

Universidade Federal de Pernambuco
Centro de Informática
Curso de Ciência da Computação

Daniel Sulman de Albuquerque Eloi

**UM ESTUDO COMPARATIVO DE METODOLOGIAS ÁGEIS NO
DESENVOLVIMENTO DE APLICATIVOS MÓVEIS**

Trabalho de conclusão de curso

Recife

2016

Daniel Sulman de Albuquerque Eloi

**UM ESTUDO COMPARATIVO DE METODOLOGIAS ÁGEIS NO
DESENVOLVIMENTO DE APLICATIVOS MÓVEIS**

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Dra. Carina Frota Alves

Recife

2016

DEDICATÓRIA

A toda minha família.

AGRADECIMENTOS

Primeiramente aos meus pais, em especial a minha mãe que sempre esteve ao meu lado me apoiando em todos os momentos, independente se bom ou ruim, sem qual eu jamais chegaria a essa etapa.

À Marina, por sempre estar ao meu lado me dando suporte e passando comigo por todas as fases da minha longa e tortuosa caminhada até este momento.

À minha orientadora, Carina Frota, por ter me orientado e compreendido minhas dificuldades ao longo deste trabalho de graduação.

Aos meus colegas e amigos de turma, que estiveram comigo em todos os momentos durante minha graduação em ciência da computação, em especial ao "team guido", que tanto me apoiou nos momentos que eu mais precisei.

A todos, muito obrigado.

RESUMO

O desenvolvimento para dispositivos móveis exigiu diversas mudanças no processo de criação de softwares. As metodologias que mais se adequaram às exigências desse tipo de desenvolvimento são as metodologias ágeis, justamente pela sua capacidade de se adequar a mudanças em qualquer etapa do desenvolvimento, com entregas frequentes além de constante interação com o cliente, o que evita erros de requisitos. Apesar do sucesso alcançado por essas metodologias, a literatura voltada exclusivamente para métodos de criação de software para dispositivos móveis não é vasta, existindo apenas algumas metodologias exclusivas para o desenvolvimento móvel, onde a grande maioria são adaptações da indústria ou do desenvolvimento para desktop.

Palavras-chave: Metodologias ágeis, desenvolvimento móvel.

ABSTRACT

The software development for mobile devices required several changes on its process. The methodologies that best suit the requirements of this type of development are agile methodologies, precisely because of their ability to adapt to changes at any stage of development, with frequent deliveries in addition to constant interaction with the customer, which avoid requirements errors. Despite the success achieved by these methods, the literature focused in the mobile development is not vast, with only a few unique methodologies for mobile development, while the majority are industry or desktop adaptations.

Keywords: Agile methodology, mobile development.

LISTA DE FIGURAS

| | |
|----------------------------------------------------|----|
| Figura 1. Fluxo do Scrum..... | 17 |
| Figura 2. Práticas do Scrum..... | 18 |
| Figura 3. Fluxo do Kanban..... | 22 |
| Figura 4. Ciclo da metodologia cascata..... | 29 |
| Figura 5. Fase de produção..... | 26 |
| Figura 6. Etapas da pesquisa..... | 36 |

LISTA DE GRÁFICOS

| | |
|-------------------------------------------------------------------|----|
| Gráfico 1. Duração dos projetos..... | 39 |
| Gráfico 2. Tempo de utilização de metodologias ágeis..... | 39 |
| Gráfico 3. Metodologia ágil utilizada..... | 40 |
| Gráfico 4. Aderência das equipes aos princípios ágeis..... | 42 |
| Gráfico 5. Comparativo entre literatura e prática..... | 43 |
| Gráfico 6. Comparação no número de atrasos..... | 44 |

Sumário

| | |
|----------------------------------------------------------------|-----------|
| 1. Introdução..... | 9 |
| 1.1. Contexto e motivação..... | 9 |
| 1.2. Objetivos | 10 |
| 1.3. Estrutura do documento..... | 10 |
| 2. Revisão da literatura..... | 12 |
| 2.1. Metodologias ágeis..... | 12 |
| 2.1.1. Histórico..... | 12 |
| 2.1.2. Conceitos..... | 13 |
| 2.1.3. Fatores sociais e humanos..... | 14 |
| 2.1.4. Principais metodologias ágeis..... | 16 |
| 2.1.4.1. Scrum..... | 16 |
| 2.1.4.2. XP..... | 18 |
| 2.1.4.3. Kanban..... | 21 |
| 2.1.4.4. Mobile-D..... | 22 |
| 2.2. Comparação entre metodologias ágeis e tradicionais..... | 26 |
| 2.2.1. Metodologia tradicional..... | 27 |
| 2.2.2. Metodologia ágil..... | 28 |
| 2.2.3. Comparativo entre as metodologias..... | 29 |
| 2.3. Características do desenvolvimento de software móvel..... | 31 |
| 3. Método de pesquisa..... | 34 |
| 3.1. Visão geral..... | 34 |
| 3.2. Objetivos..... | 34 |
| 3.3. Aplicação da metodologia..... | 35 |
| 4. Resultado da pesquisa com desenvolvedores..... | 37 |
| 4.1. Informações pessoais..... | 37 |
| 4.2. Utilização de metodologias ágeis..... | 39 |
| 4.3. Comparação entre metodologias ágeis..... | 40 |
| 4.4. Impacto na utilização de metodologias ágeis..... | 40 |
| 5. Conclusão e trabalhos futuros..... | 45 |

1. INTRODUÇÃO

1.1 Contexto e motivação

Ao longo dos anos, novas tecnologias foram criadas para atender às crescentes necessidades humanas. Com o avanço da mobilidade, criaram-se novas necessidades que até então não eram atendidas por computadores e notebooks. A partir dos anos 2000, a internet começou a ser utilizada nos celulares, atendendo assim a essa demanda. Com essa nova tecnologia, criou-se uma nova necessidade, a de aplicações específicas para aparelhos celulares ou móveis. Não bastava apenas adaptar o que era desenvolvido para desktops e notebooks para o móvel, era fundamental a criação de softwares específicos para tal finalidade.

Além da necessidade de criação de softwares específicos para dispositivos móveis, também foi preciso modificar a forma como eles eram desenvolvidos. Modelos tradicionais de desenvolvimento de software como o cascata não se apresentavam adequados para o desenvolvimento móvel, dadas as características peculiares que esses dispositivos apresentam em relação ao desktop e notebook.

Exemplos disso são a grande volatilidade dos requisitos, grande quantidade e diversidade de dispositivos presentes no mercado, cada um com diferentes características e a necessidade de entregas rápidas. Para realizar essas entregas da forma requisitada, elas são feitas em pequenos módulos funcionais, que levam menos tempo para ficar pronto em comparação com o desenvolvimento de todo o software.

Uma metodologia que se apresentou muito eficiente para abordar esse problema é chamada metodologia ágil. De acordo com o Manifesto Ágil [1], suas principais características são:

- Pessoas e interações, ao contrário de processos e ferramentas.
- Software executável, ao contrário de documentação extensa e confusa.
- Colaboração do cliente, ao contrário de constantes negociações de contratos.
- Respostas rápidas para as mudanças, ao contrário de seguir planos

previamente definidos.

1.2 Objetivos

Estudos sugerem que metodologias ágeis podem ser utilizadas para adequar o desenvolvimento de software às necessidades das aplicações móveis [5]. Metodologias não adequadas podem atrasar, encarecer ou até mesmo inviabilizar este tipo de desenvolvimento, tendo em vista que se os prazos não forem cumpridos com as modificações que surgirem ao longo do desenvolvimento, a aplicação pode se tornar ultrapassada e não mais interessar ao cliente.

Este trabalho tem como objetivo realizar uma análise crítica das principais metodologias ágeis presentes na literatura que podem ser utilizadas para melhorar e facilitar o ciclo de desenvolvimento de software voltados para dispositivos móveis. Além disso, também tem como objetivo realizar um survey com desenvolvedores de software móvel que utilizam metodologias ágeis. O objetivo do survey é fazer uma comparação entre a literatura e a prática sobre a utilização de metodologias ágeis no desenvolvimento móvel.

1.3 Estrutura do documento

Este documento apresenta a estrutura apresentada a seguir:

- **Capítulo 2 (Revisão da literatura):** Neste capítulo será realizada uma breve revisão da literatura a utilização de metodologias ágeis no desenvolvimento móvel.
- **Capítulo 3 (Método de pesquisa):** Aqui são apresentados os meios utilizados para se alcançar o objetivo deste trabalho.
- **Capítulo 4 (Survey):** Através de um survey desenvolvedores da área móvel que utilizam metodologia ágil responderão perguntas que servirão de base para um comparativo com a literatura.
- **Capítulo 5 (Conclusões e trabalhos futuros):** Aqui serão apresentadas as

considerações finais e perspectivas para trabalhos futuros.

2. REVISÃO DA LITERATURA

2.1. Metodologias ágeis

Segundo (Dennis e Wixon, 2011), uma metodologia de desenvolvimento tem a função de formalizar a ordem de desenvolvimento de um software, organizar as fases do ciclo de vida do mesmo e ter o equilíbrio entre os processos e os dados.

Com o avanço das tecnologias e a grande quantidade de pessoas utilizando-as, cada vez mais são requisitados softwares com uma maior complexidade, fazendo com que as metodologias então empregadas pelas empresas de desenvolvimento se transformassem em inadequadas e não conseguirem se adequar a essas novas exigências.

A principal razão que transformou as metodologias tradicionais em inadequadas era sua rigidez, fazendo com os projetos não se adaptem às mudanças que aconteciam de forma cada vez mais constante, sendo necessário novas formas de produção, mais adaptáveis e que pudessem fazer entregas de forma mais rápida, mesmo que não contemplando todos os requisitos em cada entrega.

2.1.1. Histórico

Na década de 90 novas formas de desenvolver software começaram a ser adotadas por diversos times de desenvolvimento, essas novas formas surgiram da necessidade de se adaptar ao mercado de computadores que estava surgindo, o uso dessas práticas pelas empresas se proliferou na década de 90, estimando-se que nesse período o tempo médio entre a validação de uma regra de negócio e a implementação via software dessa regra era de aproximadamente três anos [3].

O resultado dessa demora é que quando o projeto não era cancelado ao longo de sua produção, ao seu término ele já estava ultrapassado e não correspondia às necessidades atuais das empresas, que eram diferentes de anos anteriores quando o software foi projetado.

Existiam áreas mais sensíveis como desenvolvimento aeroespacial que chegava a levar 20 anos entre o início do projeto e sua utilização na prática, o que frustrou diversos desenvolvedores. Áreas automotivas e de comunicação também enfrentaram esse problema, o que desestimulou diversos desenvolvedores e os levou a se reunir de maneira informal para debater o que poderia ser feito para melhorar essa situação.

Em 2001 um grupo de 17 desenvolvedores se reuniu para debater suas ideias e técnicas de desenvolvimento, criando assim um conjunto de valores que posteriormente ficou conhecido como Manifesto Ágil [1]. Este manifesto tem quatro valores principais, são eles:

| | | |
|-----------------------------------|-------------|-------------------------|
| Interação entre indivíduos | mais do que | Processos e ferramentas |
| Produto funcionando | mais do que | Documentação extensa |
| Colaboração com cliente | mais do que | Contrato |
| Resposta às mudanças | mais do que | Cumprimento de planos |

Tabela 1. Valores do manifesto ágil.

2.1.2. Conceitos

Além dos quatro principais valores propostos pelo manifesto ágil, também foram propostos doze princípios para tornar o desenvolvimento ágil, são eles [1]:

1. Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.
2. Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.
3. Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
4. Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto diariamente, durante todo o curso do projeto.

5. Construir projetos ao redor de indivíduos motivados, dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
6. O Método mais eficiente e eficaz de transmitir informações para (e por dentro) de um time de desenvolvimento, é através de uma conversa cara a cara.
7. Software funcional é a medida primária de progresso.
8. Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter indefinidamente passos constantes.
9. Contínua atenção à excelência técnica e bom design, aumenta a agilidade.
10. Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
11. As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.
12. Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

Esses princípios buscam nortear a utilização de metodologias ágeis de tal forma que quem a utilize possa alcançar os quatro valores presentes no manifesto ágil. É possível fazer um mapeamento entre os princípios e os valores que eles buscam alcançar.

Como exemplo, o princípio que diz que pessoas relacionadas aos negócios e desenvolvedores devem trabalhar em conjunto diariamente está diretamente relacionado ao valor de colaboração com o cliente, enquanto que o princípio de que mudanças são bem-vindas independentemente da etapa de desenvolvimento busca alcançar o princípio de resposta às mudanças.

2.1.3. Fatores sociais e humanos

Os principais fatores sociais e humanos envolvidos na utilização de metodologias ágeis são o impacto na cultura organizacional, a cultura colaborativa

imposta aos desenvolvedores e as implicações de estar inserido em uma equipe ágil. O feedback constante para todos os participantes do time das atividades que estão sendo desenvolvidas, além do comprometimento com os objetivos do time ser fundamental para o bom funcionamento de uma equipe ágil [12].

Feedback e consciência coletiva são essenciais, opondo diferentes culturas onde o individualismo e a falta de comunicação são comuns. A adoção dessas duas culturas geram um senso de conhecimento comum, uma concordância do que é um problema e como ele deve ser resolvido.

Reuniões diárias ou semanais são apontadas como o principal fator motivador para essa consciência coletiva. Essas reuniões têm a capacidade de gerar iniciativas que podem agregar benefícios para o time, tais como comentários sobre a arquitetura do software, requisitos não funcionais e feedbacks dos clientes que poderiam ficar perdidos por causa da falta de comunicação.

Outro aspecto que justifica esses encontros é a limitação da expectativa da equipe quanto ao que é possível e plausível de ser realizado em determinado período de tempo, como histórias e iterações.

Nessas reuniões são atribuídas as atividades que cada membro terá de realizar, acordado entre todos, e também o acompanhamento das atividades que estão sendo realizadas, deixando todos cientes do que cada um está fazendo, gerando assim a consciência coletiva.

Esse elevado nível de colaboração gera um grande nível de aceitação, satisfação e um sentimento de fazer parte da equipe, gerando assim melhores resultados.

Além dessas vantagens na realização de reuniões diárias, outro ponto positivo é que quando uma tarefa não é finalizada, o responsável por ela tem a possibilidade de se explicar perante a equipe, e caso necessário pedir ajuda ou explicações, ganhando assim o suporte dos demais desenvolvedores.

A repercussão negativa por um trabalho mal feito ou que não foi desenvolvido como planejado é a principal punição presente na utilização de metodologias ágeis, que é a punição social [12]. Esta punição gera um possível mal estar junto aos outros desenvolvedores, como a possível decepção apresentada quanto a qualidade do seu trabalho, o que acaba evitando que um trabalho seja desempenhado de

forma inadequada.

2.1.4. Principais metodologias ágeis

Existem diversos frameworks disponíveis dentro do universo ágil, cada um deles com características próprias que podem torná-los mais adequados para cada tipo de situação.

2.1.4.1. Scrum

Scrum representa um conjunto de valores, princípios e práticas que fornecem uma base para se integrar a outras técnicas e facilitar o desenvolvimento de software. Os elementos que formam essa base são:

- **Product backlog:** Funcionalidades a serem implementadas em um projeto.
- **Sprint:** Tempo definido que um conjunto de atividades deve ser desenvolvido.
- **Sprint backlog:** Conjunto de tarefas que um time se compromete fazer em um sprint.
- **Reunião diária:** Geralmente realizada pela manhã para debater o que foi feito no dia anterior, possíveis impedimentos e priorizar o que será desenvolvido no dia que se segue.
- **Sprint Review meeting:** Ao final de cada sprint a equipe deve apresentar as funcionalidades implementadas.

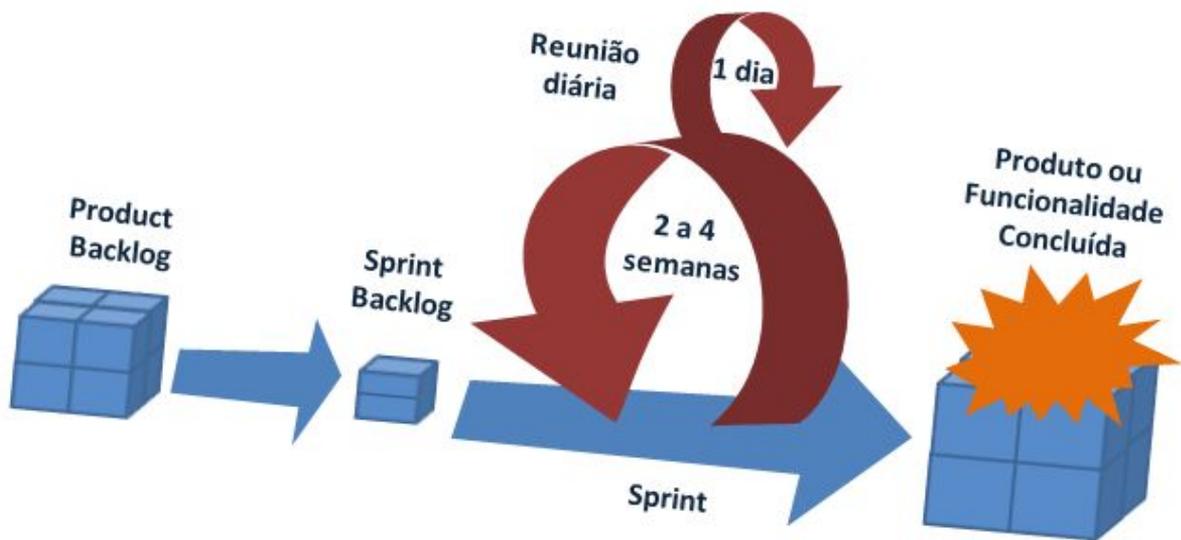


Figura 2. Fluxo do Scrum

No scrum os projetos são divididos em ciclos, chamados de sprints. Cada sprint tem um tempo pré-determinado e deve desenvolver um conjunto de atividades que ao final terão valor para o cliente. Uma lista chamada product backlog é mantida com tudo que deve ser feito no projeto, ao início de cada sprint um sprint planning meeting é realizado, que é uma reunião de planejamento junto ao product owner, idealmente o cliente, mas como nem sempre é viável, alguém da equipe de desenvolvimento pode representá-lo, para decidir as atividades que serão utilizadas no sprint que se iniciará, colocando essas atividades no sprint backlog.

Todos os dias uma reunião diária é realizada para debater o dia anterior, o que foi feito, caso alguma coisa não tenha sido realizada, deve-se determinar os motivos e como solucioná-los, além de decidir o que será implementado no dia que se segue. Por fim, ao término de cada sprint uma reunião é realizada, chamada sprint review meeting, onde são apresentadas todas as realizações que foram alcançadas.

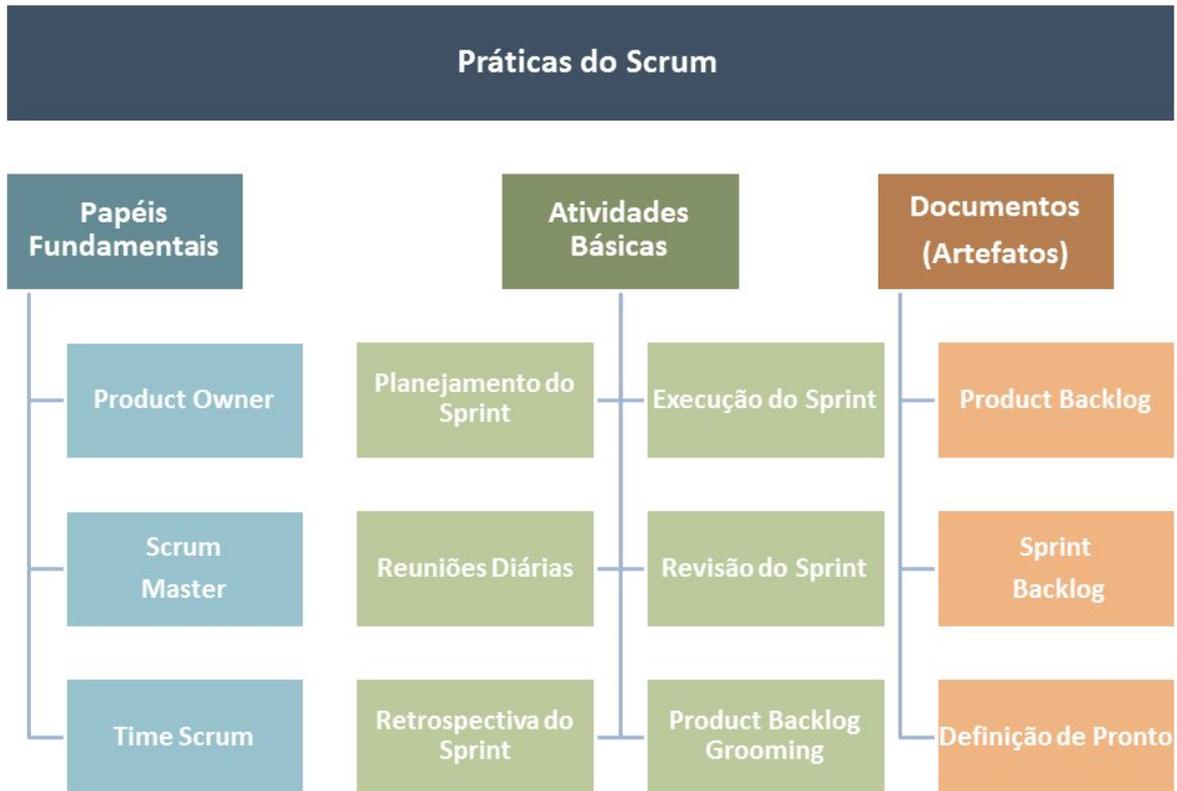


Figura 3. Práticas do Scrum

2.1.4.2. XP

A metodologia ágil denominada extreme programming (XP) nasceu nos Estados Unidos, ao final da década de 90. Seu principal objetivo é produzir softwares de uma forma mais rápida, econômica e em menos tempo do que utilizando técnicas tradicionais. Estes objetivos podem ser alcançados seguindo valores, princípios e práticas que os diferem das metodologias empregadas anteriormente a sua criação.

O principal objetivo no processo XP é a satisfação do cliente, para alcançá-la, ao invés de entregar tudo desejado pelo cliente em alguma data em um futuro distante, estimula-se a entrega constante das necessidades atuais do cliente. XP estimula o trabalho em equipe, todos os participantes do time de produção estão em um mesmo nível, como gerentes, desenvolvedores e clientes, o que torna o time altamente produtivo.

Existem cinco princípios que esta metodologia tenta melhorar em uma equipe,

são eles:

- Comunicação
- Simplicidade
- Feedback
- Respeito
- Coragem

Participantes de uma equipe de produção que trabalha utilizando a metodologia XP estão se comunicando constantemente, tanto entre si quanto com os clientes, utilizam um design simples e limpo, testando desde o primeiro dia de produção, fazendo entregas tão cedo quanto estejam prontas, implementando mudanças sempre que requisitados. Seguindo estes princípios, as equipes estão preparadas para responder a qualquer mudança, não importando em que etapa da produção eles estejam.

Para entender um pouco melhor o XP, é necessário entender as principais práticas executadas nesta metodologia, são elas:

- O cliente está sempre disponível:
 - O cliente deve estar presente o máximo possível no desenvolvimento para poder elucidar dúvidas, propor alterações e determinar prioridades.
- Planning game
 - Reuniões entre equipe de desenvolvimento e cliente, geralmente utilizando um quadro branco, cujo principal objetivo é a definição de user stories (regras de negócio expressas claramente em textos ou utilizando diagrama UML), estimando quanto tempo será necessário para execução dessas histórias.
 - Através dessas previsões, chega-se a um cronograma de desenvolvimento para o software solicitado, define-se um tempo padrão para cada iteração, definindo-se quantas histórias devem ser adicionadas em cada uma delas. A escolha de quais histórias serão

realizadas em cada interação leva em consideração as prioridades do cliente.

- Pequenas versões
 - A cada fim de interação uma pequena versão é liberada para o cliente, que realizará a validação do que foi feito. Desta forma é mais rápido e prático a realização de alterações nos requisitos.

- Integração contínua
 - Os módulos são integrados diversas vezes cada dia, sendo executados testes unitários, o código só será aprovado em definitivo após serem aprovados em todos os testes.

- Simplicidade
 - A equipe definirá padrões a serem seguidos por todos os membros, a todo momento, facilitando assim o entendimento de todos.

- Refatoração
 - Sempre que possível, a cada inclusão de um novo módulo, a refatoração deve ser executada, até que o design esteja da forma mais simples possível.

- Programação em dupla
 - A todo momento, dois programadores irão trabalhar compartilhando do mesmo computador, gerando, assim:
 - Compartilhamento de informação
 - Nivelamento do nível dos programadores
 - Aumento da cultura de coletividade do código

- Revezamento de pessoas
 - As duplas são constantemente trocadas.

- Padronização do código
 - Todo código será desenvolvido utilizando um mesmo padrão.
- Otimização da jornada de trabalho
 - Não deve haver sobrecarga de trabalho, o que diminuiria a produção dos desenvolvedores.

2.1.4.3. Kanban

O significado de kanban vem do japonês e significa placa visível, surgiu originalmente na Toyota, criado por Taiichi Ohno, que se baseou nos supermercados americanos, que somente reabasteciam suas prateleiras com novos itens quando elas se esvaziavam, como existia um espaço limitado, não era viável ficar sempre colocando novos itens.

Sua principal finalidade é a organização de processos interligados dentro de uma indústria, atualmente modificado para empresas em geral. Tem como objetivo o controle da produção e movimentação do material no processo produtivo.

A ideia básica dessa metodologia é tentar executar somente o necessário para determinado período de tempo, de forma que não exista trabalho em excesso, que fique à espera por muito tempo para ser executado. Desta forma, os elementos desnecessários são eliminados, só entrando na pauta quando de fato possam ser executados.

Os principais fundamentos do Kanban são:

- Produção nivelada
- Redução do tempo de preparação
- Padronização do trabalho
- Aperfeiçoamento do trabalho

O Kanban foi adaptado para o desenvolvimento de software, e para isso executa uma metodologia visual para gestão do fluxo de desenvolvimento. Princípios para evitar gargalos, delays e trabalho em excesso são utilizados, determinando que

o trabalho desenvolvido por uma empresa deve ser configurado de acordo com o que ela pode fazer, utilizando o conceito da teoria das restrições, que utiliza pelo menos uma restrição para limitar a performance da organização.

Para entender um pouco melhor o kanban, seguem os principais conceitos dessa metodologia:

- Ticket: Unidade de trabalho que deve ser desenvolvida
- Just-in-Time: Nada deve ser utilizado antes da hora exata
- Work in progress: Tarefas que estão sendo desenvolvidas
- Quadro: Local onde a parte visual é realizada, os tickets, etapas de desenvolvimento e quantidades de tarefas disponíveis por etapa estão claramente apresentadas para os times e está representado na figura 3.

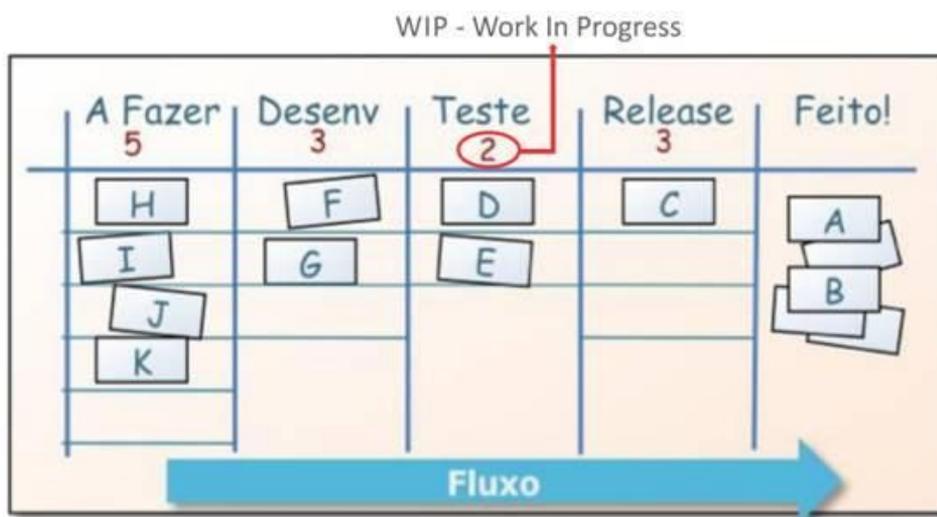


Figura 4. Fluxo do Kanban

2.1.4.4. Mobile-D

Segundo Abrahamsson [6], o desenvolvimento para dispositivos móveis apresenta diversas características que o tornam único e desafiador, devido a demandas específicas e limitações técnicas, como a rápida evolução dos

dispositivos móveis, diferentes tipos de padrões, protocolos e tecnologias de rede, necessidade de operar em diversas plataformas, necessidades específicas dos usuários que utilizam esse tipo de dispositivo e curto espaço de tempo para atender os requerimentos do mercado.

Para poder superar essas restrições, o mobile-D foi desenvolvido exclusivamente para o ambiente móvel. Ele foi baseado no XP, *Crystal Methodologies* e RUP. Essa metodologia é aplicável em outros contextos além do desenvolvimento móvel, como o financeiro, de segurança, logística e simulação de produtos.

Projetada para equipes pequenas, com menos de dez integrantes trabalhando no mesmo local e com o objetivo de entregar um produto funcional em curto espaço de tempo, geralmente menos de dez semanas.

Essa metodologia foi desenvolvida aplicando seus conceitos em três empresas que desenvolvem software e serviços para dispositivos móveis. Ela vem apresentando resultados positivos mesmo sendo avaliada por certificações rigorosas, como CMMI nível 2, que mede a maturidade de processos de software.

O desenvolvimento de software utilizando a metodologia Mobile-D se divide em cinco etapas bem definidas, são elas:

| | |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Exploração | Estabelecer os princípios iniciais do projeto além de fazer os preparativos para o desenvolvimento futuro. |
| Inicialização | Estabelecer e preparar todos os componentes da aplicação além de prever possíveis problemas que podem ocorrer ao longo do desenvolvimento. |
| Produção | Implementação dos requisitos funcionais. |
| Estabilização | Finalização da produção, incluindo integração se necessário. |
| Teste | Revisão se a aplicação foi desenvolvida como esperada. |

Tabela 2. Fases do Mobile-D

A cada uma dessas fases são atribuídas etapas e práticas que devem ser realizadas, como se pode observar na figura 4.

Fase de exploração

Na fase de exploração o objetivo é estabelecer as informações necessárias para o início do projeto e fazer a preparação para o desenvolvimento futuro. Nesta fase a equipe deve gerar um plano e estabelecer as características do projeto. Os objetivos são alcançados através de três práticas, identificar as partes interessadas, definir o escopo e estabelecer o projeto.

Fase de inicialização

Antes de iniciar o desenvolvimento propriamente dito, é preciso estabelecer tudo que será necessário para que tudo transcorra dentro do esperado. O mobile-D atinge esses objetivos através de quatro etapas. Na primeira delas, a configuração do projeto, tanto os recursos físicos quanto técnicos que serão utilizados devem ser identificados, treinamentos necessários para a equipe de desenvolvimento devem ser realizados, mecanismos de comunicação adequados com o grupo de clientes são criados.

As três fases seguintes são iguais às da fase de produção, dia de planejamento, dia de trabalho e dia de entrega. A diferença é que os objetivos nessa fase não são gerar valor para o cliente, e sim preparar todo o ambiente para a fase de produção.

No dia de planejamento, aqui chamado planejamento da iteração 0, deve-se chegar a um bom entendimento do produto que será entregue, tanto com relação a sua arquitetura quanto dos requisitos que serão implementados.

No dia de trabalho da iteração 0 deve-se confirmar que tudo está pronto para a fase de produção, caso a equipe de desenvolvimento decida produzir alguma funcionalidade nessa etapa, ela não precisa estar no topo da prioridade do cliente, mas sim ser considerada importante para o andamento do projeto, como por exemplo algo que interfira diretamente da arquitetura do projeto.

A última etapa dessa fase, o dia de entrega da iteração 0, não está bem definida pela metodologia mobile-D, pois depende do projeto que será implementado além do ambiente que ele utilizará, portanto não existe uma pré-definição do que

deveria ocorrer nessa etapa.

Fase de produção

A terceira fase é a implementação das funcionalidades utilizando o ciclo de desenvolvimento iterativo e incremental. Ao final dessa fase, a maior parte da aplicação deverá estar concluída. Essa fase é subdividida em três etapas, dias de planejamento, dias de trabalho e dias de lançamento.

Na primeira etapa deve-se planejar o que será desenvolvido na iteração atual. Os clientes devem participar dessa etapa para garantir a priorização correta dos requisitos que serão implementados, de modo que os mais importantes serão desenvolvidos primeiro.

Na segunda etapa deverá ser implementado uma versão totalmente funcional do produto. Para que a entrega ocorra como esperado, são utilizados conceitos como programação em pares, TDD (test-driven development), integração contínua, refatoração e a comunicação ao cliente do progresso real de implementação, podendo assim receber o feedback do que já foi implementado.

Na última etapa é realizada a entrega de uma parte do projeto que foi desenvolvida. Para isso, é necessário a realização da integração das diferentes partes do projeto, no caso dele ser muito grande, usualmente é subdividido em pequenas partes, que podem ser produzidos por diferentes equipes. Testes são realizados para confirmar que tudo está de acordo com o previsto. Aqui também há uma confirmação que os requisitos elicitados foram corretamente desenvolvidos. Caso tudo esteja conforme planejado, uma cerimônia de entrega é realizada, para confirmar que tudo ocorreu dentro do esperado e tudo que foi desenvolvido na iteração atual está pronto para servir de base para as iterações que se seguem.

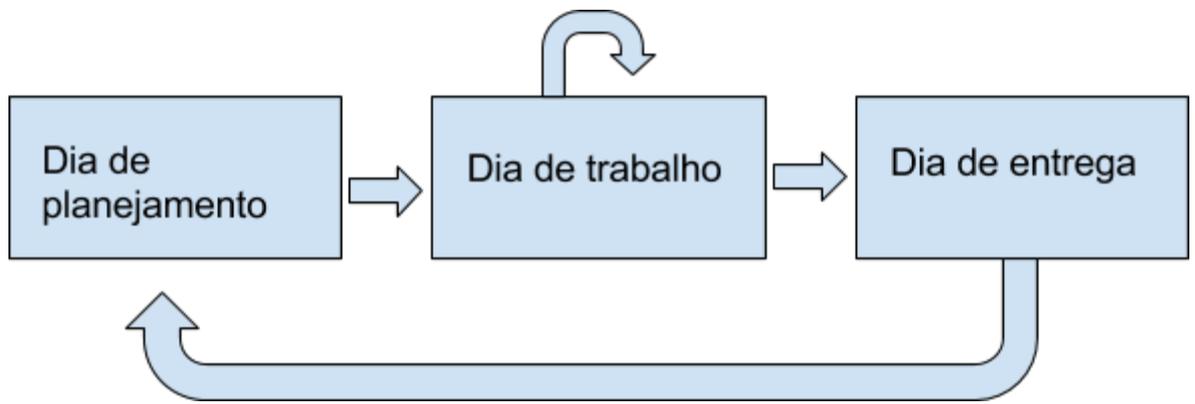


Figura 5. Fase de produção

Fase de estabilização e testes

As duas últimas fases se assemelham a fase de produção, com algumas alterações utilizadas para finalização e testes do produto. Na fase de estabilização a integração das partes do software que possivelmente foram desenvolvidas separadamente são integradas, além da documentação que necessita ser produzida.

Na fase de testes deve-se analisar se o software funciona exatamente como previsto pelo cliente, além de corrigir possíveis problemas que eventualmente sejam encontrados.

2.2. Comparação entre metodologias ágeis e tradicionais

Uma alternativa à utilização de metodologias tradicionais, também chamadas pesadas ou orientadas a documentação, é a metodologia ágil. As metodologias ágeis são ideais para projetos onde os requisitos são frequentemente alterados ou sofrem muitas mudanças, são implementados por pequenas equipes de desenvolvimento e a refatoração do código é uma atividade comum, além de possuírem entregas frequentes.

Metodologias ágeis surgiram com a proposta de aumentar o enfoque nas

pessoas e não nos processos de desenvolvimento [13]. Processos orientados a documentação limitam a capacidade de desenvolvimento das empresas, que muitas vezes não possuem recursos para esse tipo de processo.

2.2.1. Metodologias de desenvolvimento de software tradicionais

Para entender as metodologias tradicionais, é preciso primeiramente entender o contexto da época em que elas foram criadas. O desenvolvimento de software era baseado apenas em *mainframes* e terminais burros. Fazer qualquer alteração no código era muito custoso, tendo em vista que não existiam ferramentas de apoio ao desenvolvimento. Por esse motivo ele era todo planejado antes de ser produzido, pois era muito improvável que ele fosse ser alterado ao longo do desenvolvimento.

Dentre as metodologias tradicionais, destaca-se o modelo clássico ou sequencial. Neste modelo, uma sequência rígida deve ser seguida, e a cada mudança de fase uma documentação deve ser produzida, sendo necessário sua aprovação antes que a próxima fase se inicie.

São cinco as fases do modelo clássico, definição dos requisitos, projeto de sistema, implementação, teste do sistema e manutenção, como ilustra a figura 1.

O modelo clássico tem como características que cada fase se utiliza do que foi produzido na fase anterior e só se avança para fase seguinte quando o cliente valida e aceita os produtos finais da etapa atual.

Essas características trazem um nível elevado de rigidez, então esse modelo idealmente deve ser utilizado quando os requisitos são completamente compreendidos e dificilmente serão modificados ao longo do desenvolvimento.

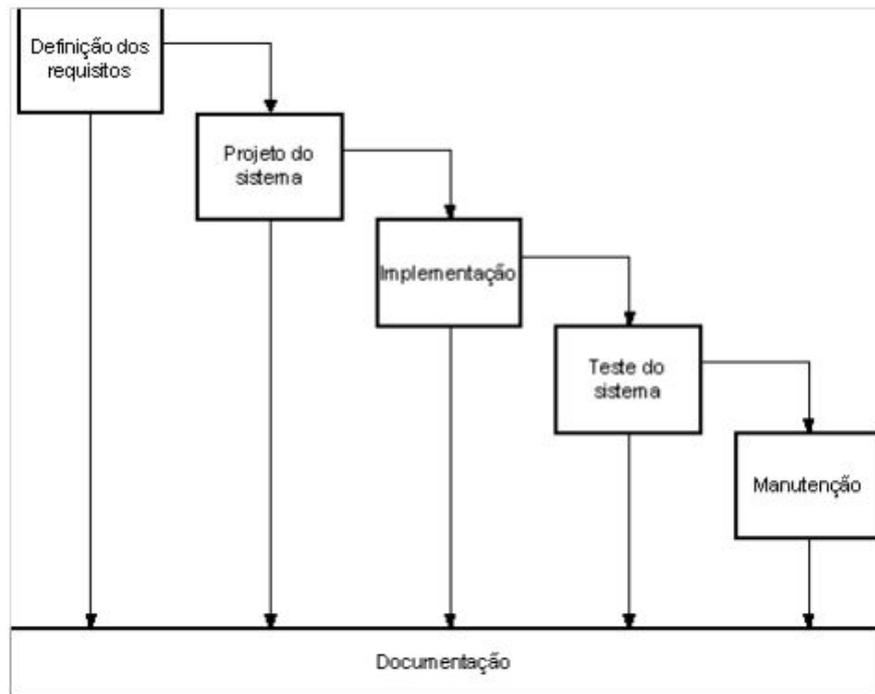


Figura 4. Ciclo da metodologia cascata.

2.2.2. Metodologias de desenvolvimento de software ágeis

Na década de 90 os princípios adotados pelas metodologias tradicionais começaram a ser questionados, os principais motivos para isso foram o alto índice de atraso, extrapolação do orçamento e a grande dificuldade para empresas de pequeno e médio porte adotarem as práticas robustas que essas metodologias utilizam.

Os conceitos chave das metodologias ágeis são a priorização das interações entre os indivíduos, software executável ao invés de documentação, colaboração do cliente ao invés de formulação de contratos e respostas rápidas a mudanças ao invés de seguir planos. Essas priorizações não significam que a documentação, negociação de contratos e planejamento são completamente esquecidos, apenas são deixados em segundo plano.

Metodologias ágeis dependem de competências individuais para que obtenham sucesso. Caso essas competências não existam, nenhum processo poderá fazer o projeto ter sucesso. Porém, pessoas trabalhando juntas com boa comunicação e interação executam tarefas de uma forma mais rápida e com melhor

qualidade [16]. Portanto, metodologias ágeis focam em um aumento tanto da capacidade individual quanto da coletiva para obter melhores resultados. Além disso, processos ágeis são feitos para priorizar os pontos fortes de cada equipe, adaptando os processos de forma a maximizar seu rendimento, extraíndo o máximo de cada desenvolvedor.

Equipe ágeis são caracterizadas pela organização própria e intensa colaboração. Organização própria não significa que elas não possuam liderança, apenas que elas têm liberdade para se organizar de diferentes formas até alcançar seus objetivos. Cabe aqui a diferenciação entre comunicação e colaboração, a primeira trata apenas da troca de mensagens, enviar e receber alguma informação, enquanto que a segunda é o trabalho em equipe para entregar algum produto ou tomar alguma decisão [16].

Existem pesquisadores que fazem ressalvas com relação às metodologias ágeis, são elas:

1. Não existe nada de novo no desenvolvimento ágil, suas práticas já são utilizadas desde a década de 60 [17].
2. Existe pouco suporte científico para várias reivindicações da comunidade ágil [18].
3. O desenvolvimento ágil é adequado para pequenas equipes, mas para grandes projetos outros processos são mais adequados [19].

2.2.3. Comparativo entre as metodologias

O que diferencia metodologias ágeis de tradicionais são o enfoque e o valor que cada uma delas emprega. Enquanto metodologias ágeis tem o enfoque voltado nas pessoas, metodologias tradicionais enfocam processos e algoritmos. Além disso, gasta-se menos tempo com documentação e mais com a implementação na metodologia ágil.

As metodologias tradicionais são baseadas na predição, cada etapa de desenvolvimento é baseada na anterior. Este tipo de abordagem só funciona se o sistema não sofrer nenhum tipo de modificação. Quando uma mudança é

necessária, as metodologias tradicionais resistem a essas mudanças, o que prejudica a evolução do sistema.

A metodologia ágil tem uma abordagem completamente oposta, ela espera que mudanças ocorram e prepara-se para tal. A medida que se conhece mais sobre o problema a ser resolvido, melhores soluções são descobertas. Ela se baseia no feedback e mudanças, ao invés de rejeitar essas mudanças, elas são absorvidas de forma a serem superadas de maneira mais simples.

A principal suposição da metodologia tradicional é que os sistemas são completamente especificáveis, predizíveis e são construídos através de um planejamento meticuloso, enquanto metodologias ágeis produzem software através de equipes pequenas, usando conceitos de melhorias contínuas e testes baseados no feedback.

A forma de comunicação empregada nas metodologias tradicionais é formal, documentando a maior quantidade de informação possível, ao contrário das metodologias ágeis, que se utilizam de uma comunicação informal, na maioria das vezes através de rápidas reuniões diárias. Na tabela 2 estão presentes as principais características tanto do desenvolvimento tradicional quando do ágil, podendo assim ser realizado um comparativo entre eles.

| | Desenvolvimento tradicional | Desenvolvimento ágil |
|---------------------------|----------------------------------------------|---------------------------------------------|
| Principal suposição | Sistemas completamente especificados | Melhorias contínuas |
| Forma de gestão | Comando e controle | Liderança e colaboração |
| Comunicação | Formal | Informal |
| Modelo de desenvolvimento | Ciclo de vida (cascata e espiral) | Modelo de entrega evolucionária |
| Controle de qualidade | Muito planejamento. Teste tardio e complexo. | Controle de requerimento e testes contínuos |

Tabela 2. Principais diferenças entre o modelo tradicional e ágil

2.3. Características do desenvolvimento de software móvel

Desenvolver software para dispositivos móveis apresenta diversas particularidades com relação ao desenvolvimento tradicional. Existem requisitos e limitações que devem ser respeitados ao longo da produção do software para os dispositivos móveis. Devido às limitações presentes, o desenvolvimento deve ser realizado com um alto nível de qualidade para poder se adequar aos diversos tipos de dispositivos presentes no mercado, além de poder se manter usável nos dispositivos que serão lançados no futuro [5].

As principais limitações presentes nos dispositivos móveis são suas limitações de hardware em comparação a dispositivos fixos, o perigo de estar sempre se locomovendo com esses dispositivos, a conectividade dos dispositivos móveis é altamente variável em performance e confiança além da quantidade de energia ser limitada. Devido ao seu tamanho, ergonomia e peso, as capacidades de processamento, armazenamento e memória são limitadas, o que exige uma atenção especial a esses requisitos para maximizar a eficiência na utilização de cada um desses recursos.

Dispositivos fixos geralmente estão armazenados em locais seguros contra furtos, são menos suscetíveis a perda ou quebra, além de ser possível a aplicação de diversas políticas de segurança contra outros tipos de risco, como desastres naturais.

Outro problema intrínseco aos dispositivos móveis é a conexão com a internet, o cenário em que o dispositivo está inserido pode ser alterado muito rapidamente, em um momento ele está em redes seguras e logo depois já muda para uma insegura, trocas entre velocidades rápidas e lentas, além de estáveis e instáveis ocorrem constantemente, basta que o usuário se mova alguns metros para que o sinal wifi, por exemplo, fique baixo. Por esse motivo é preciso que essas mudanças estejam previstas e sejam tratadas pelos desenvolvedores.

O consumo de energia talvez seja um dos principais problemas presentes em dispositivos móveis. De nada adianta ter uma aplicação excelente se ela consumir muita energia, de forma que fique inviável a sua utilização em locais que não possuam tomadas para carregar o dispositivo.

Em muitos casos é preciso utilizar políticas de backup, seja em um pendrive ou desktop, para garantir uma maior confiabilidade. É preciso que o desenvolvedor faça uma análise para atestar se essa política será necessária na aplicação que está sendo desenvolvida.

Diversos programas para desktop não tem a necessidade de se preocupar com a sincronização/replicação dos dados, já em aplicações móveis dificilmente se desenvolve algo que não exija comunicação com sistemas remotos, isso se deve a alguns fatores, como a quantidade de armazenamento disponível nesse tipo de dispositivo e a fragilidade com relação a perdas, furtos e quebras.

A sincronização acarreta outra dificuldade, dificilmente ela será feita de modo instantâneo devido a fragilidades de infra-estrutura, portanto a tecnologia de comunicação deve ser definida o mais rápido possível. Além desses problemas, existem outros que também devem ser resolvidos pelos desenvolvedores, são eles:

1. Sensores

Dispositivos modernos apresentam vários tipos de sensores, como o acelerômetro, *touch screen*, GPS, microfone, câmeras e até mesmo a força que a pessoa pressiona a tela.

2. Software e hardware

Aplicativos móveis relacionam-se com diversos tipos de hardware e sistemas operacionais, incluindo diversas versões do mesmo sistema.

3. Interface

Existe a necessidade de lidar com diversos guias externos de interface, que visam facilitar a usabilidade do usuário.

Outro aspecto que deve ser levado em consideração é a experiência do usuário, que tem grandes diferenças da experimentada pelos usuários quando utilizam aplicações desktop. Dispositivos móveis em sua grande maioria apresentam telas reduzidas, atrelado a isso possuem uma grande quantidade de formas de interagir com os usuários gerando assim uma gama muito grande de possibilidades de interação, que devem ser abordadas pelo desenvolvedor. O paradigma para desenvolvimento móvel baseia-se nos *widjets*, *touch*, movimento físico e teclado

(tanto virtual quanto físico), ao invés do tradicional WIMP (Windows, ícones, menus e ponteiros) [5].

Com a necessidade de utilizar cada espaço de tela disponível, dado seu tamanho reduzido, o design passa a ter uma importância fundamental. Outro aspecto de grande relevância relacionado ao design é o desejo e necessidade que os usuários têm de alcançar seus objetivos de forma rápida e com poucos cliques. Cada plataforma móvel possui suas próprias bibliotecas e regras de design a serem seguidas, para que diferentes aplicativos possuam aspectos semelhantes e facilitem sua utilização pelo usuário. Os desenvolvedores devem seguir regras para que a curva de aprendizagem do usuário seja a menor possível e ele não perca muito tempo procurando onde está a funcionalidade que ele deseja, ocasionando assim o desinteresse de vários usuários.

3. MÉTODO DE PESQUISA

Este capítulo apresentará a metodologia utilizada, dividido entre os objetivos gerais, objetivos do trabalho e por último uma descrição da aplicação da metodologia.

3.1. Visão geral

Segundo Fonseca (2002), *metodos* significa organização, e *logos*, estudo sistemático, pesquisa, investigação. Em outras palavras, metodologia é o caminho a ser percorrido para se realizar um estudo ou realizar ciência, a qual compreende o conjunto de conhecimentos precisos e metodicamente ordenados em relação a determinado domínio do saber [10].

O conhecimento científico deste trabalho foi obtido através da revisão da literatura sobre a utilização de metodologias ágeis no desenvolvimento móvel. Baseado na literatura foi feito um survey com o objetivo de entender as opiniões dos desenvolvedores que se adequam a esta situação. Posteriormente o resultado do survey foi analisado para se fazer uma comparação com o que está presente na literatura.

Para alcançar o objetivo deste trabalho, um survey foi realizado com desenvolvedores de software que possuem experiência no desenvolvimento móvel ágil.

3.2. Objetivos

A seguir serão apresentados os objetivos deste trabalho.

- Objetivos gerais
 - Apresentar as diferenças entre a utilização de metodologias clássicas e ágeis no desenvolvimento móvel.

- Apontar as principais metodologias ágeis que se adequam a criação de aplicações móveis
- Realizar uma comparação entre as diversas metodologias ágeis disponíveis para o desenvolvimento de aplicações móveis
- Apresentar vantagens e desvantagens entre essas metodologias

3.3. Aplicação da metodologia

A metodologia utilizada neste trabalho foi dividida em três etapas, como pode ser observado na figura 6. Na primeira delas, foi realizada a definição do tema, a revisão da literatura e definição da metodologia que seria utilizada. A partir das informações obtidas na revisão da literatura, um survey foi formulado e disponibilizado online para que desenvolvedores pudessem respondê-lo. Na última fase, foi realizada a análise dos resultados obtidos com os survey.

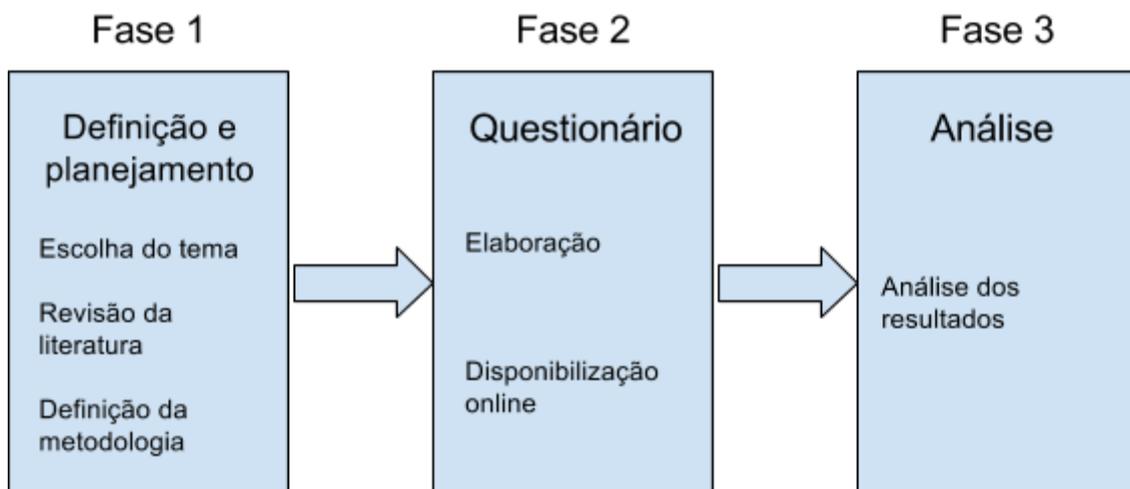


Figura 6. Etapas da pesquisa

Fase 1: Definição e planejamento

A primeira realização na fase de definição e planejamento foi a definição do tema que seria abordado, juntamente com a orientadora deste trabalho. Após o tema ser escolhido, uma revisão da literatura foi implementada. Foram consultados livros, artigos, monografias e websites para construção de um conhecimento sólido acerca

do tema.

Fase 2: Elaboração do survey

Para a construção do survey foram utilizadas as informações colhidas durante a primeira fase, na revisão da literatura. Após a finalização das perguntas, o survey foi disponibilizado na internet para que os desenvolvedores de aplicações móveis que se utilizam de metodologias ágeis pudessem respondê-lo.

O survey foi dividido em três etapas, a primeira coletou informações pessoais além dos tipos de projetos desenvolvidos pelas empresas dos respondentes. A segunda é acerca da utilização da metodologia ágil, buscando obter as vantagens e desvantagens na sua utilização. Por último, foram realizadas perguntas sobre os impactos que a utilização de métodos ágeis trouxeram para a empresa.

Fase 3: Análise dos resultados

Após finalizado o período de respostas ao survey, foi iniciada a fase de análise dos resultados obtidos, para tal análise foi feito um levantamento tanto qualitativo quanto quantitativo dessas respostas. Após essa análise, uma comparação com resultados obtidos previamente em outros surveys encontrados na literatura foi realizado, além de uma comparação com o que foi produzido na etapa de revisão da literatura.

4. RESULTADO DA PESQUISA COM DESENVOLVEDORES

Este capítulo analisará de forma comparativa as respostas obtidas através do survey disponibilizado na internet. O survey foi respondido por 14 pessoas e é dividido em quatro partes, a primeira traz informações pessoais sobre o desenvolvedor, a segunda sobre a utilização de metodologias ágeis, suas vantagens e desvantagens, na terceira foram abordadas comparações entre as diversas metodologias ágeis que podem ser utilizadas para o desenvolvimento móvel, enquanto que a última questionou sobre o impacto da utilização de metodologias ágeis no desenvolvimento móvel.

4.1. Informações pessoais

Acerca das informações pessoais foram realizadas cinco perguntas, a primeira delas foi sobre a idade dos participantes, tendo como resposta uma média de 26 anos. A segunda abordou o nível de escolaridade, onde 83% dos respondentes possui o título de graduação, enquanto que 17% possuem pós-graduação.

Na terceira pergunta foi abordado os tipos de aplicações que eram desenvolvidas pelos respondentes. Uma grande variedade de respostas foi obtida, como produção de SDK para ajudar outros desenvolvedores, entretenimento, aplicativo de mobilidade, jogos, etc.

A quarta pergunta teve como objetivo colher o tempo de duração para produção dos aplicativos, as respostas também foram muito variáveis, indo desde 1 mês até 1 ano, como pode ser visto no gráfico 1.

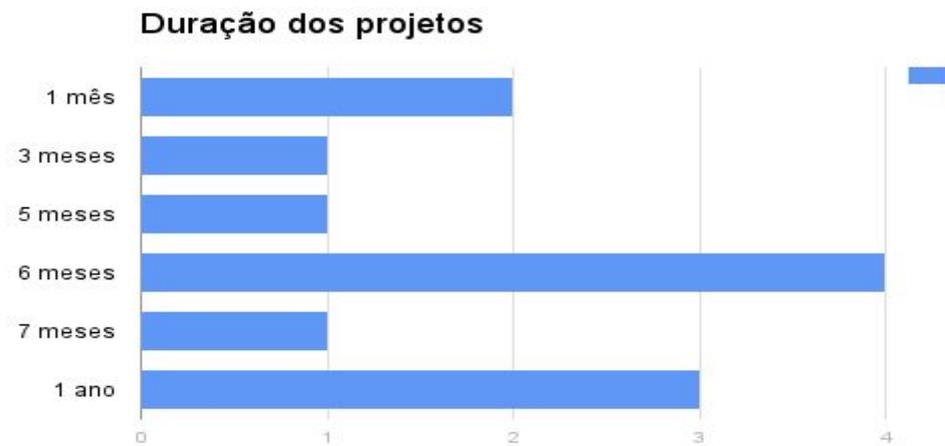


Gráfico 1. Duração dos projetos

Por último foi perguntado há quanto tempo a pessoa utiliza metodologias ágeis para o desenvolvimento de aplicações para dispositivos móveis, que pode ser observado no gráfico 2.

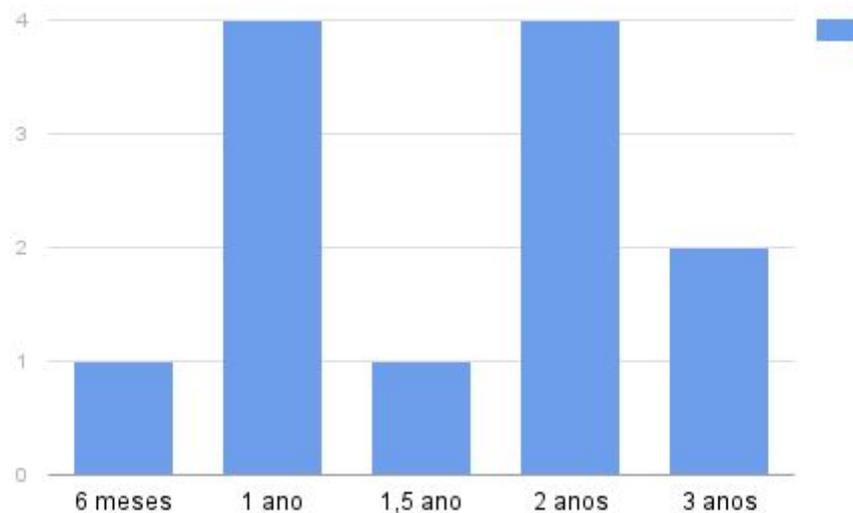


Gráfico 2. Tempo de utilização de metodologias ágeis

4.2. Utilização de metodologias ágeis

Quatro perguntas foram feitas sobre a utilização de metodologias ágeis, na primeira 100% dos entrevistados afirmaram que utilizam ou já utilizaram alguma metodologia ágil, o gráfico 3 apresentada quais metodologias foram citadas.

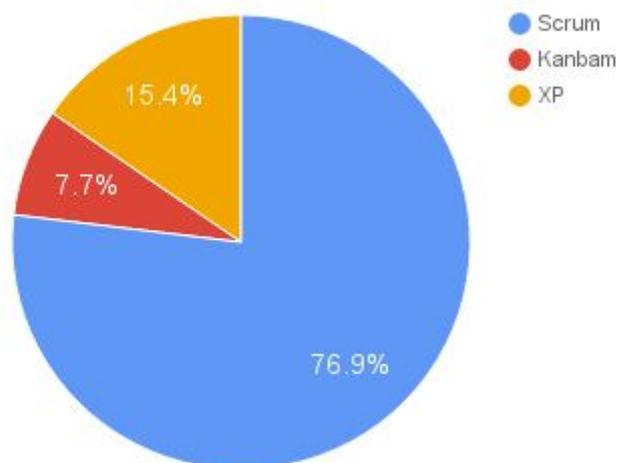


Gráfico 3. Metodologia ágil utilizada

83% dos participantes utilizaram apenas uma metodologia ágil na sua vida profissional, e destes 83% que utilizaram apenas uma metodologia ágil, 88% utilizam scrum, o que significa que de fato eles acham o scrum, e conseqüentemente a metodologia ágil adequada ao desenvolvimento móvel.

A pergunta número três não obteve nenhuma resposta, pois perguntava qual era o motivo para não se utilizar a metodologia ágil, no caso de desenvolvedores que não utilizem, porém 100% dos entrevistados disseram que utilizam alguma metodologia ágil.

Para finalizar essa etapa foi perguntado quais as principais vantagens e desvantagens na utilização das metodologias ágeis, o ponto mais abordado com

relação às vantagens foi a grande facilidade de adaptação às mudanças de requisitos que ocorrem ao longo do desenvolvimento. Dentre as outras vantagens que foram citadas pode-se destacar as entregas constantes em um curto espaço de tempo e a maior interação entre os desenvolvedores.

Com relação às desvantagens houve uma grande variedade de respostas, às principais foram a liberdade que as equipes de desenvolvimento possuem, o que pode causar sérios problemas ao longo do desenvolvimento caso a equipe não esteja comprometida, a grande quantidade de reuniões que existem e a falta de documentação.

4.3. Comparação entre metodologias ágeis

Apenas 17% dos participantes já utilizaram mais de uma metodologia ágil para desenvolvimento móvel, o que limitou bastante as respostas nesse grupo de perguntas. A primeira das perguntas foi sobre o motivo da mudança entre as metodologias, no caso do participante que respondeu, ele mudou do scrum para o kanban, e o motivo alegado foi que a segunda metodologia se adequa melhor as rotinas de correções dos bugs.

A segunda pergunta foi sobre qual a metodologia que mais se adequa à realidade do participante, a primeira resposta obtida alegou que o kanban é o ideal pois é mais adequados para ambientes de evolução de software e correção dos bugs, enquanto que a segunda resposta alegou que o XP é a melhor pois os prazos de entrega da equipe são muito curtos.

4.4. Impacto na utilização de metodologia ágeis

Nesta etapa do survey nove perguntas foram realizadas a fim de compreender quais impactos a utilização de metodologias ágeis trouxe para o processo de desenvolvimento.

A primeira pergunta foi sobre os impactos que a metodologia ágil trouxe para a empresa e equipe do participante, na grande maioria das respostas foi a maior velocidade de entrega, incorporação de mudanças de forma mais natural e simples

além de uma maior interação da equipe de desenvolvimento.

A segunda e terceira perguntas foram sobre a interação com o cliente ao longo do processo de desenvolvimento, 58% dos participantes responderam que o envolvimento do cliente ao longo do processo foi constante, enquanto que 42% disseram que a envolvimento é esporádico. Das pessoas que responderam que o envolvimento não foi constante, a principal justificativa foi que o cliente nem sempre se interessa pelos métodos. Já dentre os que disseram possuir uma interação constante, o que facilitou essa interação foram as entregas constantes, que facilitam o entendimento por parte do cliente do que ele realmente deseja.

As próximas duas perguntas foram relacionadas a como as equipes e empresas que os participantes estão inseridos seguem os princípios ágeis. Os gráficos 4 e 5 apresentam as estatísticas dessas perguntas. No gráfico 4 eles responderam, em uma escala de 0 a 10, o quanto eles julgam seguir os princípios ágeis, enquanto que no gráfico 5, na mesma escala entre 0 a 10, o quão próximo da literatura eles acreditam adotar esses princípios.

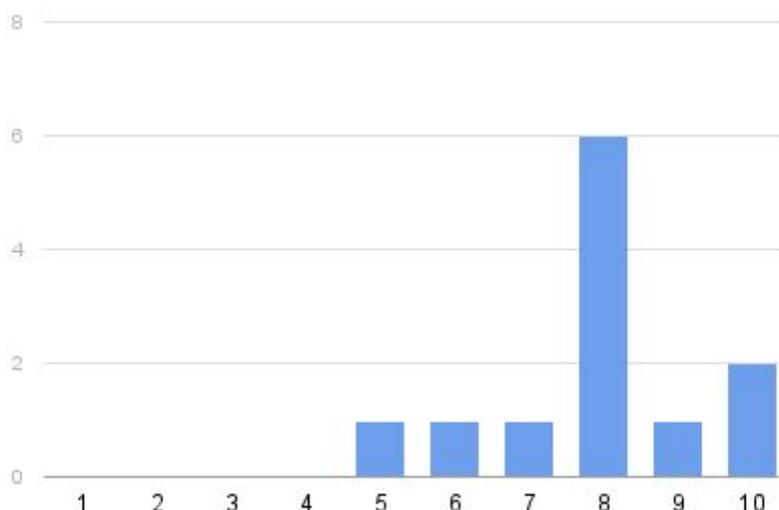


Gráfico 4. Aderência das equipes aos princípios ágeis

No gráfico 4, é possível observar que os desenvolvedores buscam seguir os princípios ágeis, à desenvolvimento ágeis implica processos disciplinados, e quanto mais indisciplinado for o desenvolvimento de um projeto maiores são os riscos dele falhar [].

Portanto, o fato dos desenvolvedores estarem seguindo a maior parte dos princípios implica uma maior chance de sucesso nos projetos desenvolvidos por eles, como menos atrasos e falhas, maior satisfação do cliente e conseqüentemente um melhor desempenho da empresa.

No gráfico 5, os respondentes avaliaram a relação entre a prática de como eles buscam alcançar os princípios ágeis e como a literatura afirma que eles deveriam. Em outras palavras, se eles seguem as metodologias ágeis de forma adequada. Como pode ser observado, a média é bem abaixo da observada no gráfico 4, isso se deve ao fato das particularidades presentes em cada empresa.

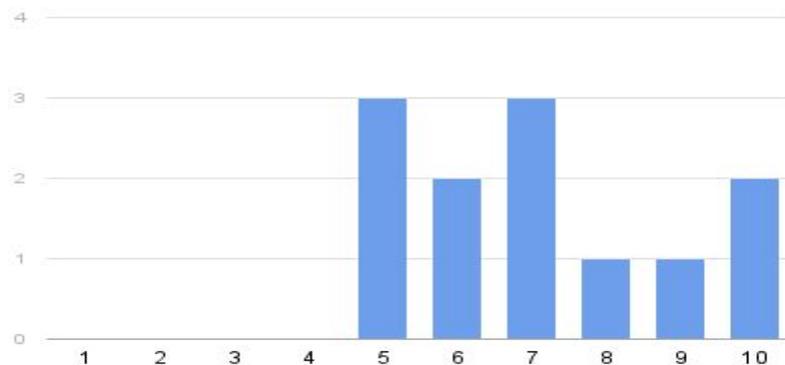


Gráfico 5. Comparativo entre literatura e prática

A discrepância entre os resultados se deve ao fato das diferentes necessidades de cada empresa e de seus respectivos grupos de desenvolvedores. Quando perguntados sobre as principais desvantagens na utilização de metodologias ágeis, uma das respostas foi a grande liberdade oferecida aos desenvolvedores, essa característica provavelmente foi alterada na empresa do respondente para dar menos liberdade, se adequando assim aos objetivos da empresa.

Cada uma das metodologias ágeis presentes no mercado tem o objetivo de alcançar os doze princípios ágeis, cada uma tendo formas diferentes de chegar a esses objetivos. Ao se alterar essas formas, deve-se ter muito cuidado para não fugir dos princípios, pois quanto mais distante deles maior a chance de fracasso do projeto [23].

As próximas três perguntas foram relacionadas a erros e atrasos nos projetos. 100% das pessoas afirmaram que houve uma diminuição no número de erros após a implementação de metodologias ágeis. Com relação aos atrasos, o gráfico 6 mostra as respostas relacionadas a diminuição ou não na sua incidência.

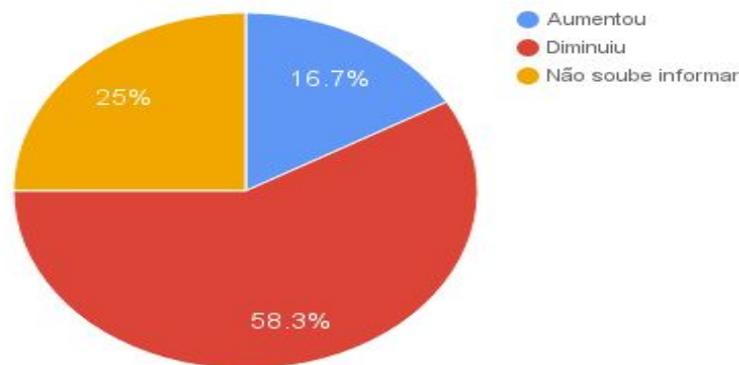


Gráfico 6. Comparação no número de atrasos

Segundo os participantes, o motivo para os atrasos continuarem a ocorrer estão majoritariamente associados a requisitos, podendo ser o mau entendimento, a mudança ou um erro na estimativa do tempo que será necessário para desenvolvê-los, outro motivo que também foi citado foi a falta de um planejamento adequado.

A última pergunta obteve diversas respostas, que em sua grande maioria corrobora os princípios ágeis e sua importância para o bom funcionamento do desenvolvimento móvel. A pergunta feita foi qual a principal lição é possível tirar da utilização de metodologias ágeis no desenvolvimento móvel, as respostas foram resumidas abaixo:

- **Requisitos:** As técnicas adotadas pelas metodologias ágeis facilitam a previsão de quanto tempo será gasto para desenvolver cada requisito, além disso, mudanças sempre são bem vindas.
- **Comunicação e feedback:** Ambos auxiliam bastante o desenvolvimento, pois fica mais fácil corrigir e problemas identificar os problemas.
- **Trabalho em equipe:** Juntamente com a comunicação, provocam uma maior interação entre os desenvolvedores, diminuindo assim o número de bugs e mal entendidos ao longo do processo de desenvolvimento.

5. Conclusões e trabalhos futuros

Este trabalho apresentou uma revisão da literatura acerca da utilização de metodologias ágeis no desenvolvimento de software para dispositivos móveis. Metodologias ágeis são consideradas ideais para o desenvolvimento móvel, pela sua rápida resposta às mudanças, previsão de interação constante com o cliente além de entregas rápidas, que garantem um alto grau de satisfação do cliente. Os ciclos de desenvolvimento são rápidos, com possibilidade de mudanças no projeto até mesmo perto da sua conclusão, essa característica talvez seja a principal vantagem da utilização deste tipo de metodologia ao invés de metodologias clássicas, que são rígidas e não permitem respostas rápidas às mudanças.

Baseado no survey realizado, a principal dificuldade encontrada pelos desenvolvedores foi a interação constante com o cliente. Nem sempre eles estão disponíveis para estar presente em todas as reuniões que os métodos ágeis sugerem que eles estejam. Para superar essa dificuldade, as empresas indicam uma pessoa da equipe para representar o cliente, tomar as decisões por ele, tanto para aprovar o que está sendo feito quanto para solicitar mudanças que sejam mais adequadas às necessidades do cliente.

As metodologias ágeis já são amplamente utilizadas pelo mercado no desenvolvimento de software para dispositivos móveis, porém as principais metodologias foram pensadas para o desenvolvimento desktop, e adaptadas para o universo móvel. Poucas pesquisas foram encontradas com metodologias ágeis exclusivas para o ambiente móvel, com o principal representante sendo o Mobile-D. Portanto uma expansão nessas pesquisas poderiam trazer um grande benefício para o desenvolvimento móvel.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] MANIFESTO ágil. Disponível em: <<http://www.manifestoagil.com.br/>>. Acesso em: 30 out. 2016.

[2] WHAT is agile. Disponível em: <<https://www.agilealliance.org/agile101/>>. Acesso em: 30 out. 2016.

[3] EXTREME Programming: A gentle introduction. Disponível em: <<http://www.extremeprogramming.org/>>. Acesso em: 02 nov. 2016.

[4]. EXTREME Programming – Conceitos e Práticas. Disponível em: <<http://www.devmedia.com.br/extreme-programming-conceitos-e-praticas/1498>>. Acesso em: 02 nov. 2016.

[5] A. I. Wasserman. 2010. Software Engineering Issues for Mobile Application Development. Acesso em: 02 nov. 2016.

[6] P. Abrahamsson, A. Hanhineva, H. Hulkko, T. Ihme, J. Jääliñoja, M. Korkala, J. Koskela, P. Kyllönen, and O. Salo. 2004. MobileD: An Agile Approach for Mobile Application Development. Acesso em: 02 nov. 2016.

[7] T. Ihme, P. Abrahamsson. 2005. "The Use of Architectural Patterns in the Agile Software Development of Mobile Applications." International conference on agility Helsinki. Acesso em: 02 nov. 2016.

[8] M. Satyanarayanan. Fundamental Challenges in Mobile Computing. Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing. (PODC 1996). Acesso em: 02 nov. 2016.

[9] METODOLOGIAS ÁGEIS PARA O DESENVOLVIMENTO DE

SOFTWARE: APLICAÇÃO E O USO DA METODOLOGIA SCRUM EM CONTRASTE AO MODELO TRADICIONAL DE GERENCIAMENTO DE PROJETOS. [S.l.: s.n.], 2013. Disponível

em: <<http://revistas.ung.br/index.php/computacaoaplicada/article/viewFile/1408/1194>>. Acesso em: 02 nov. 2016.

[10] MÉTODOS de pesquisa. Rio Grande do Sul: UFRGS, 2009. Disponível em: <<http://www.ufrgs.br/cursopgdr/downloadsSerie/derad005.pdf>>. Acesso em: 02 nov. 2016.

[11] DYBÅ, Tore ; DINGSØYR, Torgeir . Empirical studies of agile software development: A systematic review. Disponível em: <https://www.researchgate.net/publication/222827396_Dingsoyr_T_Empirical_studies_of_agile_software_development_A_systematic_review_Inf_Soft_Tech_509-10_833-859>. Acesso em: 05 nov. 2016.

[12] WHITWORTH, Elizabeth. The Social Nature of Agile Teams. Disponível em: <http://www.ewhitworth.com/documents/thesis/agile2007_WhitworthBiddle_social.pdf>. Acesso em: 02 nov. 2016.

[13] SANTOS, Michel . Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software. Disponível em: <https://www.researchgate.net/publication/222827396_Dingsoyr_T_Empirical_studies_of_agile_software_development_A_systematic_review_Inf_Soft_Tech_509-10_833-859>. Acesso em: 05 nov. 2016.

[15] SILVA DE OLIVEIRA, Ebenezer. Uso de Metodologias Ágeis no Desenvolvimento de Software. Disponível em: <<http://www.cpdee.ufmg.br/~renato/TesesEDissertacoesOrientadas/Monografia-EbenezerSilvaOliveira.pdf>>. Acesso em: 06 nov. 2016.

[16] COCKBURN, Alistair. Agile Software Development: The People Factor.

Disponível

em:<<http://www.uml.org.cn/softwareprocess/pdf/IEEEArticle2Final2.pdf>>.

Acesso em: 08 nov. 2016.

[17] MERISALO-RANTANEN, Hilka ; TUUNANEN, Tuure; ROSSI, Matti . Is Extreme Programming Just Old Wine in New Bottles: A Comparison of Two Cases.. Disponível

em:<https://www.researchgate.net/publication/220373787_Is_Extreme_Programming_Just_Old_Wine_in_New_Bottles_A_Comparison_of_Two_Cases>.

Acesso em: 09 nov. 2016.

[18]MCBREEN, Pete ; BECK, Kent . Questioning Extreme Programming.

Disponível

em:<https://www.researchgate.net/publication/234797626_Questioning_Extreme_Programming>. Acesso em: 09 nov. 2016.

[19]COHEN, David ; LINDVALL, Mikael; COSTA, Patricia . An Introduction to Agile Methods. Disponível

em:<https://www.researchgate.net/publication/222704289_An_Introduction_to_Agile_Methods>. Acesso em: 09 nov. 2016.

[21] B. Boehm and R. Turner, Balancing agility and discipline: A guide for the perplexed, Addison-Wesley, 2003. Acesso em: 09 nov. 2016.

[22] P. Abrahamsson, “Keynote: Mobile software development – the business opportunity of today”, in Proc. of the International Conference on Software Development, 2005, pp. 20-23. Acesso em: 09 nov. 2016.

[23]PAULK, Mark C. . Agile Methodologies and Process Discipline. Disponível em: <<http://repository.cmu.edu/cgi/viewcontent.cgi?article=1012&context=isr>>. Acesso em: 18 jan. 2017.