Universidade Federal de Pernambuco
Centro de Informática
Graduação em Sistemas de Informação

UM ESTUDO DE CAMPO DE USO DE AMBIENTES INFORMATIVOS DO NTI - UFPE

Rebeka Rayane Cunha Vogeley

Recife
2016

Universidade Federal de Pernambuco
Centro de Informática
Graduação em Sistemas de Informação

Rebeka Rayane Cunha Vogeley

UM ESTUDO DE CAMPO DE USO DE AMBIENTES INFORMATIVOS DO NTI – UFPE.

Trabalho de Conclusão de Curso apresentado ao curso de Sistemas de Informação da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação, sob orientação do Prof. Célio Santana.

Recife
2016

FOLHA DE APROVAÇÃO

UM ESTUDO DE CAMPO DE USO DE AMBIENTES INFORMATIVOS DO NTI – UFPE

Trabalho de Conclusão de Curso apresentado ao curso de Sistemas de Informação da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação, sob orientação do Prof. Célio Santana.

Data da aprovação: / /

Banca Examinadora:

_____

Prof. Dr. Célio Andrade de Santana Júnior

_____

Profa. Dra. Carla Taciana Lima Lourenço Silva Schuenemann

Recife
2016

Dedico esta monografia primeiramente à Deus pois sem ele nem estaria onde estou, também aos meus pais, Humberto Vogeley Silva e Nadja Cunha Vogeley que me deram todo apoio e suporte para alcançar este feito. Dedico também a minha irmã, meu noivo que sempre me deu estímulos em toda minha vida acadêmica e aos meus amigos que sempre estiveram presentes.

RESUMO

O trabalho de campo foi desenvolvido com servidores e bolsistas do Núcleo de Tecnologia da Informação (NTI) da Universidade Federal de Pernambuco, com o objetivo de investigar como a prática dos ambientes informativos estão presentes nos projetos do NTI. A motivação deste trabalho veio a partir da observância desta prática adotada pela organização. Foi feito um questionário, com o intuito de avaliar como o NTI utiliza os ambientes informativos em seu dia a dia, e para isso foram feitas questões de forma quantitativa e qualitativa. A análise quantitativa foi feita usando porcentagens e gráficos, enquanto a análise qualitativa se deu por fichamentos simples de texto. Através de análise dos dados recolhidos pelo questionário, foram extraídos alguns resultados bem conclusivos. Pôde-se notar que nem todos os respondentes sabem o nome do quadro usado em seu ambiente de trabalho, que é o Kanban, porém o número passou de 90%. Já o número foi de 100% para a importância do Kanban em vários tipos de projetos, onde desses, 81% preferem o Kanban na forma física, e não na forma virtual. Sobre o uso de fato do Kanban, foi possível concluir que em 100% dos casos acontecem reuniões em torno do quadro, mas que é possível acessar essas informações em outros sistemas como Redminer, OTRS e até o Trello, sendo possível dar andamento a atividade caso ocorra algum problema. Em suma, se tornou possível perceber que esse ambiente informativo age de forma positiva para todos.

Palavras-chave:. Ambientes Informativos, Espaço de trabalho informativo, Métodos Ágeis.

ABSTRACT

Fieldwork was developed with Information Technology Center workers of the Federal University of Pernambuco, with the purpose to investigate how the information environments are present in NTI projects. The work's motivation came from the observance of the practice of informative workspace in the center. A questionnaire with quantitative and qualitative questions was used to assess how the NTI uses the informative workspace in their work activities. The quantitative analysis was done using percentages and graphs, while the qualitative analysis was done by text analysis. Through analysis of data collected in NTI it could be done some very conclusive results. It might be noted that not all respondents know the table name used on your workspace, which is the Kanban, but the number is bigger than 90%. The number was 100% for the importance of Kanban in various types of projects where these, 81% prefer the Kanban in physical form, not in virtual form. On the use of fact kanban, it was concluded that in 100% of cases happen meetings about the board, but you can access this information in other systems such as Redminer, OTRS and even Trello, it is possible to progress the activity case a problem occurs. In short, it became possible to realize this informative environment acts in a positive way for everyone.

Key words: Informative Environments, Informative Workspace, Agile Methods.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização

O mundo vive numa constante mudança, e o Software é um elemento chave dessa evolução, logo é necessário que este acompanhe as mudanças de requisitos impostas pelo ambiente na qual ele está inserido, pois é a partir da utilização desses Software que empresas são estimuladas, incitadas e modernizadas com o intuito de aumentar sua competitividade.

O desenvolvimento de Software tem sido uma prática amplamente utilizada pelas empresas visando amenizar problemas, situações críticas de negócios ou requisitos técnicos. No entanto, a tecnologia de desenvolvimento, criação ou adaptação de um novo software apresenta-se como um fenômeno bastante complexo (FERNANDES, 2003).

E é nesse contexto de complexidade que entra a importância da comunicação em projetos de Software. A comunicação é uma importante ferramenta para o compartilhamento da informação, de conhecimento, e principalmente das necessidades e interesses. Ela se torna fundamental entre os atores envolvidos no desenvolvimento de Software, porém, obter uma comunicação eficiente ainda é um desafio quando tratamos de equipes de desenvolvimento, principalmente porque muitas vezes os colaboradores trabalham em horários flexíveis causando desencontros, ou até mesmo em turnos diferentes.

Essas falhas na comunicação entre a equipe, pode gerar vários problemas como, atraso do trabalho por falta de informação, retrabalho, erros, entre outros. Além dos problemas de comunicação entre os próprios colaboradores da equipe, também existem constantes falhas de comunicação entre equipe de desenvolvimento, e cliente e isso vem originando as principais falhas em projetos de Software uma vez que o que é compreendido como necessidade do cliente na maioria das vezes não é, de fato, o que ele realmente deseja.

Além do problema de comunicação, outros fatores tornavam o desenvolvimento de Software pesado. A quantidade de documentação e de burocracia para que um determinado sistema começasse de fato a ser construído levava tempo. Por vezes, o cliente já tinha mudado de ideia sobre requisitos, levando muitas vezes ao atraso do projeto, e até mesmo depois da entrega, o cliente não recebia aquilo que ele realmente idealizou, levando a uma insatisfação.

Essa foi a motivação para que na década de 1990 fossem buscadas maneiras de tornar o desenvolvimento de Software mais leve, menos engessado, e foi nesse momento que

surgiram os métodos ágeis, como uma alternativa ao desenvolvimento tradicional baseado no ciclo de vida em cascata. Entretanto, os tais métodos ágeis só foram formalizados e unificados em 2001 com a criação do Manifesto Ágil, quando dezessete membros da comunidade mundial de desenvolvimento de Software se reuniram para, além de outras atividades, discutir sobre boas práticas adotadas pelos profissionais no desenvolvimento de seus trabalhos como, questões que caracterizavam fatores para o sucesso dos projetos de Software, dos quais aqueles profissionais estiveram relacionados durante suas vidas profissionais.

Modelos baseados no desenvolvimento ágil começaram a surgir, tais como o Extreme Programming e Scrum, e é nos métodos ágeis que a comunicação ganha uma importância crucial. A gestão da comunicação e das informações nos projetos são tão importantes quanto a equipe, pois a comunicação constante e em tempo real dos desenvolvedores ataca os problemas de entendimento e, ao mesmo tempo, diminuem a necessidade de documentação formal e extensiva, diminuem as expectativas dos interessados no decorrer do projeto.

Então a comunicação é fundamental para se chegar na raiz de problemas que podem estar interferindo nos resultados, além de que com uma boa comunicação, todos os envolvidos podem exercer suas atividades em uma mesma direção, em busca de objetivos comuns. A comunicação gera informação e o principal objetivo do ambiente informativo é disseminar informação, além do entendimento implícito que o ambiente transmite. No instante que a mente recebe uma imagem ela logo associa a um significado, pois imagens são mais efetivas do que palavras, e esse é um conceito bem definido no meio da computação. Então se associarmos informações a imagens, ocorre o aumento da capacidade cognitiva do ser humano.

O mesmo acontece se introduzirmos esse conceito em um ambiente de trabalho, pois a todo instante a mente recebe informações não textuais e essas informações vão sendo agregadas, gerando conhecimento a todo instante só com o fato de se trabalhar em um ambiente com potencias fontes de informação. Outra vantagem de se ter esse tipo de ambiente, é a autonomia que os colaboradores adquirem, pois se a informação já está exposta, não se faz mais necessário a dependência de outra pessoa para se adquirir aquela informação, além de que as informações de status são transmitidas sem interromper os membros da equipe e ajuda a melhorar a confiança das partes interessadas.

1.2 Motivação

Não foi desde o princípio que a comunicação interna nas organizações foi valorizada ou reconhecida como importante para o desenvolvimento de Software e a consequente sobrevivência das organizações, mas, é através de uma comunicação interna eficiente que acontece a troca de informações úteis aos projetos. Quando o conhecimento é compartilhado isso é refletido diretamente no desempenho da equipe, aumentando assim a sua produtividade.

Gráficos, quadros, cartazes na parede com informações relevantes para a equipe, como por exemplo, a quantidade de defeitos descobertos em produção desde a última implantação do sistema, aprimora a comunicação da equipe e faz com que ela ganhe tempo, pois essas informações ficam disponíveis de forma que todos os envolvidos do projeto em questão possam ter acesso e estão sempre visíveis. Tais informações precisam ser ativas, ou seja, devem ser atualizadas com frequência para que sejam importantes, então, para que funcione é necessário que todos cumpram com a proposta do quadro, cartaz, imagem, dependendo da metodologia, ou seja, interajam, e realizem as alterações que forem precisos.

Considerando este quadro geral, a principal motivação deste trabalho veio a partir da observância de uma prática adotada no local onde trabalho que tem como intuito melhorar a comunicação entre os participantes dos projetos. Lá são utilizados alguns gráficos, quadros e relatórios “coloridos” com o intuito de informar os membros dos times a situação de determinada questões do projeto e foi baseado nessa observação que surgiu o interesse por entender de onde vieram e como funciona tal princípio, que só mais tarde foi descoberto como os ambientes informativos.

1.3 Objetivos

1.3.1 Objetivo Geral

Os objetivos gerais deste trabalho foram investigar como os ambientes informativos estão presentes nos projetos do Núcleo de Tecnologia da Informação (NTI) da Universidade Federal de Pernambuco. Entender como esses ambientes estão sendo utilizados, quais os projetos, e qual é o sentimento dos times em usar estes tipos de ambientes.

1.3.2 Objetivos Específicos

- Realizar uma investigação da literatura sobre ambientes informativos em projetos de Software.

- Identificar quais são as principais práticas, técnicas e métodos utilizados nestes trabalhos teóricos.
- Identificar no NTI-UFPE projetos que utilizam ambientes informativos.
- Criar um questionário para coleta de informações no NTI.
- A partir das respostas analisar o uso de ambientes informativos no NTI.

2 FUDAMENTAÇÃO TEÓRICA

2.1 Métodos Tradicionais

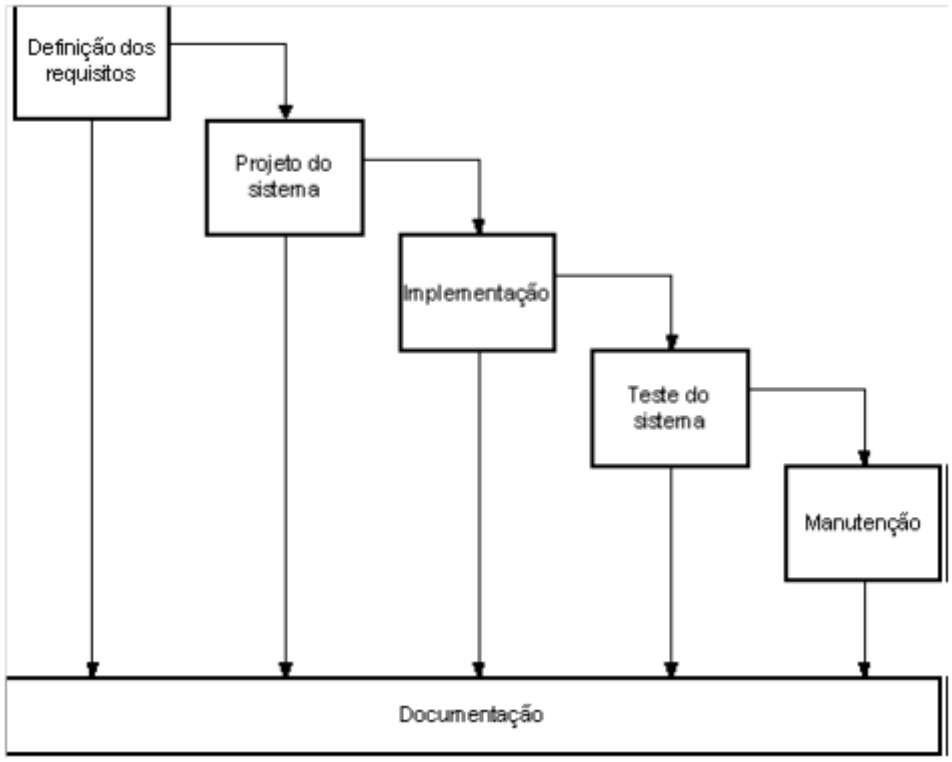
Modelos de Processos de Software foram criados para tornar a atividade de desenvolvimento de software menos caótica e visam organizar o desenvolvimento utilizando técnicas e métodos. Metodologia de Desenvolvimento é o conjunto de práticas recomendadas para o Desenvolvimento de Software. Essas práticas podem ser subdividas em fases, para ordenar e gerenciar o processo (SOMMERVILLE, 2008).

Por muito tempo o desenvolvimento de Software seguiu uma metodologia tradicional, mas, que ainda é muito usada nos dias atuais. Segundo Soares (2004) as metodologias tradicionais são também chamadas de pesadas ou orientadas a documentação. Elas foram muito utilizadas no passado em um contexto de desenvolvimento de Software muito diferente do atual, baseado apenas em um *mainframe* e *terminais burros.* Naquela época, o custo de fazer alterações e correções era muito alto, uma vez que o acesso aos computadores era limitado e não existiam ferramentas modernas de apoio ao desenvolvimento do Software, como depuradores e analisadores de código. Por este motivo o Software era todo planejado e documentado antes de ser implementado. Uma das metodologias tradicionais mais utilizadas até hoje é o modelo Clássico ou Cascata.

A metodologia Cascata, também conhecida por abordagem "*top-down*", representou um grande avanço no desenvolvimento de software pois foi uma das primeiras criadas para minimizar os problemas citados acima. Ele presume que o projeto terá uma sequência correta, sem desvios e sem problemas, não considerando os riscos (COSTA, 2004). Uma representação visual do modelo cascata pode ser visto na Figura 1.

A abordagem adotada pela metodologia cascata acaba trazendo alguns problemas. Dentre estes problemas merece destaque o fato de que os projetos reais dificilmente seguem o fluxo sequencial, o cliente quase sempre não consegue exprimir todas as suas necessidades além de ser exigida dele muita paciência visto que o Software só estará pronto para uso num ponto tardio do cronograma. E o maior dos problemas é que se ocorrer um erro em qualquer uma das etapas o resultado pode ser desastroso e frequentemente caro (PRESSMAN, 2006).

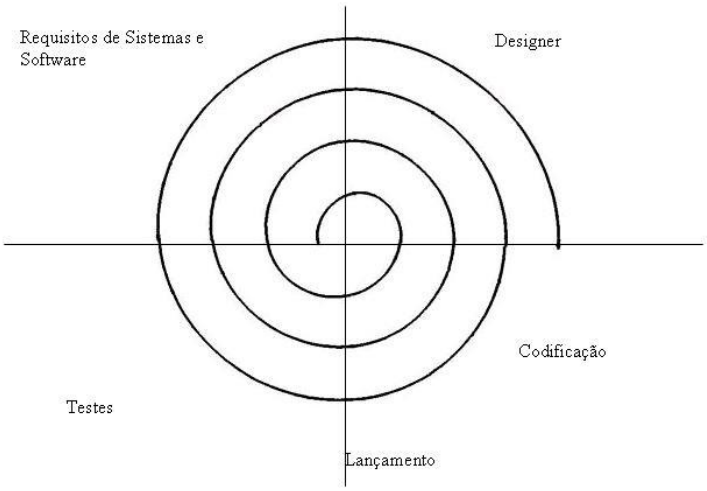
Figura 1 – Descrição do Modelo Cascata



Fonte: (GUSMÃO, 2013)

O ciclo de vida Espiral veio como uma alternativa para esse problema, que seria dividir a tarefa de levantar requisitos em etapas que não acontecem todas de uma vez só no início. O ciclo de vida espiral mantém as etapas no cascata, porém são executadas várias vezes ao longo do ciclo de desenvolvimento. E ao invés da entrega ser apenas de uma vez, ela é feita parcialmente ao longo do ciclo. A Figura 2 apresenta o modelo.

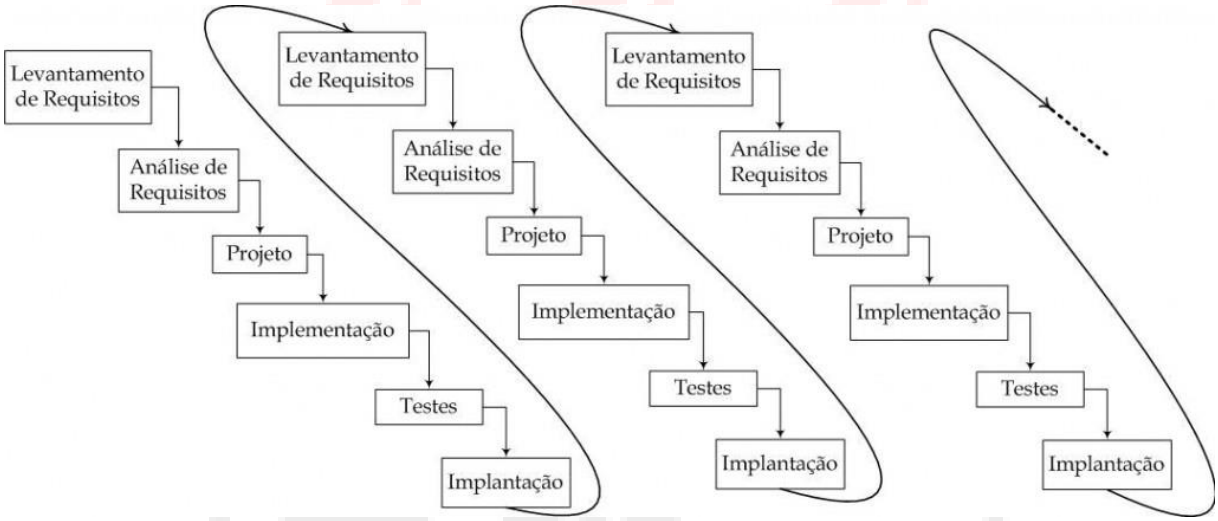
Figura 2 – Descrição do Modelo Espiral



Fonte: (JUNIOR, 2013)

Outro ciclo de vida surgiu um pouco mais tarde foi chamado de Iterativo e Incremental, tendo com princípio os mesmos que o ciclo de vida espiral, mas o termo incremental faz referência ao fato de que cada entrega significa aumento, ou incremento em relação ao cenário anterior. Podemos ver o funcionamento deste ciclo de vida na Figura 3.

Figura 3 – Descrição do Modelo Interativo e Incremental



Fonte: (MAROTO, 2012)

Mas devido a grandes pressões do mercado tecnológico por inovação, produtividade e flexibilidade, surgiu a necessidade de melhorar a forma como esses Software estavam sendo desenvolvidos. Com isso, alguns programadores começaram a se questionar sobre uma série de práticas adotadas em abordagens tradicionais de Engenharia de Software e Gerência de Projetos, e assim, em 2001, assinaram um manifesto que busca o desenvolvimento ágil de sistemas.

2.2 Métodos Ágeis

As metodologias ágeis surgem então como contraposto às metodologias tradicionais. Mas, em sua grande maioria, não aparecem com algo novo, mas mudam os enfoques e os valores do gerenciamento de projetos. O enfoque agora é direcionado a pessoas, e não mais em processos. Começa, também, a existir a preocupação em reduzir os custos com o tempo e documentação extensiva. A implementação passa a ocupar mais tempo, proporcionalmente, no projeto (COCKBURN, 2001).

O termo “Metodologias Ágeis” tornou-se popular em 2001 depois que um grupo de especialistas, em desenvolvimento de software, reuniram-se para compartilhar experiências e discutir medidas que aumentassem as chances de sucesso de um projeto.

Conforme o avanço tecnológico foi ocorrendo, à mudança de paradigma no desenvolvimento de sistemas aconteceu e atualmente se trabalha com o conceito de metodologia ágil. Agora com foco no código e otimizados para alterações de requisitos, já que o processo de criação de software na maioria das vezes é bastante mutável, pois o cliente sempre chega com novos requisitos, sendo assim necessário a constante modificação da documentação e do Software.

A essência do método ágil de gestão de projetos está em responder de forma mais rápida e menos dispendiosa às mudanças de requisito ocasionadas pelos clientes ou ambientes, no aumento da confiança na equipe de desenvolvimento e na entrega cíclica de versões ao cliente (COCKBURN, 2001).

Ou seja, uma forte característica das metodologias ágeis é que elas são adaptativas ao invés de serem preditivas, como disse Pressman (2006). Usam uma abordagem de planejamento e execução iterativa e incremental voltado para processos empíricos (complexos, caóticos ou com muita incerteza, tem mudança ao longo do processo, não são repetitivos e são imprevisíveis) além de que tem o propósito de focar na solução do problema, com o objetivo de acelerar as entregas do Software, utilizando prototipação, desenvolvimento incremental e times reduzidos.

Dessa forma passam a prioriza o produto de Software em si, sem focar tanto na documentação, e sempre com o objetivo de entregar de forma mais rápida ao cliente dividindo o problema em produtos menores e que visa entregar Software funcionando regularmente, atacando os riscos mais comuns e dessa forma o cliente poderia prover um feedback podendo fazer alterações, pois um dos conceitos do desenvolvimento ágil é que as mudanças e adições tanto no conteúdo quando no layout são bem-vindas para garantir a vantagem competitiva do cliente. Dessa forma, elas se adaptam a novos fatores durante o desenvolvimento do projeto, ao invés de tentar analisar previamente tudo o que pode ou não acontecer no decorrer do desenvolvimento.

Inicialmente, os métodos ágeis eram conhecidos como métodos leves. Apenas em 2001 quando membros proeminentes da comunidade se reuniram em Snowbird é que se

foi adotado o nome métodos ágeis com a publicação do Manifesto Ágil, documento que reúne os princípios e práticas desta metodologia de desenvolvimento. Mais tarde, algumas pessoas formaram a *Agile Alliance*, uma organização sem fins lucrativos que promove o desenvolvimento ágil.

2.3 Manifesto Ágil

Em 2001 então, 17 especialistas reúnem-se para propor um conjunto de princípios e valores para agilizar o desenvolvimento dos seus sistemas, tendo como base suas experiências em programação. Eles concluíram que os processos de desenvolvimento estavam se tornando cada vez mais burocráticos, dificultando a visibilidade e o entendimento das equipes de construção de software. Decidiram escrever então um documento que serviria como grito de guerra aos novos processos de desenvolvimento de software para a definição de um novo enfoque, baseado na agilidade, flexibilidade e nas habilidades de comunicação (BERNARDO, 2014)

Na primeira parte do documento contém uma declaração das crenças e valores. Através desse trabalho se passou a valorizar mais indivíduos e interação entre eles do que processos e ferramentas, pois o Software não é feito apenas por uma pessoa e sim por uma equipe, ou seja, existe uma interação, existem pessoas trabalhando juntas e isso é mais importante do que ferramentas e processos.

Também se passou a dar mais valor ao Software em funcionamento do que documentação abrangente, mesmo sabendo que essa documentação precisa existir para ajudar a compreensão do que é o sistema, mas é bem mais difícil entender como um sistema funciona sem vê-lo em funcionamento, então a documentação não seria o principal.

A colaboração com o cliente mais que negociação de contratos foi também abordada na reunião, pois com certeza ter um contrato é muito importante para a negociação, mas só o cliente sabe o que ele quer, e espera do sistema, além de que muitas vezes eles não sabem passar claramente o que querem e até mudam de ideia várias vezes, então a comunicação é de extrema importância para entender bem quais são as reais necessidades do cliente e melhor satisfazê-lo.

E por último e não menos importante, responder a mudanças mais que seguir um plano, pois a mudança hoje é uma realidade nos negócios, então o projeto precisa prever esse

tipo de acontecimento, sendo assim mais flexível até o ponto delas serem irrelevantes para o andamento do projeto.

Na segunda parte do documento podemos encontrar os 12 princípios que foram construídos ainda em uma parte da reunião e nos meses seguintes, e são eles:

- Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.
- Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
- Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
- Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.
- Construir projetos em torno de indivíduos motivados. Dando a eles o ambiente e o suporte necessário, e confiando neles para fazer o trabalho.
- O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face.
- Software funcionando é a medida primária de progresso.
- Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
- Contínua atenção a excelência técnica e bom design aumenta a agilidade.
- Simplicidade: a arte de maximizar a quantidade de trabalho não realizado é essencial.
- As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.
- Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

O Manifesto Ágil não rejeita os processos e ferramentas, a documentação, a negociação de contratos ou o planejamento, mas simplesmente mostra que eles têm

importância secundária quando comparado com os indivíduos e interações, com o software funcionando, com a colaboração com o cliente e as respostas rápidas a mudanças e alterações. (SOARES, 2004)

2.4 Métodos ágeis mais conhecidos

Cada método ágil existente hoje carrega consigo os valores e princípios do manifesto ágil, por isso são denominados ágeis.

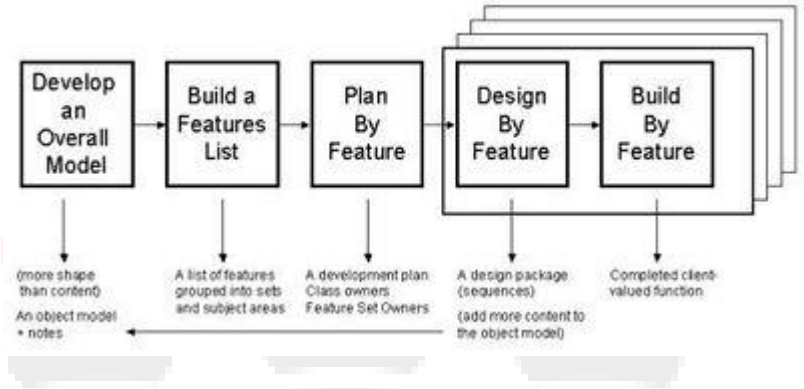
Dentre tantos métodos ágeis que existem hoje, uns são mais utilizados no meio organizacional que outros, são esses: Scrum, Extreme Programming (XP), Feature Driven Development (FDD) e o Crystal.

2.4.1 Feature Driven Development

Concebido originalmente por Jeff de Luca, FDD está entre as metodologias ágeis que existiam antes da criação do manifesto ágil, o FDD "Feature Driven Developement" surgiu em Singapura, entre os anos de 1997 e 1999 buscando o desenvolvimento por funcionalidade e é prático tanto para projetos iniciais como para projetos existentes. Entre ele e o XP existem diferenças, mas é possível usar as melhores práticas de cada metodologia no mesmo projeto. Também atua bem em conjunto com o Scrum, onde o Scrum atua no gerenciamento do projeto e o FDD no desenvolvimento. Um ponto que diverge do XP é que o FDD se incentiva o desenvolvedor como único responsável pelo módulo que este desenvolve, já no XP, o código é comunitário (Prática do código coletivo). Assim como outras metodologias o FDD pode ser adaptadas à necessidade de cada empresa. (GOMES, 2013)

FDD é um modelo prático para modelagem em orientação a objetos. É descrito como um processo adaptável e ágil, que pode ser aplicado em projetos de software moderados ou maiores. No contexto de FDD, uma *feature* (funcionalidade) pode ser descrita como uma função da necessidade do cliente que pode ser implementada em uma semana ou menos (PRESSMAN, 2006). A Figura 4 apresenta o passo a passo do FDD.

Figura 4 - Feature Driven Development



Fonte: (TECHNICAL KNOW HOW,2011)

2.4.2 Crystal

Crystal é uma família de metodologias de desenvolvimento de software e, como os cristais, possui diferentes cores e rigidez, referindo-se ao tamanho e ao nível crítico do projeto. São focados nas habilidades e nos talentos das pessoas, fazendo com que o processo de desenvolvimento seja moldado de acordo com as características da equipe, pois possui uma abordagem voltada a gestão de pessoas e seus princípios são voltados para cada tipo de projeto, unindo diferentes modelos de processo, mas com elementos centrais, comuns a todas, papéis, processos e práticas específicas de cada uma. (REIS, 2016).

Em Crystal cada organização pode avalie seu projeto por duas visões: número de pessoas e consequência dos erros, onde é definida graficamente por cores indicando o “peso” da metodologia. Assim, quanto mais escura a cor mais densa é a metodologia, ou seja, mais complexa. Portanto, a escolha da cor baseia-se no tamanho e criticidade do projeto. (FILHO, 2011)

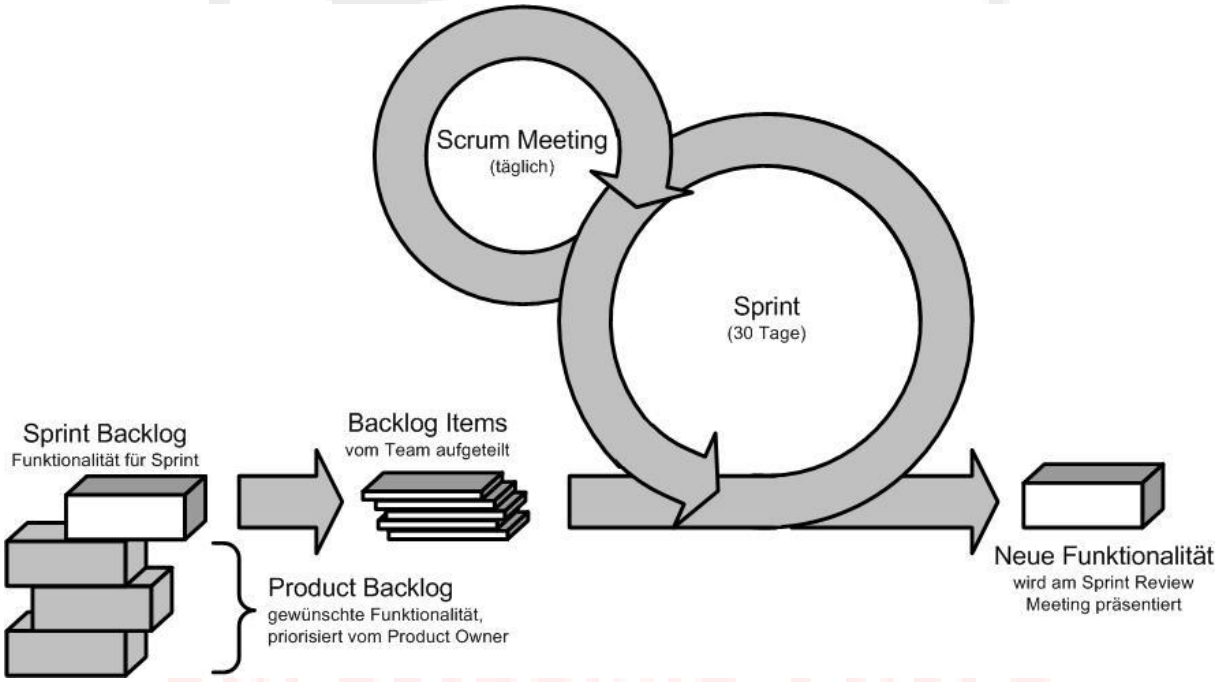
2.4.3 Scrum

Teoricamente, a melhor maneira de acelerar o processo de desenvolvimento é construir apenas o que o cliente valoriza e nada mais. O uso em exagero de formalização – planejamento extensivamente detalhado das tarefas e documentação em excesso – pode impedir o andamento do projeto, mas o contrário, ou seja, sem a utilização de algum modo de gerenciamento, pode gerar um cenário em que os objetivos do projeto não irão ser alcançados. O uso do Scrum – Metodologia Ágil para Gestão de Projetos – permite desenvolver projetos

bem mais adaptados à nova realidade das organizações de forma rápida. O foco do Scrum é descobrir uma forma que os membros da equipe trabalhem para produzir um software flexível em um ambiente de constantes mudanças (SCHWABER, 2004).

A maioria dos projetos em que se é escolhido inserir o Scrum é complexo e imprevisível. Ele provavelmente não vai solucionar todos os problemas do projeto, mas ajudará a percebê-los. Essa metodologia ágil serve como guia de boas práticas para o alcance do sucesso. Uma das características positivas do Scrum é a adaptabilidade, ou seja, a aplicação das mesmas práticas de formas variadas. Decisões de como usá-lo e criação de estratégias para chegar a uma produtividade maior e realizar entrega de artefatos mais rapidamente ficam por responsabilidade de quem está aplicando o processo (SCHWABER, 2004). A Figura 5 apresenta uma visão geral do Scrum.

Figura 5 - Visão do processo na metodologia Scrum



Fonte: (BERTHOLDO, 2010)

2.4.4 Extreme Programming (XP)

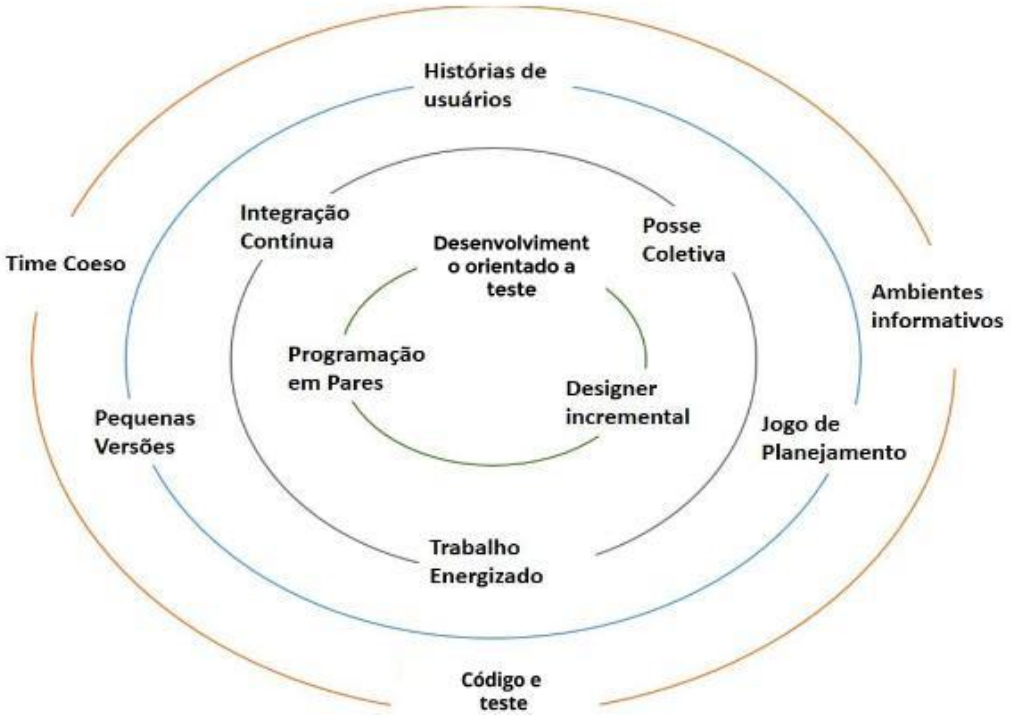
Extreme Programming (XP) possui algumas semelhanças com Scrum, mas enquanto o Scrum é considerado um framework para o gerenciamento de projetos, de qualquer tipo de projeto, O XP fornece algumas regras específicas para o desenvolvimento de Software em si,

nesse caso se pode fazer uso do framework Scrum juntamente com os conceitos do XP nos projetos de desenvolvimento de Software

Para Kent Beck (2000), criador da Extreme Programming, XP é uma metodologia ágil para pequenas e médias equipes desenvolvendo Software com requisitos vagos e em constante mudança. Extreme Programming usa times integrados de programadores, clientes, e gerentes para desenvolver Software de alta qualidade em velocidade alta. Reúne também um conjunto de práticas de desenvolvimento de Software já testadas, que quando estimuladas as sinergias entre elas, gerarão vastas melhorias em produtividade global e satisfação do cliente (JEFRIES, 2009).

Para conseguir atingir os valores e princípios no desenvolvimento de Software, o XP indica algumas práticas. Há uma ligação muito forte entre as práticas, pois elas se completam, fazendo com que os pontos fracos de uma, sejam recuperados pelos pontos fortes de outra. Na sequência, apresentamos as práticas, falando um pouco sobre cada uma (TELES M. , 2004): A Figura 6 abaixo apresenta todas as práticas que serão detalhadas a seguir.

Figura 6 – Resumo das Práticas de XP



Fonte: A Autora

Teste Primeiro: A ideia contida na prática do *test-first* é bastante diferente das práticas convencionais de teste, em que os testes são criados depois do código ter sido escrito.

No test-first, os testes são elaborados antes do programa, isto é, criam-se os casos de teste pensando em como será a implementação. E, a partir disso, o desenvolvimento consiste em fazer o Software que permita que o teste funcione. Essa característica fez com que essa prática fosse inserida dentro de um modelo de desenvolvimento chamado "*test-first development*", isto é, desenvolvimento dirigido por teste. (BECK, 2002)

Programação em Pares: Programar em dupla em um mesmo computador, onde a dupla é composta por um iniciante na linguagem usada e uma pessoa que domine a linguagem para servir de instrutor. O instrutor acompanha e assessora o iniciante, que fica à frente do computador. Com isso sempre busca a evolução da equipe, melhorando a qualidade do código fonte, pois o código acaba sendo revisto por duas pessoas, evitando e diminuindo a possibilidade de erros. (LEONEL, 2007)

Designer Incremental: O objetivo é criar a solução mais simples possível que seja suficiente para implementar as funcionalidades de cada iteração. Qualquer característica que possa ser implementada para dar apoio a funcionalidades futuras, só são codificadas de fato se e quando tais funcionalidades forem priorizadas para uma iteração futura. Dessa forma a equipe fica focada naquilo que com certeza será necessário, já que o cliente já tinha priorizado. E o que não foi feito deixa para ser feito quando realmente necessário. (TELES M. , 2006)

Integração Contínua: Quando produzir uma nova funcionalidade, sempre integra-la em menos de uma semana na versão atual do sistema. Pois demorando muito para integra-la, aumenta a possibilidade de conflitos e de ocorrer erros no código fonte. A integração contínua do sistema, faz com que você sempre possa saber o status real da programação. (LEONEL, 2007)

Posse Coletiva: Todos podem conhecer todas as partes do sistema. O código fonte não possui dono, com isso ninguém precisa pedir a permissão para modificar o código, tendo livre acesso. (LEONEL, 2007)

Trabalho energizado: Muitas vezes as pessoas esquecem que desenvolvedores são seres humanos, eles são levados a cargas horárias que chegam a fazem mal até a saúde. O XP segue a filosofia que mais importante não é trabalhar mais e sim de forma mais inteligente. (TELES M. , 2006)

Histórias de Usuário: As equipes planejam utilizado histórias que normalmente são escritas pelo próprio cliente em formato de cartão e tem o objetivo de armazenar uma história. Tendo isso, a equipe se reuni com o cliente para aprender o máximo possível sobre cada detalhe da história, e ai o cartão ira servir como m lembrete para cada história. (TELES, 2006)

Jogo de Planejamento: O desenvolvimento do software, é divido em várias etapas, e essas etapas são desenvolvidas semanalmente. No começo de cada semana é feito uma reunião chamada de Jogo de Planejamento, aonde o cliente e os desenvolvedores participam. O cliente tem papel fundamental na reunião, pois é nela que ele identifica as prioridades, e os desenvolvedores as estimam, com isso o cliente fica ciente do que está acontecendo, e o que irá acontecer. Como a cada começo de semana o escopo do projeto é reavaliado, o projeto gira em torno de um contrato de escopo negociável, que é diferente das formas normais de contratação de desenvolvimento de software. Ao final de cada semana, os desenvolvedores entregam o resultado desenvolvido durante a semana toda, podendo assim o cliente já colocar as funcionalidades desenvolvidas em ação. (LEONEL, 2007)

Pequenas Versões: A metodologia XP usa o desenvolvimento por partes, partes tão pequenas, que chegam ainda ser menores que as produzidas por outras metodologias incrementais, como o RUP. Conforme essas pequenas partes vão sendo desenvolvidas, a equipe de desenvolvimento libera para os usuários, e os usuários já colocam elas em ação. Com isso, os clientes vão tendo uma visão do sistema, auxiliando muito no processo de aceitação do cliente, que já pode testar uma parte do sistema. (LEONEL, 2007)

Código e Teste: Mantenha apenas código e testes como artefatos permanentes. Gere outros documentos que se façam necessários a partir do código e dos testes. Utilize mecanismos sociais para manter viva a história do projeto. Código e testes é uma prática fácil de ser seguida um pouquinho de cada vez. Um processo complicado, orientado a documentos, de cinco estágios, pode ficar mais leve um pouco de cada vez, à medida que a equipe se torna mais habilidosa. (TELES M. , 2004)

Time Coeso: A equipe de desenvolvimento é composta pelos desenvolvedores e também pelo cliente e quaisquer outras pessoas que devam ser ouvidas ao longo do desenvolvimento. (Leonel, 2007; Teles, 2004).

Ambientes informativos: O ambiente de trabalho de uma equipe XP deve ser um reflexo do projeto. Alguém que entre na sala da equipe deve conseguir obter, em poucos

segundos, uma noção clara de como está o andamento do projeto. Existem vários artifícios que tornam isso possível, e para isso a equipe XP utiliza cartões com histórias, quadro branco, *post it* sobre as paredes, *Flip chart*, gráficos, tudo isso na parede com informações relevantes. Aprimorando a comunicação da equipe e faz com que ela ganhe tempo, já que as informações mais importantes sobre o projeto estão sempre visíveis nas paredes. (TELES M. , 2004). Por ser o tema central deste trabalho, essa prática será mais bem detalhada na Seção a seguir.

2.4.4.1 Ambientes informativos

Apesar de ser uma prática criada no XP, hoje os ambientes informativos são adotados por muitas organizações que sequer ouviram falar de Extreme Programming, já que ao utilizar essa prática no local de funcionamento, independente da metodologia utilizada, esta melhora a comunicação como um todo.

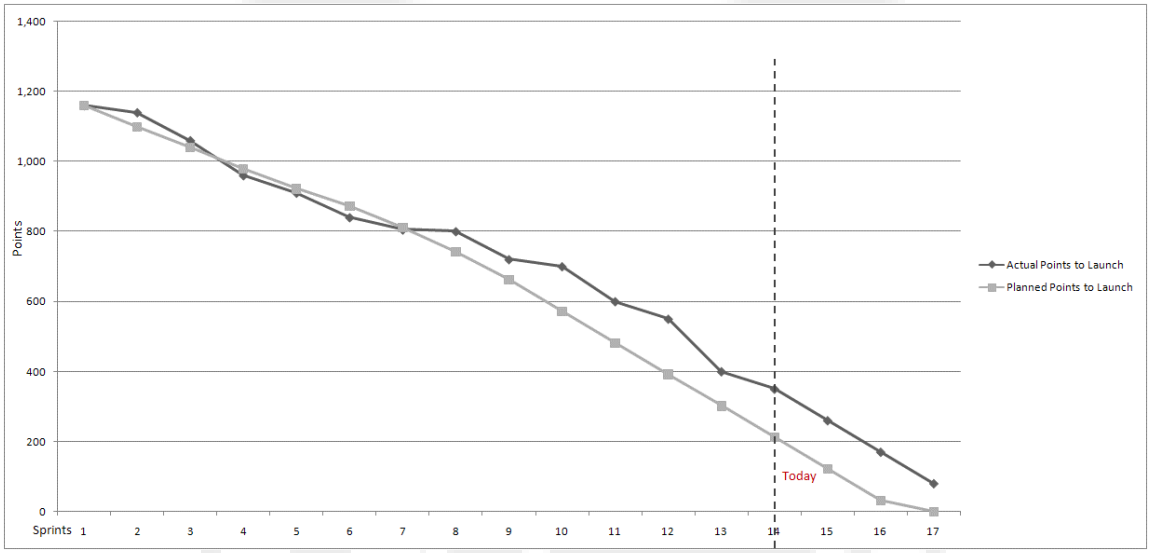
O ambiente informativo é uma das características mais marcantes de um projeto ágil, propõe a disponibilização de informações no local de trabalho de uma equipe ágil por meio de quadros e cartazes, de forma que fique visível para todos os envolvidos, o que está sendo feito no projeto, os objetivos, as preocupações, como está a versão de produção, enfim, a situação do projeto como um todo. Além de ser um propício para a interação social, facilita a partilha de conhecimento e melhora o desempenho da equipe, aumentando a sua produtividade (VALE, 2007; OIVEIRA, 2014).

Um espaço de trabalho com grandes quadros e gráficos visíveis que se comunicam. Os quadros devem ser imediatamente compreensíveis e evitar interpretações erradas. A informação que transmite deve ser fácil e rapidamente absorvida, sem a necessidade de interromper os desenvolvedores, pois, ao entrar no espaço de trabalho da equipe, um observador deve ser capaz de entender o andamento geral do projeto em apenas quinze segundos e observando mais atentamente, este também deve ser capaz de identificar problemas reais ou potenciais da equipe, pois o ambiente trabalho da equipe deve ser um reflexo do projeto. Os gráficos podem assumir várias formas, mas podem ser classificados geralmente como Radiadores de Informação e dispositivos de realimentação extremos (BAKER, 2005).

Os radiadores de informação são conjuntos de informações, apresentadas de maneira simples e com a maior exposição possível, para que todos recebam a informação necessária de maneira eficiente e normalmente requerem uma atualização manual onde podem ser usados

adesivos coloridos e Post-Its para fornecer informações adicionais e aumentar a visibilidade. Em métodos como Extreme Programming, Scrum, Crystal, e Lean Software Development, podemos encontrar a disponibilização de métricas e outras informações no ambiente de desenvolvimento, com o objetivo de gerenciar tarefas, lidar com problemas, e acompanhar essas métricas ou informações importantes. Os formatos utilizados para apresentar a informação podem variar de acordo com cada projeto, os mais utilizados são, o Burn Down Chart (utilizado para apresentação do andamento do projeto e controle de riscos), Earned Value Chart (para apresentação de índices financeiros, retorno de investimento e também para controle de riscos), Kanban (para visão de andamento de atividades), entre outros (NOVELLI, 2014). As Figuras 7 a 9 apresentam imagens dos três elementos a seguir.

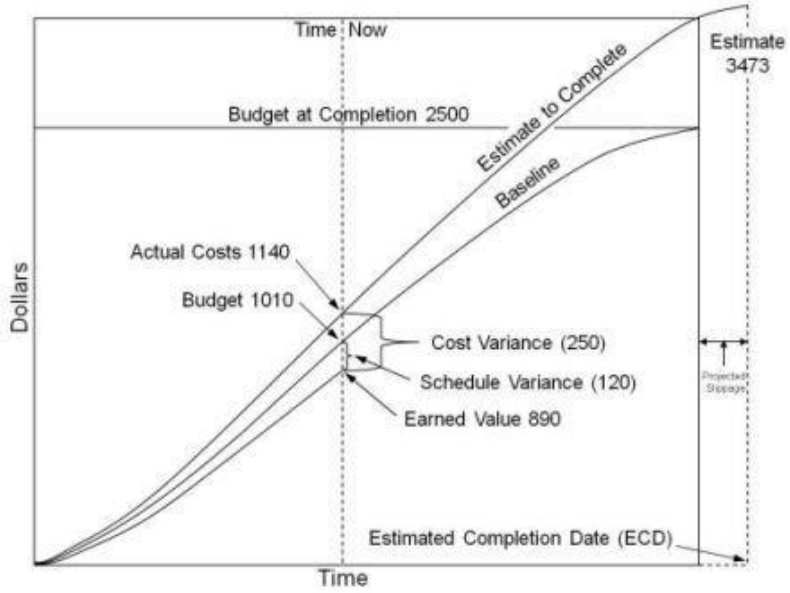
Figura 7 - Burn Down Chart



Fonte: (WATERS, 2011)

O Burndown chart ou gráfico de Burndown é o gráfico utilizado pelas equipes Scrum para representar diariamente o progresso do trabalho em desenvolvimento. Ou seja, após cada dia de trabalho o gráfico apresenta a porção de trabalho finalizada em comparação com o trabalho total planejado. É comum a Equipe de Desenvolvimento usar esse gráfico ao longo da Sprint, para medir os pontos das histórias finalizadas ao longo dos dias da Sprint e ter uma visibilidade do seu ritmo de trabalho, verificando se o ritmo está adequado para atingir a meta da sprint, cumprindo com o que foi planejado (CAMPOS, 2012).

Figura 8 - Earned Value Chart



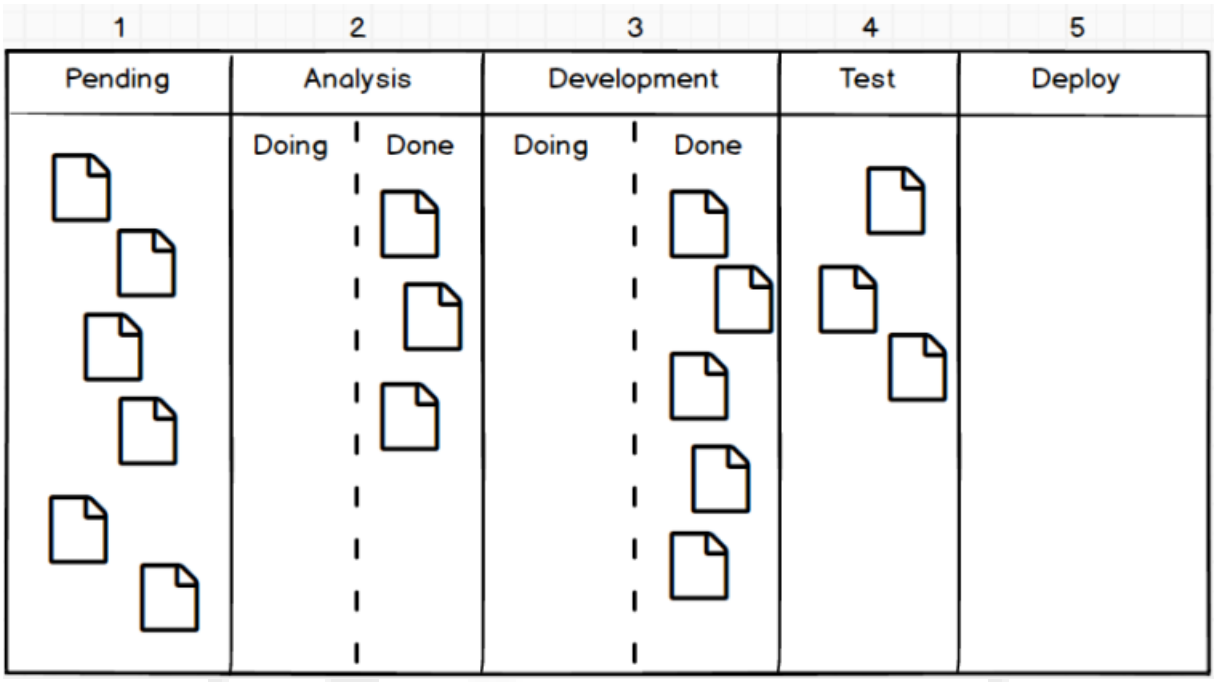
Fonte: (HERYMADAPASKA, 2015)

É uma metodologia utilizada para integrar escopo, cronograma e recursos em gerência de projetos, que consiste em medir objetivamente o desempenho e o progresso do projeto comparando custos (real e planejado) e valor agregado.

O kanban é uma simbologia visual utilizada para registrar ações. A palavra que tem origem japonesa pode ser traduzida como “cartão visual”. Foi criada pela empresa Toyota e integra o famoso sistema Toyota de produção, estando relacionada a sistemas puxados e ao conceito de entrega just in time, termo esse que embora seja confundido com o sistema Kanban, não significa a mesma técnica. (HENRIQUE E FIORINO, 2013)

Já os dispositivos de realimentação externos são indicadores simples e altamente visíveis, que transmitem informações sobre o progresso em termos de aprovação / reprovação testes unitários e de aprovação / reprovação testes de aceitação derivados automaticamente do processo contínuo de integração / build. O aspecto da interação física com um projeto de software puramente virtual, dá a todos em sua equipe uma visão instantânea para o estado atual do projeto. Estes dispositivos ajudam a criar a atenção adequada para situações como sistemas que quebraram. Normalmente a notificação é através da visualização ou som, mas se configurado de forma errada, o XFD pode ser extremamente irritante, ao ponto de alguém pegar e joga-lo para fora da janela (Paluch, 2012; Baker, 2005).

Figura 9 - Kanban



Fonte: (DETILIUM, 2016)

3 METODOLOGIA

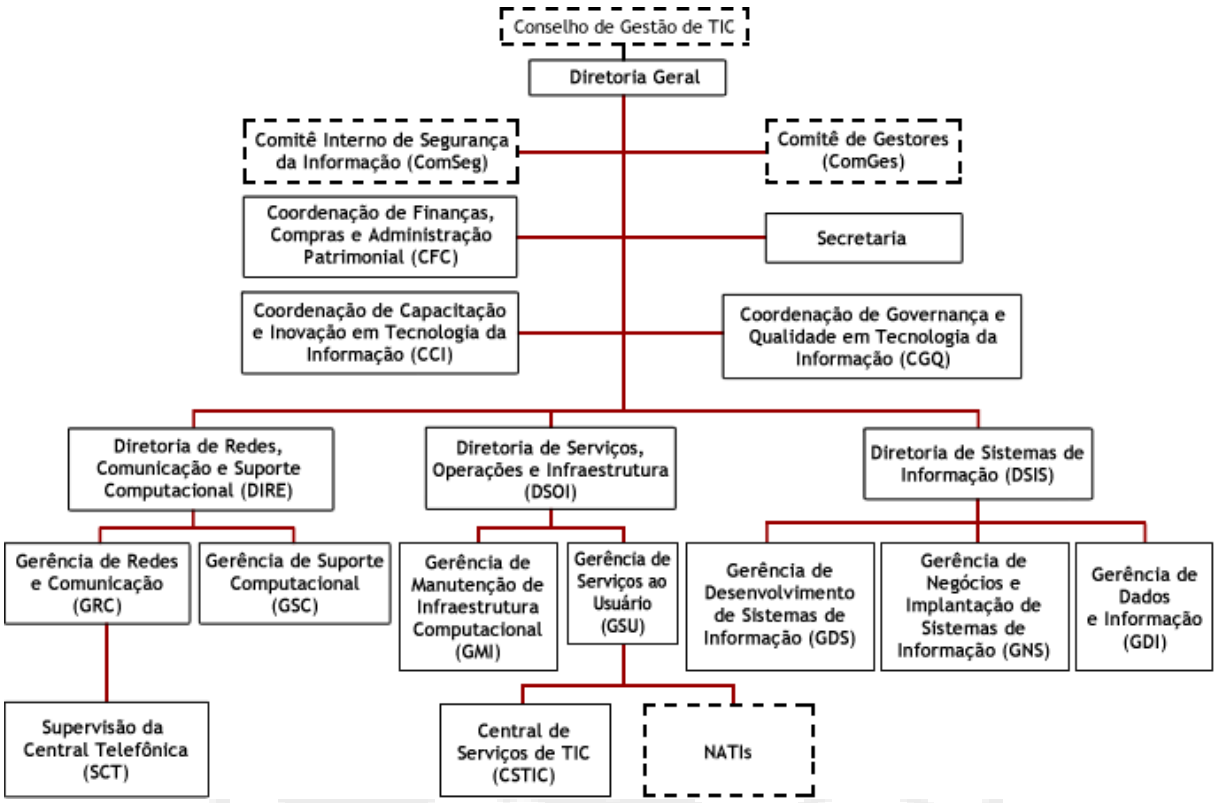
Para a realização desta pesquisa foi conduzido um estudo de caso no Núcleo de Tecnologia da Informação da Universidade Federal de Pernambuco realizado através de um questionário sobre o como é aplicado o ambiente informativo no contexto de trabalho do NTI, já que o mesmo utiliza de recursos como o quadro Kanban que é utilizado em praticamente todas as áreas, tornando o ambiente no NTI informativo. Para entender melhor o estudo de caso, se faz necessário apresentar o ambiente onde o mesmo ocorreu.

3.1 O NTI

O Núcleo de Tecnologia da Informação (NTI) é o órgão suplementar da UFPE responsável por realizar a gestão de infraestrutura de software e hardware da UFPE e o planejamento e execução da política de informática da universidade. O NTI tem também a responsabilidade de pesquisar, desenvolver, executar e participar de projetos em Tecnologia de Informação e serviços de informática, bem como de captar recursos através de projetos, consultorias e serviços. Foi criado em 1967, onde era responsável pela instalação e gerenciamento do sistema computacional da UFPE, e ao longo dos anos foi ampliando sua ajuda em diferentes áreas como desenvolvimento de sistemas, criação de projetos, desenvolvimento de pesquisas, tudo com o objetivo de atender melhor às necessidades acadêmicas. E foi na década de 90 que sofreu uma significativa expansão tecnológica, onde adquiriu novos e modernos equipamentos. E só no fim do milênio que foi batizado de Núcleo de Tecnologia da Informação, e hoje com quase 50 anos de serviços prestados (UFPE, 2009).

Logo abaixo pode-se encontrar a estrutura organizacional do NTI, onde recentemente sofreu mudanças logo após a realização das pesquisas para esse estudo de caso. Logo, quando as pesquisas foram realizadas a estrutura organizacional se encontrava dessa forma da na Figura 10.

Figura 10 – Estrutura Organizacional



Fonte: (UFPE, 2009)

Analisando a Figura 10 podemos perceber a presença da Coordenação de Governança e Qualidade em Tecnologia da Informação (CGQ) e vemos também a Diretoria de Sistemas de Informação (DSIS), esses dois setores do NTI fazem uso do Kanban para a realização das suas atividades e por isso foram escolhidos para a realização desse estudo de caso.

3.2 ETAPAS DA PESQUISA

A Figura 11 a seguir apresenta as etapas que foram realizadas na condução dessa pesquisa.

Figura 11– Etapas da Pesquisa

Fonte: A autora

Na primeira etapa foi realizada uma pesquisa bibliográfica sobre ambientes informativos e como estes eram usados na indústria. Esta fase se mostrou particularmente difícil, por não existir uma literatura disponível tão abrangente sobre este tema. A partir das restritas fontes de informação que foram encontradas, foi elaborado um questionário, segunda etapa, com o intuito de avaliar como o NTI utiliza os ambientes informativos em seu dia a dia. O formulário está disponível no Apêndice A deste trabalho.

A terceira etapa se deu a partir do preenchimento do formulário por parte dos respondentes. Os sujeitos que estariam aptos a participar da entrevista, universo da amostra, são os integrantes da Diretoria de Sistemas (DSIS) e da Governança e Qualidade em Tecnologia da Informação (CGQ) que totalizam 46. Deste total, 21 pessoas foram selecionadas, em comum acordo com a gerência das duas áreas.

A análise foi feita de forma quantitativa para as perguntas 1, 4, 5 e 6 por serem perguntas fechadas e uma análise qualitativa para as demais questões. A análise quantitativa foi feita apenas usando porcentagens, gráficos de pizza enquanto a análise qualitativa se deu por fichamentos simples te texto.

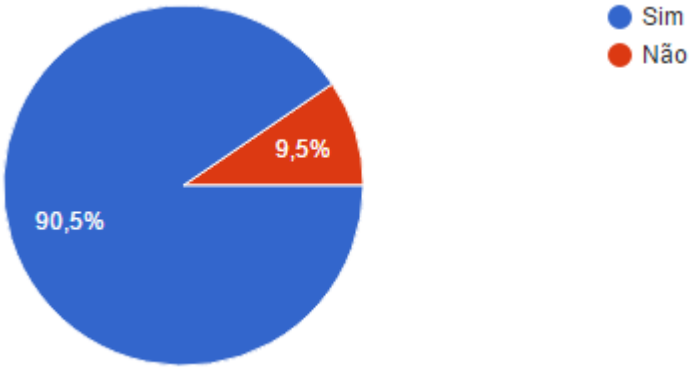
4 RESULTADOS DO QUESTIONÁRIO

A seguir serão apresentadas as respostas do questionário:

4.1 Questão 1 - Você sabe o nome do quadro com os *post-its* que é usado no seu ambiente de trabalho?

O quadro referente na questão se encontra presente na sala de todos os respondentes, onde todos fazem uso do mesmo. A Figura 12 mostra um gráfico referente as respostas.

Figura 12 – Resultado da Pergunta 1



Fonte: A autora

O quadro utilizado em muitos setores do NTI é o quadro Kanban, O sistema Kanban fornece um método simples, barato e fácil de implementar e rapidamente começa a apresentar resultados permitindo gerenciar o limite de atividades em andamento, e analisando o resultado pude concluir claramente que a grande maioria tem o conhecimento de como é chamado esse quadro usado em seu ambiente de trabalho, que pode ter sido obtido antes ou depois de trabalhar no NTI. Como sugestão para que esse número se torne 100%, poderia ser criada uma sessão na Wiki de ambientação que o NTI criou para pessoas que acabaram de entrar, falando e explicando a importância do Kanban para o NTI.

4.2 Questão 2 - O que você acha do quadro Kanban?

Na segunda pergunta foi uma questão aberta e a grande maioria dos entrevistados respondeu que era importante o uso do Kanban, pois ele possibilita uma comunicação visual do fluxo de trabalho, podendo acompanhar o andamento das tarefas, tanto as que foram feitas, quanto as que estão por vir, dando um engajamento maior para a equipe e até melhorando a produtividade. Mas também foi ressaltado que não é só a equipe que se beneficia, mas com o

auxílio das métricas, outras pessoas de fora da equipe também podem visualizar os resultados obtidos e o andamento dos projetos. Outra questão que foi bem abordada foram os impedimentos, com o uso de kanban se faz possível saber onde estão os gargalos do projeto, pontos de falhas, e pontos de melhorias, o que também influência na produtividade. Dentre todas as respostas dessa questão, apenas uma pessoa ressaltou um ponto que pode se tornar um problema para a equipe, pois um, ou vários membros da equipe podem não seguir o mesmo ritmo, o que criaria também um gargalo no projeto. Vendo as respostas pôde-se perceber que 100% acha importante o uso do Kanban nos projetos e processos.

4.3 Questão 3 - Em qual tipo de projeto você acha importante a presença do Kanban?

A terceira pergunta também era uma questão aberta e nela foram levantados alguns pontos de vista interessantes, alguns membros apontaram que deveria ser em projetos com equipes grandes, outros com equipes médias, uns falaram que poderia ser com equipes grandes ou pequenas, também que se aplicaria em projetos na fase de manutenção, projetos em que os papéis são bem definidos dentre outras respostas. Outra resposta frequente foi que a presença do Kanban é muito válida em projetos que necessitam de rapidez. A resposta mais comum, mais de 30% dos respondentes, afirmaram que o Kanban poderia ser aplicado em qualquer tipo de projeto.

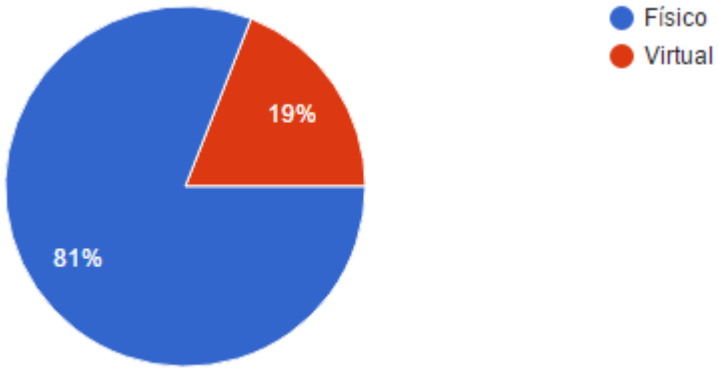
É necessário ressaltar alguns pontos para que o Kanban funcione como ele realmente deveria dentro de um projeto. É importante que a equipe esteja preparada para assimilar os conceitos e princípios do kanban, podendo acontecer de um ou mais membros da equipe criar uma resistência, pois o Kanban traz transparência, mostrando quem é produtivo e quem não é, por isso devem ser apresentado para a equipe o conceito do Kanban, e os benefícios que ele traz para o projeto. Identificar os Estágios de Trabalho que sua equipe segue para concluir um produto, projeto ou serviço, também Identificar as Classes de Trabalho, como, por exemplo [user storie], [bug], [defeito], [melhoria], [teste], [requisito], etc. Além de posicionar o que é mais prioritário e manter o controle, fazendo mudanças nas prioridades se for preciso. (MARTINS, 2013)

4.4 Questão 4 - Prefere: Físico ou Virtual?

81% dos respondentes preferem levantar, escrever e trocar o papel ao invés de utilizar o quadro virtualmente. Um dos entrevistados relatou que se fosse virtual seria como mais uma ferramenta dentre todas as outras que eles utilizam, e ele não teria a devida atenção,

e sendo na forma física, daria um engajamento maior. Na Figura 13 podemos observar as respostas de forma gráfica.

Figura 13 – Resultado da Pergunta 4



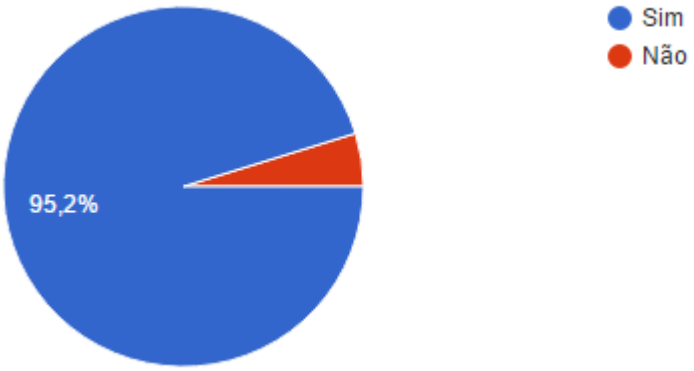
Fonte: A autora

4.5 Questão 5 - Você sabe para que serve cada raia/etapa (Mesmo aquelas que você não usa)?

Uma das coisas mais importante do quadro kanbam para uma equipe é que além dela entender o conceito do kanban, os ganhos que ele traz, ela também precisa entender o quadro montado para sua equipe, pois depois que sua equipe entender o funcionamento do Kanban, certamente sugerirá melhorias para que o quadro se torne ainda mais prático e visual.

A quinta pergunta se referia se o membro do time sabia para que serve cada raia/etapa, mesmo aquelas que você não usa. E como podemos observar no gráfico abaixo 4,8% não sabe quais são todas as raias. Esse problema poderia ser resolvido de várias formas, como uma reunião, ou até mesmo uma folha explicativa onde qualquer pessoa poderia ter acesso.

Figura 14 – Resultado da Pergunta 5

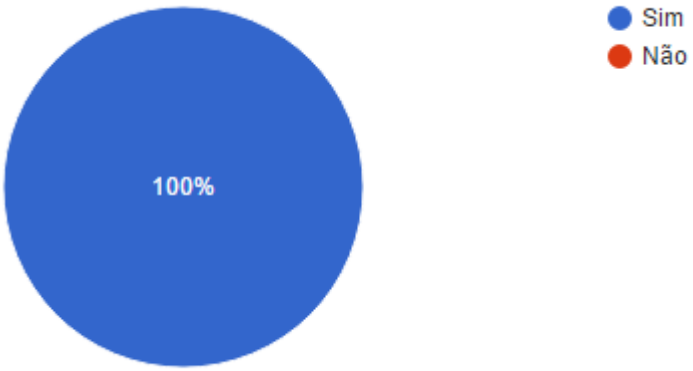


Fonte: A autora

4.6 Questão 6 - Acontecem reuniões em torno dele?

Perguntas como: O que eu fiz ontem? O que vou fazer hoje? Existem impedimentos no meu caminho? Podem e devem ser feitos em torno do quadro Kanban, isso acontece pois o Kanban é incompleto por natureza e requer algumas cerimônias, práticas, abordagens complementares que serão necessárias para se adequar ao contexto organizacional e ao time. E nesse caso 100% dos entrevistados responderam que sim, acontecem reuniões em torno do Kanban em seus setores, como mostra a Figura 15.

Figura 15 – Resultado da Pergunta 6



Fonte: A autora

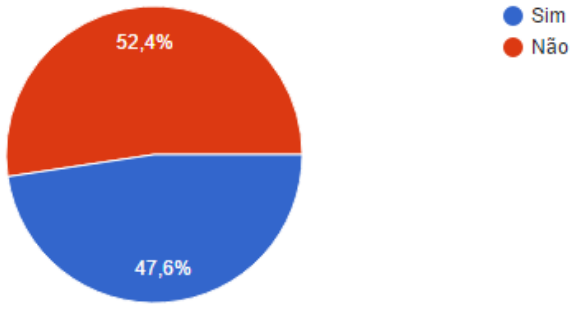
4.7 Questão 7 - Você depende dele para dar andamento a sua atividade? Porquê?

Aqui, muitos entrevistados responderam que sim, que foi criada uma cultura de analisar o quadro esperando que as pessoas da equipe troquem seus *post-its* de lugar, dando uma maior praticidade e facilitando o gerenciamento, já outras responderam que não necessariamente no entanto, com o quadro (físico) é possível ter um contato direito com a pessoa que está realizando a atividade que está "prendendo" a sua, trazendo com isso, uma certo grau de dependência no sentido de resolução de impedimentos. Quase 50% responderam que não, que ele serve para refletir o andamento do trabalho, mas o andamento do trabalho não é dependente dele, as informações contidas no quadro podem ser acessadas nos sistemas usados pela equipe. Em resumo deixaram claro que ajuda sim na transparência do projeto, mas que é possível acessar essas informações em outros sistemas, sendo possível dar andamento a atividade.

4.8 Questão 8 - Você sabe como o Kanban surgiu?

De forma surpreendente mais da metade dos entrevistados não sabe. No final de 1940, a Toyota encontrou um novo processo de engenharia que poderia ser empregado em seus negócios, dentro de um supermercado. Eles perceberam que os funcionários apenas reabasteciam as prateleiras quando o produto estava perto de se esgotar, o espaço para cada item era limitado e por isso somente quando necessário novos itens eram adicionados. Engenheiros da Toyota perceberam que uma peça poderia ser produzida apenas se a anterior a ela já estivesse vendida. A Toyota passou então a utilizar o kanban (BERNARDO, 2014). Como podemos ver na Figura 16, 52,4% das pessoas que fazem uso do Kanban em sua rotina não sabe como ele surgiu, nem de onde veio.

Figura 16 – Resultado da Pergunta 8



Fonte: A autora

4.9 Questão 9 - Conhece outros tipos de ferramenta dos ambientes informativos (Ambientes que deixam exposto informações no local de trabalho, por meio de quadros e cartazes)? Qual?

Nessa nona questão do questionário foram citadas várias ferramentas como: Redmine, GitLab, Mantis, Trello, OTRS, Planilhas compartilhadas, Webmail, Wiki, Scrumy e github. Mas o OTRS e o Redmine foram os dois mais citados. Redmine é um Software, gerenciador de projetos baseados na web e ferramenta de gerenciamento de Bugs. Ele contém calendário e gráficos de Gantt para ajudar na representação visual dos projetos e seus deadlines (prazos de entrega). Ele pode também trabalhar com múltiplos projetos. Já o OTRS, é um sistema de gerenciamento de incidentes livre e de código aberto que uma empresa, organização ou outra entidade pode usar para atribuir rótulos para a entrada de requisições e acompanhar comunicações futuras sobre elas. É um meio de gerenciar consultas recebidas, reclamações, pedidos de suporte, relatórios de defeitos e outras comunicações.

4.10 Questão 10 - Quais desses canais utilizam algum tipo de tratamento visual da informação? Relatórios, gráficos e etc?

E como na questão anterior, o Redminer e o OTRS também foram os mais citados, mas, outra ferramenta foi muito citada nessa questão, o Trello, que é bastante conhecido por ser uma ferramenta de gerenciamento de projetos em listas extremamente versátil e que pode ser ajustada de acordo com as necessidades do usuário.

Essas ferramentas oferecem vários recursos visuais, como gráficos, calendários, quadros, entre outros. O trello por exemplo, permite a criação de diversos quadros, nos quais podemos criar quantas colunas quisermos. Dentro de cada coluna é possível adicionar um ou mais "cards" (que são as tarefas propriamente ditas), contendo o conteúdo que o usuário desejar. Já o Redmine contém calendário e gráficos de Gantt para ajudar na representação visual dos projetos e seus deadlines (prazos de entrega). Ele pode também trabalhar com múltiplos projetos. O OTRS é um sistema feito para melhorar e facilitar o atendimento ao cliente aumentando a qualidade, a partir de uma interface flexível e intuitiva, com visão Kanban, painéis gerenciais, temas e campos personalizáveis, calendário, e várias outras funcionalidades.

4.11 Questão 11 - Essas informações visuais ficam disponíveis para todos os interessados do projeto (clientes, gerentes, time)?

Alguns respondentes não tinham muita certeza quando deram a resposta, mas, quase 30% das pessoas responderam que sim. Os outros 70% ficaram divididos entre apenas o time e o gerente tem acesso, e o cliente tem acesso e há casos em que nem mesmo o time tem acesso a todas as informações.

5 CONCLUSÃO

Uma prática criada no XP, os ambientes informativos hoje são adotados por muitas organizações uma vez que esta melhora a comunicação como um todo. Esse estudo de caso teve o objetivo de analisar, através de pesquisas, os ambientes informativos do NTI.

A partir desses resultados coletados, foi possível notar que o ambiente informativo do NTI age de forma positiva sobre as pessoas que trabalham fazendo uso desses ambientes, já que elas fazem questão do uso, e acham de extrema importância não apenas no NTI, mas em qualquer projeto. Porém, os ambientes informativos em si, não é um tema que tem muito espaço no meio da literatura ou da web, mesmo sendo muito utilizado e bastante difundido, não existem muitos conteúdos explorando as experiências empresarias com o uso desses espaços de trabalhos informativos, como eles surgiram, e a diferença que eles trouxeram para as organizações.

Um tema a ser abordado futuramente, seria estudar a fundo os quadros e gráficos usados pelo NTI, analisando métricas, configurações, resultados e promover possíveis melhorias para esses ambientes, tentando torna-lo mais produtivo, já que muitos por questões culturais acabaram aderindo o quadro como importante fonte de informação para suas futuras e atuais atividades.

Um grande facilitador para a realização dessa pesquisa foi a colaboração de todos os respondentes para a obtenção dos dados, sendo de imensa importância o tempo dedicado para colaborar com essas pesquisas. Além de outros trabalhos publicados em sites, revistas, periódicos, que trouxe a possibilidade de um maior entendimento sobre o tema em questão.

REFERÊNCIAS

BAKER, S. (1 de Agosto de 2005). *INFORMATIVE WORKSPACE*. Fonte: Energized Work: http://www.energizedwork.com/weblog/2005/08/informative-workspace

BECK, K. (2002). *Test-Driven Development by Example.*

BERNARDO, K. (8 de Desembro de 2014). *Kanban: Do início ao fim!* Fonte: Cultura Ágil: http://www.culturaagil.com.br/kanban-do-inicio-ao-fim/

Bernardo, K. (8 de Desembro de 2014). *Manifesto ágil, como tudo começou*. Fonte: Cultura Ágil: http://www.culturaagil.com.br/manifesto-agil-como-tudo-comecou/

BERTHOLDO, L. (2010). Adaptação do Scrum ao Modelo Incremental. p. 9. Fonte: http://www.ft.unicamp.br/liag/Gerenciamento/monografias/Monografia_ModeloIncremental_SCRUM.pdf

CAMPOS, E. L. (9 de Janeiro de 2012). *Burndown chart – Mede o progresso da sprint e dá indicativos do processo de trabalho da equipe*. Fonte: Blog Scrum Half: http://blog.myscrumhalf.com/2012/01/burndown-chart-medindo-o-progresso-de-sua-sprint-e-trazendo-indicativos-do-processo-de-trabalho-da-equipe/

COCKBURN, A. e. (2001). *Agile Software Development: The Business of Innovation.*

COSTA, E. (2004). *Modelo cascata apresentação*. Fonte: Modelo Cascata ou Clássico: http://modelocascata.blogspot.com.br/

DETILIUM. (1 de Fevereiro de 2016). *AGILE DEVELOPMENT METHODOLOGIES WITHIN THE GAME INDUSTRY – PART II – KANBAN*. Fonte: UNREAL METHODS: https://unrealmethodscom.wordpress.com/2016/02/01/agile-development-methodologies-within-the-game-industry-part-ii-kanban/

FERNANDES, J. H. (2003). Qual a prática do desenvolvimento de software. *Ciência e Cultura*, 29-33.

FILHO, B. (Janeiro de 2011). Um estudo analítico entre as. UNIVERSIDADE FEDERAL DE PERNAMBUCO.

GOMES, F. (2013). *Introdução ao FDD - Feature Driven Development*. Fonte: DEVMEDIA: http://www.devmedia.com.br/introducao-ao-fdd-feature-driven-development/27971

GUSMÃO, W. (1 de Maio de 2013). *Engenharia de Software - Modelo Cascata ou Clássico*. Fonte: Engenharia de Software: http://adsbaixarengenhariadesoftware.blogspot.com.br/2013/05/engenharia-de-software-modelo-cascata.html

HENRIQUE E FIORINO, F. e. (19 de Setembro de 2013). *O que é Kanban?* Fonte: Industria Hoje: http://www.industriahoje.com.br/o-que-e-kanban

HERYMADAPASKA. (8 de Março de 2015). *W2_HMIP_Using EVM as Standar Project Information for FEED WBS Layout Template*. Fonte: GARUDA AACE 2015: https://garudaaace2015.wordpress.com/2015/03/08/w2_hmip_using-evm-as-standar-project-information-for-feed-wbs-layout-template/

HOW, T. K. (Maio de 2011). *What is Feature Driven Development?* Fonte: Technical Know How: http://techknowhowforyou.blogspot.com.br/2011/05/what-is-feature-driven-development.html

JEFRIES, R. (fevereiro de 2009). *What is Extreme Programming.* Fonte: Extreme Programming: http://www.extremeprogramming.com

JUNIOR, N. (2013). *Desenvolvendo softwares Orientados a Objeto*. Fonte: DevMedia: http://www.devmedia.com.br/desenvolvendo-softwares-orientados-a-objeto/3853

LEONEL, S. (2007). Gestão de Produtividade : eXtreme Programming. São Paulo, Brasil: Universidade de São Paulo.

MAROTO, B. (28 de Abril de 2012). *Engenharia de Software para Concursos - Parte 3 - Métodos Formais, Interativos e Incrementais*. Fonte: Bruno Maroto: http://brunomarota.blogspot.com.br/2012/04/engenharia-de-software-para-concursos_28.html

MARTINS, R. (2013). *Kanban: 4 passos para implementar em uma equipe*. Fonte: DevMedia: http://www.devmedia.com.br/kanban-4-passos-para-implementar-em-uma-equipe/30218

MEDEIROS, H. (s.d.). *Introdução aos Processos de Software e o Modelo Incremental e Evolucionário*. Fonte: DevMedia: http://www.devmedia.com.br/introducao-aos-processos-de-software-e-o-modelo-incremental-e-evolucionario/29839

MEIER, J. (4 de Abril de 2010). *Os quatro círculos de XP (Extreme Programming)*. Fonte: Microsoft Developer: https://blogs.msdn.microsoft.com/jmeier/2010/04/04/the-four-circles-of-xp-extreme-programming/

NOVELLI, T. (14 de Janeiro de 2014). *Radiadores de Informação – Informações de Forma Ágil*. Fonte: Avanade: http://blog.avanade.com/brasil-blog/lideranca-e-projetos/radiadores-de-informacao-informacoes-de-forma-agil/

OLIVEIRA, R. (14 de novembro de 2014). *Como utilizar o Espaço de Trabalho Informativo*. Fonte: Univale: http://www.univale.com.br/unisite/mundo-j/artigos/59_EspacoTrabalho.pdf

PALUCH, M. (28 de Junho de 2012). *Xtreme Feedback Device*. Fonte: Paluch: http://www.paluch.biz/blog/57-xfd.html

PRESSMAN, R. S. (2006). *Engenharia de Software .* São Paulo: McGraw Hill/Nacional.

REIS, L. (Março de 2016). *Metodologia ágil Crystal*. Fonte: leandromtr: http://www.leandromtr.com/tecnologia-informacao/metodologia-agil-crystal/

SCHWABER, K. (2004). *Agile Project Management with Scrum.* Microsoft Press.

SOARES, M. d. (2004). Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software. *Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software.*

SOMMERVILLE, I. (2008). *Engenharia de Software.* Rio de Janeiro: Prentice Hall.

TELES, M. (2004). *Extreme Programming.* NOVATEC.

TELES, M. (02 de 10 de 2006). Fonte: DesenvolvimentoAgil.com.br: http://www.desenvolvimentoagil.com.br/xp/praticas/trabalho_energizado

UFPE. (2009). *NTI*. Fonte: Universidade Federal de Pernanbuco: https://www.ufpe.br/nti/index.php?option=com_content&view=article&id=88&Itemid=71

VALE, A. (4 de Abril de 2007). *Ambiente Informativo*. Fonte: Just in Time for Knowledge Work: http://alissonvale.com/englishblog/post/2007/04/04/Ambiente-Informativo.aspx

WATERS, K. (7 de Fevereiro de 2011). *Track Your Agile Projects with a Project Burndown Chart*. Fonte: All About Agile: http://www.allaboutagile.com/track-your-agile-projects-with-a-project-burndown-chart/

APÊNDICE – A

Estudo de Campo - NTI

Estou fazendo tcc e gostaria da ajuda de vocês para recolher alguns dados.

Você sabe o nome do quadro com os post-its que é usado no seu ambiente de trabalho? *

- ( ) Sim
- ( ) Não

O que você acha do quadro Kanban? *

Texto de resposta longa

Em qual tipo de projeto você acha importante a presença do Kanban? *

Texto de resposta longa

Prefere: *

- ( ) Físico
- ( ) Virtual

Você sabe para que serve cada raia/etapa (Mesmo aquelas que você não usa)? *

- ( ) Sim
- ( ) Não

Acontecem reuniões em torno dele? *

- ( ) Sim
- ( ) Não

Você sabe como o Kanban surgiu ? *

- ◯ Sim
- ◯ Não

Conhece outros tipos de ferramenta dos ambientes informativos (Ambientes que deixam exposto informações no local de trabalho, por meio de quadros e cartazes) ? Qual? *

Texto de resposta longa

Quais desses canais utilizam algum tipo de tratamento visual da informação? Relatórios, gráficos e etc? *

Texto de resposta longa

Essas informações visuais ficam disponíveis para todos os interessados do projeto (clientes, gerentes, time)? *

Texto de resposta longa