercado de trabalho impacta diretamente nesse prazo longo. Com isso, a formação dos times está comprometida, com impacto direto nos prazos de entrega e ambientação dos desenvolvedores.

6 PROPOSTA DE SOLUÇÃO DA SITUAÇÃO-PROBLEMA

Dada a situação identificada, podemos enumerar os principais problemas a serem resolvidos no contexto do desenvolvimento de software em escala em:

- 1) Falta de objetivos claros de negócio a serem atingidos pelo desenvolvimento;
- 2) Time não participou das estimativas. Com isso, o escopo requerido pelo cliente era muito grande para o prazo estabelecido;
- 3) Não havia clareza de como os times seriam organizados / distribuídos;
- 4) A falta de clareza na distribuição dos times acarretava na indefinição dos processos de trabalho ou se os times seguiriam algum framework específico para gestão ágil.
- 5) Elevado número de dependência em todo o processo de desenvolvimento, principalmente pela burocracia peculiar da instituição financeira;
- 6) Impacto elevado na demora para chegada de novos desenvolvedores.

6.1 PROPOSTA DE MELHORIA PARA A REALIDADE ESTUDADA

Bem, para cada problema identificado inicialmente, foi proposta uma solução, conforme descrito no quadro 2.

Quadro 2 – Proposta de Melhoria para os Problemas Encontrados

Problema	Proposta de Melhoria
Falta de objetivos claros de negócio a serem atingidos pelo desenvolvimento	<ul style="list-style-type: none"> - Realizar discussão com Donos do Produto e Áreas de Negócio a respeito das funcionalidades solicitadas para desenvolvimento - Analisar as funcionalidades baseando-se em dados de mercado e uso dos serviços pelos clientes - Definir uma priorização orientada à prioridades de negócio e análise de mercado
Trabalho esperado com estimativa abaixo da realidade	<ul style="list-style-type: none"> - De acordo com as prioridades definidas pela área de negócios, definir entregas de ciclos curtos para que o time possa medir o esforço de desenvolvimento e prover estimativas mais assertivas
Falta de clareza na organização / distribuição dos times	<ul style="list-style-type: none"> - Organizar os times por grupos de funcionalidades

Indefinição nos processos de trabalho dos times	- Definida a organização do time, organizar o processo nas etapas de desenvolvimento seguindo os princípios de agilidade: entregas curtas e incrementais e ciclo adaptativo.
Elevado número de dependências no processo de desenvolvimento	- Mapear as dependências de toda a cadeia de valor do processo de desenvolvimento - Avaliar quais os papéis que deveriam ser agregados ao time e solicitar integração de pessoas com esses papéis.
Alto impacto na demora para chegada de novos desenvolvedores	- Trabalhar junto ao Recursos Humanos da empresa consultora para priorizar os desenvolvedores para o projeto - Alocar os desenvolvedores nas features prioritárias

Fonte: o próprio autor (2017)

6.2 RESULTADOS ESPERADOS

Com as propostas de melhorias apresentadas na seção anterior, espera-se atingir os seguintes resultados:

- Desenvolvimento orientado a valor de negócios: as funcionalidades que gerem mais valor para o negócio serão desenvolvidas primeiro pelo time de desenvolvimento;
- Estimativas mais assertivas: com as dependências reduzidas e o time conhecendo o processo de desenvolvimento de ponta-a-ponta, os times serão capazes de ser mais assertivos nas estimativas;
- Organização de células de desenvolvimento autônomas, capazes de desenvolver incrementos de software de ponta-a-ponta;
- Estruturação de um processo ágil, com planejamento, melhoria contínua e incentivo à inspeção e adaptação constantes, tanto do time dos processos do time de desenvolvimento como do produto.
- Redução no tempo de integração e aprendizagem de novos integrados do time de desenvolvimento.

Consequentemente, o conjunto de melhorias apresentadas, integradas, trarão uma redução nos tempos de entrega, aceleração de aprendizagem coletiva e entregas contínuas de maior valor agregado.

6.3 VIABILIDADE DA PROPOSTA

As propostas de melhoria apresentadas são consideradas viáveis e de média complexidade de implementação. Basicamente, trata-se de uma reorganização de pessoas e processo à luz dos princípios ágeis de desenvolvimento. A única proposta que poderia enfrentar uma certa resistência ou dificuldade de implantação seria ter todos os papéis disponíveis dentro do time de desenvolvimento pois a cadeia de valor do processo de desenvolvimento de software na instituição financeira ainda não está total e claramente mapeada. Com isso, pode-se descobrir etapas do processo não previstas inicialmente e, por conseguinte, as pessoas com as habilidades necessárias não estarem disponíveis para integração ao time de desenvolvimento.

7 CONSIDERAÇÕES FINAIS

O desenvolvimento de software é, por natureza, uma atividade complexa que demanda muitos esforços de coordenação e alinhamento. Há muitas atividades e artefatos a serem considerados quando da criação de um incremento de software “pronto” para uso pelos clientes ou usuários finais. Desde a década de 90, os métodos ágeis vem desenvolvendo-se, reinventando-se, para encontrar novas maneiras de desenvolver software e ensinar outros a fazê-lo, conforme propõe o próprio Manifesto para o Desenvolvimento Ágil de software.

A complexidade do desenvolvimento de software se potencializa consideravelmente quando o desenvolvimento dá-se em escala. Por escala, entende-se quando três ou mais times de desenvolvimento estão trabalhando sob a mesma base de código para entregar funcionalidades de um mesmo produto. Esse foi o cenário encontrado no presente estudo de caso, onde um grupo de 40 desenvolvedores trabalhavam sob a mesma base de código desenvolvendo um produto de *mobile banking*. Ainda há muita discussão na comunidade e em empresas que utilizam métodos ágeis sobre como manter um ambiente ágil no cenário do desenvolvimento em escala. Isso configurou-se em alguns frameworks estabelecidos por algumas instituições, como *Nexus*, *LeSS*, *SAFe* estão entre os mais conhecidos.

No estudo de caso, optou-se por não utilizar um framework pré-estabelecido, pois a equipe ainda não possuía o nível de maturidade e os elementos necessários para adotar algum dos frameworks. Era uma equipe em formação que ainda carecia de alguns elementos, em termos de pessoas, experiência e até mesmo estrutura para adotar algum modelo de trabalho pré-estabelecido. O time decidiu definir suas próprias práticas, tomando como ponto de partida o manifesto ágil de desenvolvimento de software. As propostas apresentadas, de certa forma, estão ainda em caráter de experimentação, pois nenhum dos membros do time, apesar de experiência em times ágeis, havia trabalhado em um modelo de ágil em escala. Basicamente, as propostas apresentadas buscam perseguir alguns objetivos em comum do desenvolvimento ágil: priorização orientada a valor, entregas contínuas e incrementais, times multifuncionais e auto-organizáveis e ciclos curtos de inspeção e adaptação.

Espera-se, com as propostas estabelecidas, validar boas práticas de desenvolvimento de software e compartilhar conhecimento com outras organizações que estejam em busca de referências para resolver os problemas vivenciados nesse estudo de caso.

REFERÊNCIAS

Agile Manifesto website. Disponível em: <[http://www. http://agilemanifesto.org/](http://www.agilemanifesto.org/)>. Acesso em: 15 ago. 2017.

CAVALCANTI, Marcelo e MOREIRA, Enzo. **Metodologia de estudo de caso:** livro didático. 3. ed. rev. e atual. Palhoça: Unisul Virtual, 2008. 170 p.

COHN, Mike. **Succeeding with Agile: software development using Scrum.** Boston: Addison-Wesley, 2010.

DE LA RIVA, Fernando. **Silicon Valley is Here!: como usar a agilidade corporativa para competir.** São Paulo: Concrete Solutions, 2016.

GARTNER. **Gartner Says Global IT Spending to Reach \$3.5 Trillion in 2017.** Disponível em: <<http://www.gartner.com/newsroom/id/3482917>>. Acesso em: 01 ago. 2017.

LOHN, Joel Irineu. **Metodologia para elaboração e aplicação de projetos:** livro didático. 2 ed. rev. e atual. Palhoça: Unisul Virtual, 2005. 100 p.

SCHWABER, Ken; SUTHERLAND, Jeff. **Guia do Scrum – Um guia definitivo para o Scrum: As regras do jogo.** Scrum.org, 2016.

SCRUM ALLIANCE. **The State of Scrum Report: 2017 edition.** Disponível em: <<https://www.scrumalliance.org/why-scrum/state-of-scrum-report/2017-state-of-scrum>>. Acesso em: 10 ago. 2017.

STANDISH Group. **The Standish Group Report CHAOS.** Disponível em: <<https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>>. Acesso em: 13 ago. 2017.

VERHEYEN, Gunther. **Scrum: a pocket guide.** Van Haren Publishing, 2013.