



**ULBRA**

**CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS**

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"  
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

Gabriela Fachine Brito

PROPOSTA DE UTILIZAÇÃO DO *KANBAN* E *LEAN* PARA MELHORIA  
DO GERENCIAMENTO DE *SOFTWARE*

Palmas - TO  
2013

Gabriela Fachine Brito

PROPOSTA DE UTILIZAÇÃO DO *KANBAN* E *LEAN* PARA MELHORIA  
DO GERENCIAMENTO DE *SOFTWARE*

Trabalho de Conclusão de Curso (TCC) elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof<sup>ª</sup>. M.Sc. Cristina Filipakis.

Palmas - TO  
2013

Gabriela Fachine Brito

PROPOSTA DE UTILIZAÇÃO DO *KANBAN* E *LEAN* PARA MELHORIA  
DO GERENCIAMENTO DE *SOFTWARE*

Trabalho de Conclusão de Curso (TCC) elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof<sup>ª</sup>. M.Sc. Cristina Filipakis.

Aprovada em 05 de Dezembro de 2013.

BANCA EXAMINADORA

---

Prof. M.Sc. Edeilson Milhomem  
Centro Universitário Luterano de Palmas

---

Prof. M.Sc. Madianita Bogo  
Centro Universitário Luterano de Palmas

---

Prof. M.Sc. Cristina Filipakis  
Centro Universitário Luterano de Palmas

Palmas - TO  
2013

## AGRADECIMENTOS

Agradeço primeiramente a Deus, que foi o responsável por tudo existir e pela capacidade me dada para suportar todas as dificuldades. Foi Ele quem me guiou durante todos os dias de luta desde os primeiros dias de aula, que não foram nada fáceis, foi Ele o responsável pela minha coragem em continuar durante os momentos de desespero, onde o cansaço era maior do que a força de vontade. Foi Ele quem me confortou nos momentos difíceis, como: as viagens diárias de 120km para chegar na faculdade, as noites de provas, noites de defesas, noites de entregas e/ou apresentação de trabalhos e outras inúmeras noites. Ele quem sempre me mostrou o melhor caminho a seguir, iluminando os meus passos e me ajudando a cumprir as missões me oferecidas.

Agradeço aos meus pais, que sempre me apoiaram e que sempre foram os maiores incentivadores desta vitória. Primeiro a minha mamãe (Lilian), minha melhor amiga, sempre me transmitindo calma e forças pra continuar. Segundo o meu pai (Pascual), meu herói, que sempre me mostrou o caminho da honestidade e do amor ao próximo. Agradeço a Deus pela chance de ser filha (única) do melhor casal desse mundo, que me ajudaram a seguir até aqui. Sei que não foi fácil me manter em uma faculdade particular, pagar as despesas de ônibus entre Paraíso X Palmas todos os dias durante todos estes anos, mas a vontade de vencer foi maior e conseguiu superar todas as dificuldades. Agradeço aos meus pais pela liberdade de tomar algumas decisões e de me apoiarem independentemente das minhas escolhas, é por eles que vivo e esta vitória é para eles. Agradeço minha mamãe e meu papai por tudo que eles me deram, agradeço ao meu namorado (Thalisson) por toda a paciência, por todas as piadinhas para me alegrar, por todo incentivo durante o curso e por todos os planos investidos... Agradeço a minha sogra (Dona Rosilmar) por ter me colocado em suas orações para que eu pudesse atingir este objetivo. Agradeço a minha madrinha (Tia Andréa) por todo apoio e incentivo, pelo abraço apertado e pelas orações durante o curso, agradeço ao meu primo André pelas conversas sobre os nossos cursos, pelos abraços apertados e pelo incentivo me dado. Agradeço a vóvó mais linda deste mundo (Vóvó Jussara), que sempre orou para que eu conseguisse atingir meus objetivos, quem sempre me incentiva nas minhas decisões e que mesmo estando longe se faz presente, seja por mensagens no *Facebook* ou breves ligações e SMS. Muito obrigada, amo vocês.

Agora os agradecimentos a quem me ajudou dentro da faculdade, desde uma caneta emprestada até um trecho de código. Agradeço ao Diego (amigo de Paraíso) por toda a

paciência e auxílio em todo o curso, ao Jan Tarik por toda alegria contagiante, ao Mário e Willian por sempre me ajudarem nas disciplinas sempre quando precisei, ao Fábio pelo companheirismo e gentileza, a Cristiane pelas risadas, tardes de estudos e amizade verdadeira, a Cleydiane que mesmo de longe continua me ajudando muito, ao meu primo Douglas pelos trabalhos que me ajudou e pelas primeiras conversas sobre o curso e tardes de introdução aos algoritmos, agradeço a Katia pelo companheirismo em algumas disciplinas e pela amizade verdadeira. Agradeço a todos os meninos do curso que sempre me ajudaram quando precisei, seja para responder alguma dúvida minha ou para me ajudar a andar em Palmas. Muito obrigada. Agradeço a todos os amigos de Paraíso, que sempre me incentivaram e apoiaram o estudo em Palmas, que mesmo de longe se faziam presente.

Agradeço a professora Cristina, minha mãe dentro do curso, pela orientação, pela amizade, pelas conversas, pelos conselhos, pelas matérias mais interessantes do curso, pelo exemplo de profissional, pelo apoio e ajuda quando precisei e vários outros fatores que nem cabem aqui. Muito obrigada Cris. Agradeço ao professor Jackson, por toda a paciência comigo durante as aulas de programação, pelas conversas, pela fé nos alunos e pelas caronas para Paraíso. Muito obrigada Jack. Agradeço ao professor Fernando por tudo que foi ensinado, pelos conselhos ditos em salas de aula. Muito obrigada Fernando. Agradeço ao professor Fabiano por fazer das aulas mais atrativas, sempre com piadas educativas. Muito obrigada Fabiano. Agradeço ao professor Edeilson por todas as conversas, danças no corredor e piadas. Muito obrigada Ed. Agradeço a professora Madianita, minha segunda mãe dentro do curso, por todas as conversas, conselhos, paciência e orientação. Muito obrigada Madia. Por fim, agradeço a professora e tia Parcilene, que sempre me ajudou seja de forma direta ou indireta, de algum modo, minha vida acadêmica, agradeço os *emails* breves, conselhos, caronas para Paraíso e por ter me recebido em sua casa nestes últimos meses. Muito obrigada Tia Parci. Vocês serão inesquecíveis na minha vida profissional e pessoal, todos foram exemplos de vida para mim.

Enfim, agradeço a todos, que me ajudaram diretamente ou indiretamente no desenvolvimento do curso. Foram cinco anos (2009/1 à 2013/2) de lutas e glórias. Isto é um sonho realizado que jamais esquecerei. Muito obrigada.

## RESUMO

O presente trabalho tem como objetivo propor a utilização da ferramenta *Kanban* juntamente com a metodologia *Lean Software Development* para melhor monitoramento de atividades, estas no âmbito de gerenciamento de *software*. Isto se faz necessário devido aos atrasos nas entregas, mal divisão de tarefas e responsabilidades, e, mal direcionamento de recursos. A ferramenta *Kanban* foca no controle visual, onde todas as equipes acompanham todas as atividades do projeto por meio de um quadro *Kanban*. Já a metodologia *Lean* possui várias abordagens e seu foco principal é a eliminação de desperdícios e conseqüente agregação de valor para o produto final. Com a junção destas duas práticas foi possível definir princípios, papéis, processos, etapas, artefatos e melhorias contínuas no *software* desenvolvido para uma maior satisfação tanto para o desenvolvedor quanto para o cliente. Com isso, a proposta abrange o desenvolvimento de processos para melhor gerenciamento de projetos de *software*.

**PALAVRAS-CHAVE:** *Lean Software Development*; *Kanban*; monitoramento; gerenciamento; atividades; melhoria contínua.

## LISTA DE FIGURAS

Figura 1 - Fases do Rup (Fonte: BASSI, 2008).....	6
Figura 2 - Ciclo de vida do RUP (Fonte: SEGET, 2008) .....	7
Figura 3 - O modelo em Cascata (Fonte: SOMMERVILLE, 2010).....	8
Figura 4 - Desenvolvimento iterativo e em espiral (Fonte: DEVMEDIA, 2010) .....	9
Figura 5 - Ciclo do Scrum (Fonte: BARDUSCO, 2010 ) .....	10
Figura 6 - Fontes de Desperdícios Lean.....	13
Figura 7 - Relação de Valor, Custo e Desperdício (Fonte: FORTES, 2010) .....	15
Figura 8 - Mapa mental básico Lean.....	21
Figura 9 - Fontes de Desperdícios Lean de Software .....	23
Figura 10 - Lean e Métodos Ágeis .....	26
Figura 11 - Fluxo Lean (Fonte: UNIVERSITY OF ST ANDREWS, 2011 - Adaptado) .....	28
Figura 12 - Solicitação .....	29
Figura 13 - Escopo .....	30
Figura 14 - Planejamento .....	31
Figura 15- Treinamento.....	32
Figura 16 - Redesign.....	32
Figura 17 - Implementação.....	33
Figura 18 - Revisão.....	34
Figura 19 - Feedback .....	35
Figura 20 - Exemplo da utilização do quadro Kanban (Fonte: PUPO, 2011).....	37
Figura 21 - Exemplo básico de quadro Kanban (Fonte: KNIBERG e SKARIN, 2009) .....	41
Figura 22 - Mau direcionamento da limitação do trabalho em progresso (Fonte: KNIBERG e SKARIN, 2009).....	42

Figura 23 - Aumentando a limitação do trabalho em progresso. (Fonte: KNIBERG e SKARIN, 2009).....	43
Figura 24 - Problemas com a limitação do trabalho em progresso (Fonte: KNIBERG e SKARIN, 2009).....	43
Figura 25 - Identificação do problema (Fonte: KNIBERG e SKARIN, 2009).....	44
Figura 26 - Limite do trabalho em progresso alterado (Fonte: KNIBERG e SKARIN, 2009)	45
Figura 27 - Equipes e Atividades no Kanban (Fonte: KNIBERG e SKARIN, 2009).....	45
Figura 28 - Sistema kanban e Método Kanban .....	50
Figura 29 - Kaizen .....	54
Figura 30 - Processo Just-in-time (Fonte: SILVEIRA, 2012).....	56
Figura 31- Metodologia do Projeto.....	60
Figura 32 - Princípios da Metodologia .....	63
Figura 33 - Componentes essenciais.....	64
Figura 34 - Visão geral das Abordagens utilizadas .....	66
Figura 35 - Etapas e Papéis .....	68
Figura 36 - Juntando os processos: <i>Lean</i> e <i>Kanban</i> .....	69
Figura 37 - Papéis da Metodologia.....	70
Figura 38 - Tipos de Cliente.....	71
Figura 39 - Legenda dos Papéis.....	72
Figura 40 - Início do projeto.....	73
Figura 41 - Escopo do projeto .....	75
Figura 42 - Planejamento .....	77
Figura 43 - Treinamento Lean.....	80
Figura 44 - Kanban .....	81
Figura 45 - Implementação.....	83



Figura 46 - Pontos de Melhorias Contínuas .....	85
Figura 47 - Redesign.....	86
Figura 48 - Fim do projeto .....	88
Figura 49 - Exemplo de Gráfico Visual (Fonte: DENK, 2005) .....	90
Figura 50 - Exemplos de pautas para as Reuniões .....	91
Figura 51 - Microsoft Test Manager (Fonte: MICROSOFT, 2012).....	92
Figura 52 - Rounds da Metodologia .....	94
Figura 53 - Coluna de atividades a serem feitas (Fonte: NICOLAY, 2013).....	95
Figura 54 - Coluna de atividades em andamento (Fonte: NICOLAY, 2013).....	96
Figura 55 - Coluna de atividades concluídas (Fonte: NICOLAY, 2013) .....	97
Figura 56 - Lista de Membros e Papéis.....	98
Figura 57 - Plano de Reunião .....	99

**LISTA DE TABELAS**

Tabela 1 - Evolução da terminologia <i>Lean</i> . (Fonte: Fortes, 2010).....	16
Tabela 2 - Desenvolvimento e Excelência <i>Lean</i> .....	17
Tabela 3 - Princípios da Produção Enxuta e sua adaptação para Serviços (Fonte: PASA, 2004 - adaptado) .....	19
Tabela 4 - Características específicas do <i>Kanban</i> .....	47
Tabela 5 - <i>Lean</i> e <i>Kanban</i> .....	51
Tabela 6 - Correlação e disparidade entre as metodologias .....	52
Tabela 7 - <i>Kanban</i> e <i>Just-in-time</i> .....	57

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>1</b>
<b>2. REFERENCIAL TEÓRICO .....</b>	<b>4</b>
<b>2.1. Gerência de Projetos .....</b>	<b>4</b>
<b>2.2. Metodologias de Gerenciamento de Projetos .....</b>	<b>5</b>
<b>2.2.1. Desenvolvimento de <i>Software</i> Dirigido a Planos .....</b>	<b>6</b>
<b>2.2.1.1. <i>Rational Unified Process</i> (RUP) .....</b>	<b>6</b>
<b>2.2.1.2. Modelo em Cascata .....</b>	<b>7</b>
<b>2.2.2. Desenvolvimento Ágil de <i>Software</i> .....</b>	<b>8</b>
<b>2.2.2.1. Scrum.....</b>	<b>10</b>
<b>2.3. Metodologia <i>Lean</i> .....</b>	<b>11</b>
<b>2.3.1. Abordagens <i>Lean</i>.....</b>	<b>12</b>
<b>2.3.1.1. Fontes de Desperdício – Produção <i>Lean</i>.....</b>	<b>13</b>
<b>2.3.2. Desenvolvimento <i>Lean</i> de <i>Software</i> – <i>Lean Software Development</i>.....</b>	<b>22</b>
<b>2.3.3. Fluxo da metodologia <i>Lean</i> .....</b>	<b>27</b>
<b>2.4. Ferramentas de Produção.....</b>	<b>35</b>
<b>2.4.1. <i>Kanban</i>.....</b>	<b>36</b>
<b>2.4.2. <i>Kaizen</i>.....</b>	<b>53</b>
<b>2.4.3. <i>Just-in-time</i> .....</b>	<b>55</b>
<b>3. MATERIAIS E MÉTODOS .....</b>	<b>59</b>
<b>3.1. Local e Período.....</b>	<b>59</b>
<b>3.2. Materiais.....</b>	<b>59</b>
<b>3.3. Métodos .....</b>	<b>59</b>
<b>4. RESULTADOS E DISCUSSÃO .....</b>	<b>62</b>
<b>4.1. Os Princípios da Metodologia.....</b>	<b>62</b>
<b>4.2. Etapas e Processos da Metodologia .....</b>	<b>67</b>
<b>4.3. Papéis da Metodologia .....</b>	<b>69</b>
<b>4.4. Etapas e Processos da Metodologia .....</b>	<b>73</b>
<b>4.5. Artefatos da Metodologia .....</b>	<b>89</b>
<b>5. CONSIDERAÇÕES FINAIS.....</b>	<b>100</b>
<b>5.1. Trabalhos Futuros .....</b>	<b>101</b>
<b>6. REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>102</b>

## 1. INTRODUÇÃO

Com o crescente progresso na globalização de tecnologias de comunicação e informação, houve uma necessidade maior das organizações se coordenarem de forma com que seus processos, recursos e custos sejam melhores aplicados e direcionados, para que seus produtos e serviços sejam produzidos de maneira satisfatória e, conseqüentemente, novos clientes sejam conquistados.

Frequentemente, por meio de diagnósticos é possível observar que as organizações e empresas sofrem inúmeros problemas, e estes problemas muitas vezes são causados pela má gerência ou a falta dela. A falta de gerenciamento faz com que os objetivos e metas das empresas se distorçam do que é de fato desejado, ficando cada vez mais difícil de serem alcançados de acordo com a realidade da empresa.

Muitas destas organizações fracassam por causa de uma gerência ineficiente ou pela falta dela, isto foi evidenciado pelo relatório do *Standish (CHAOS REPORT, 2011)*, através da análise e pesquisas em projetos de TI, que sempre citam a falta de utilização de técnicas de gerenciamento de projeto, a realização de estimativas não confiáveis e falta de acompanhamento e controle dos processos de *software*. Para minimizar estes índices, segundo Leite (2010, *online*), surgiram as metodologias de desenvolvimento de *software*, que possuem como objetivo auxiliar no processo de produção do *software*. A pesquisa mais recente do *Standish (2011)* mostra, através de análises nos projetos de TI, que o índice de sucesso dos projetos pode estar diretamente ligado com a utilização de processos ágeis em seu desenvolvimento.

Da mesma forma, o *Standish Group (2001)* apresentou por meio de seu relatório *Chaos* e análises de projetos, que tem como foco o desenvolvimento de *software* comercial, que “31% dos *softwares* são cancelados antes de serem concluídos; 53% dos projetos de *software* excedem mais do que a sua estimativa de custo; e, somente 9% dos projetos, em grandes empresas, foram entregues no tempo e orçamento estabelecidos”.

No cenário nacional, o SEBRAE realiza frequentemente pesquisas sobre os fatores condicionantes e taxa de mortalidade de empresas no Brasil. A má gestão financeira é um dos principais motivos do fechamento de micro e pequenas empresas brasileiras. Segundo o artigo científico “Pequenas Empresas Fecham por Falta de Gerenciamento” (SEBRAE, 2011), das 11 mil indústrias existentes no cenário brasileiro, 95% são micro e pequenas empresas. E

destes 95%, 90% decretam falência por problemas na administração, e não por falta de mercado. O SEBRAE (2011) afirma que “um dos fatores que contribuem para a mortalidade das empresas é a falta de gerenciamento dos recursos com o exato controle de estoque, de compras, financeiro e contabilidade”.

Com o objetivo de auxiliar no processo de adaptação da engenharia de *software*, conforme Poppendienck (2003), foram desenvolvidas as metodologias *Lean* e *Kanban*, a primeira aplicada nas indústrias automobilísticas, e a segunda aplicada de forma a obter um melhor controle visual dos processos. Direcionando-as para computação e desenvolvimento de sistemas, surgiu o conceito *Lean Software Development* (LSD), que é a aplicação da metodologia *Lean* em desenvolvimento de *software*. Um dos criadores do conceito *Lean Software Development*, Poppendienck (2003, *online*), apresenta princípios e práticas da melhor forma de desenvolvimento dos projetos de *software*, focando na eliminação dos desperdícios. O *Kanban* visa o trabalho em progresso e, conforme Santos (2011, p. 13), é um método ágil de desenvolvimento de *software* e, sempre que necessário, as metodologias recorrem ao *Kanban* para colocar em prática boa parte de seus processos de monitoramento. O *Kanban* foca na visualização em progresso das atividades, gerando uma melhor apresentação e controle das mesmas.

Existem inúmeras ferramentas que as metodologias utilizam para garantir uma implantação eficaz e sólida na organização para completar e auxiliar o *Kanban*. As ferramentas *Kanban*, *Kaizen* e *Just-in-time* serão aplicadas na metodologia *Lean Software Development* com o intuito de completá-la. Diante de todas as metodologias e técnicas destinadas a auxiliar e gerenciar projetos existentes, neste trabalho, foram selecionadas as metodologias RUP e SCRUM. Esta seleção foi realizada para elaborar uma tabela que ilustra as disparidades e vantagens que a metodologia *Lean Software Development* e a técnica *Kanban* possuem diante das demais supracitadas. A relação do RUP e SCRUM com a proposta do trabalho é apresentar o que a metodologia *Lean Software Development* e o *Kanban* possuem de diferenças em relação às metodologias tradicionais. Isto justificará o motivo dos quais estas foram escolhidas para o desenvolvimento da proposta.

Este trabalho, portanto, apresenta uma proposta para amenizar, através da transparência e visibilidade no monitoramento de atividades, os problemas relacionados ao gerenciamento de projetos e controle de atividades. Esta proposta foca no envolvimento e utilização da técnica *Kanban* em conjunto com a metodologia *Lean Software Development* para melhoria do gerenciamento de *software*. E como problema de pesquisa, foi possível observar, através do estudo aprofundado dos temas, se a relação entre *Kanban* e *Lean* pôde

tornar a etapa de monitoramento de atividades mais eficaz. Esse estudo englobou os fluxos e atividades existentes em ambas metodologias para possibilitar a inserção de novos passos e processos. Isto se fez viável devido a adequação realizada em ambas para que fosse garantido um novo fluxo de processos, que resolveu o problema proposto.

A metodologia *Lean Software Development* se difere das demais por ser um modelo baseado na manufatura das indústrias, mais precisamente da Toyota. Segundo Anderson (2010), um dos idealizadores do *Lean* para desenvolvimento de *software*, “um processo precisava ser adaptado para cada situação, e que para fazer isso, era necessária uma liderança ativa em cada equipe”.

A metodologia *Lean Software Development* busca aumentar a satisfação dos clientes através de um melhor aproveitamento de recursos, evitando os desperdícios. Dessa forma é garantida uma maior satisfação por parte de seus clientes, com uma aplicação menor de valores e aproveitando ao máximo os recursos disponíveis, tendo como consequência, um melhor custo benefício para o cliente.

Outra ferramenta comumente utilizada para auxiliar o gerenciamento dos projetos é a técnica *Kanban*, que utiliza de cartões e um quadro branco para permitir agilizar a entrega de projetos e a visualização sobre o que acontece como um todo. É uma ferramenta com o foco no controle visual, permitindo visualizar o projeto de forma mais clara e sem acúmulo de estoque.

Diante deste contexto, é possível observar a deficiência nas gestões de projetos, muitas vezes por aplicação errada de metodologias existentes. Com isso, o presente trabalho objetiva criar uma proposta de metodologia de desenvolvimento e planejamento para área de computação, baseada nas técnicas *Kanban* e metodologia *Lean Software Development*.

## 2. REFERENCIAL TEÓRICO

Esta seção abordará os conceitos relacionados ao trabalho, de modo a contextualizá-los em uma revisão de literatura de acordo com as referências bibliográficas encontradas sobre os assuntos: Gerência de Projetos; Metodologias de Gerenciamento de Projetos; Metodologia *Lean* e *Kanban*.

### 2.1. Gerência de Projetos

Um projeto segundo o *Project Management Institute* (2011), “é um empreendimento temporário com o objetivo de criar um produto ou serviço único”. Os projetos poderão ser desenvolvidos em todos os níveis hierárquicos das empresas, podendo ter início, meio e fim bem planejados. Para que o projeto seja vantajoso e benéfico é importante ressaltar que seja diferente dos demais produtos e serviços semelhantes.

Conforme a ONU (1984), “Projeto é um empreendimento planejado que consiste num conjunto de atividades inter-relacionadas e coordenadas, com o fim de alcançar objetivos específicos dentro dos limites de um orçamento e de um período de tempo dados”. A probabilidade de projetos se tornarem bem sucedidos ao seu final aumenta quando aplicada a Gerência de Projetos, que possui como base a utilização de cronogramas, equipes, coordenadores, análises e vários outros critérios que colaboram positivamente com o desenvolvimento de projetos.

Conforme o artigo *Gestão de Projetos de Software* (2010):

Gestão de Projetos é um conjunto de práticas que serve de guia a um grupo para trabalhar de maneira produtiva. Ela compreende métodos e ferramentas que organizam as tarefas, identificam sua sequência de execução e dependências existentes, apoia a alocação de recursos e tempo, além de permitir o rastreamento da execução das atividades e medição do progresso relativo ao que foi definido no plano do projeto.

Neste contexto, a Gestão de Projetos possui algumas características, como: envolver a comunicação dos progressos e resultados às equipes de desenvolvimento do projeto; análise e controle de tarefas, prazos, custos e recursos envolvidos no projeto.

Turner (1994) possui uma concepção significativa sobre a Gestão de Projetos, onde ele se refere a ela como:

Um processo através do qual um projeto é levado a uma conclusão. Tem três dimensões: objetivos (âmbito, organização, qualidade, custo, tempo); processo de gestão (planejar, organizar, implementar, controlar); e, níveis (integrativo, estratégico, tático).

A importância da Gerência de Projetos é observada, conforme Schlichter (2001), como “uma disciplina articulada em nível organizacional, onde a competência é fundamental para a realização das estratégias corporativas, através da contínua entrega de projetos bem sucedidos”. Além disto, o gerenciamento de projetos engloba o conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de cumprir os requisitos pré-estabelecidos (PMBOK, 2010).

O Gerenciamento de Projeto de *Software*, conforme afirmado por Dinsmore (1992), “é a capacidade de equilibrar dedicação e implementação de tecnologia, ao mesmo tempo liderando e inspirando os membros da equipe”. Ou seja, é aplicar as técnicas de gerenciamento de projeto que englobam habilidades e conhecimentos para aplicar à informática para atingir às solicitações do projeto. A Gerência de Projetos de *Software* envolve, dentre outros, o planejamento e acompanhamento das pessoas envolvidas no projeto, o produto sendo desenvolvido e o processo para evoluir o *software* para uma implementação concreta e operacional (PMBOK, 2010).

Conforme abordado nesta seção, a Gerência de Projetos é de suma importância no desenvolvimento de projetos e para isso existem as metodologias de Gerenciamento de Projetos, que servem como guias aos gestores, possuindo diversas características e possibilitando a escolha de acordo com as estratégias de trabalho. Estas metodologias serão abordadas na próxima seção.

## **2.2. Metodologias de Gerenciamento de Projetos**

Existem diversas divisões entre as metodologias que possibilitam o desenvolvimento dos projetos de *software*. A presente seção propõe a explanação de três grupos, são eles: Desenvolvimento de *Software* Dirigido a Planos; Desenvolvimento Ágil de *Software*; e, Desenvolvimento *Lean* de *Software*.



### 2.2.1. Desenvolvimento de *Software* Dirigido a Planos

Segundo Peterson (2010), “o desenvolvimento de *software* dirigido a planos está focado em planejar tudo desde o início do projeto”. Este grupo de desenvolvimento foca na produção de artefatos no planejamento inicial do projeto, e estes artefatos acompanham todo o desenvolvimento do projeto até que seja concluído.

Esta abordagem é dividida por fases e, no final de cada fase, são produzidos novos artefatos para comprovação de que a fase foi concluída para que possa ser iniciada a próxima fase (BOSNIC, 2012). Este grupo de metodologias possui como exemplo o RUP e o Ciclo de Vida Clássico, que apesar de serem adaptáveis foram desenvolvidos primeiramente para grandes projetos. Ambos serão abordados nas próximas seções.

#### 2.2.1.1. *Rational Unified Process* (RUP)

Um exemplo de processo dirigido a planos é o *Rational Unified Process*, RUP. Este processo contém as atividades de engenharia de *software* as quais são executadas durante todo o ciclo de vida do projeto. O RUP possui quatro fases: concepção, elaboração, construção e transição.

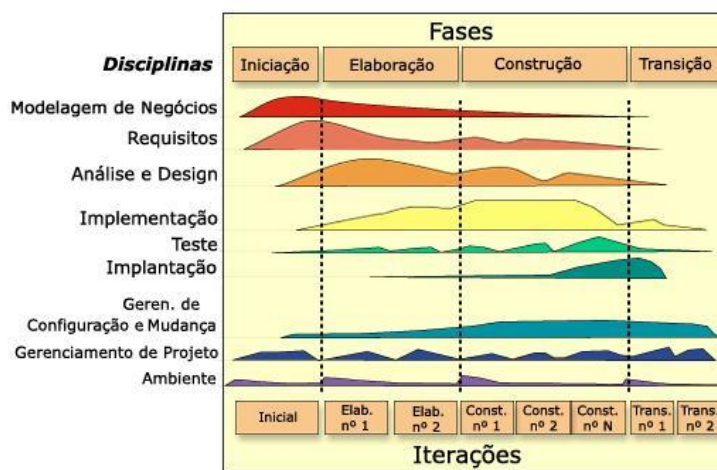


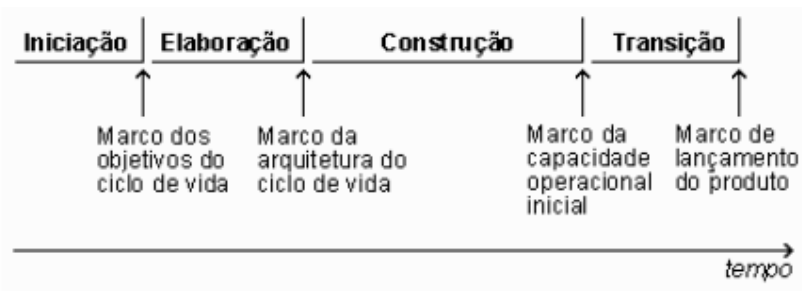
Figura 1 - Fases do RUP (Fonte: BASSI, 2008)

Conforme exibido pela Figura 1, as disciplinas da engenharia de *software* participam de quase todas as fases, a única diferenciação existente é a intensidade entre elas.

De acordo com Kroll e Kruchten (2003), o RUP possui três definições, são elas: O RUP é uma maneira de desenvolvimento de *software* que é iterativa, centrada à arquitetura e guiada por casos de uso; O RUP é um processo de engenharia de *software* bem definido e bem estruturado; e, O RUP é também um produto de processo que oferece estrutura customizável para a engenharia de *software*.

Uma vantagem considerável do RUP é que ele é bastante flexível, podendo ser aplicado a projetos de pequeno, médio e grande porte. O RUP possui princípios, que segundo (KRUTCHEN, 2003), “acompanham as premissas referentes à garantia da qualidade do processo visando alcançar a garantia da qualidade do produto de *software* a ser desenvolvido”. O RUP possui alguns elementos como papéis, atividades, artefatos, fluxos de trabalho e disciplinas.

Como exibido na Figura 2, o Ciclo de Vida de um projeto RUP possui quatro fases: Iniciação, Elaboração, Construção e Transição.

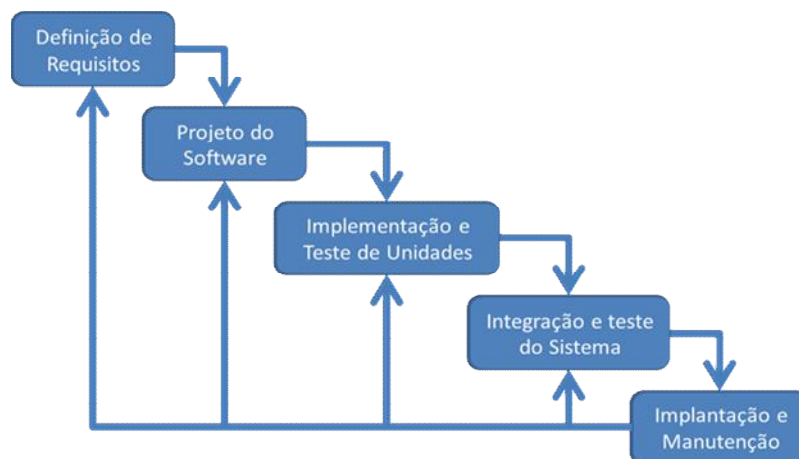


**Figura 2 - Ciclo de vida do RUP (Fonte: SEGET, 2008)**

Sempre que uma fase acaba considera-se um marco (Figura 2) e cada fase é constatada como um intervalo de tempo entre dois marcos principais. A fase de Iniciação foca nos riscos relacionados com o caso de negócio; A fase de elaboração foca nos riscos técnicos e arquiteturais; A fase de Construção foca nos riscos lógicos e boa parte do trabalho é realizado; Por fim, a fase de transição foca nos riscos associados a logística de distribuição do produto para os usuários. O RUP possibilita uma maior qualidade do *software* e um maior controle sobre o desenvolvimento dos projetos a respeito de custos, prazos e níveis.

#### **2.2.1.2. Modelo em Cascata**

Outro exemplo de processo dirigido a planos é o modelo em cascata (SOMMERVILLE, 2011). Este processo é executado de forma sequencial, seguindo um conjunto de passos das diferentes etapas do desenvolvimento de *software*. A Figura 3 ilustra o modelo em cascata e suas fases.



**Figura 3 - O modelo em Cascata (Fonte: SOMMERVILLE, 2010)**

Segundo Bosnic (2012), o principal objetivo desta abordagem é “prover uma estrutura para execução do processo de desenvolvimento de *software*”. Presman (2004) define ainda que em cada uma destas fases seja realizado um conjunto predefinido de tarefas. Estas tarefas produzem os artefatos que liberarão o início do desenvolvimento da fase seguinte.

Conforme Yourdon (1990), todo projeto desenvolvido segundo o Ciclo de vida Clássico é executado mediante algum método de análise de sistemas, projeto e implementação. Assim, o número de fases varia de organização para organização.

O Ciclo de vida Clássico exige que o usuário declare explicitamente todas as suas prioridades, mas nem sempre isso é perceptível, dificultando o processo, isso às vezes impede que o projeto siga o fluxo sequencial que o modelo propõe (PRESSMAN, 1995).

Este ciclo colabora com um melhor controle de qualidade e abrange atividades, ciclos e é comumente usado em sistemas transacionais. As metodologias de ciclo de vida clássico são as mais antigas e continuam sendo as mais utilizadas no desenvolvimento de *software* por conta de sua estrutura significativa nos projetos.

### **2.2.2. Desenvolvimento Ágil de *Software***

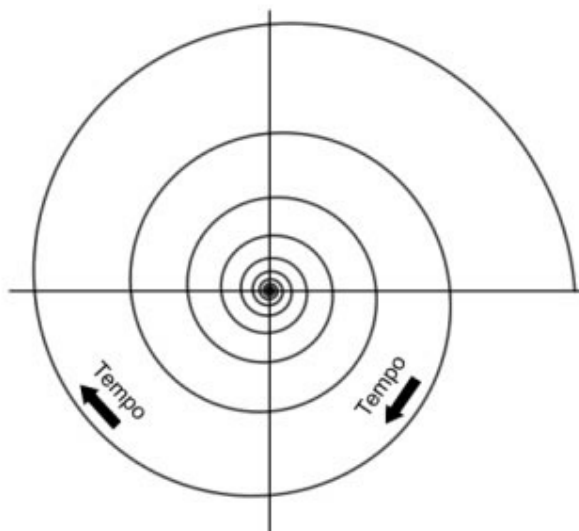
Esta abordagem surgiu a partir de um manifesto, que tinha como objetivo satisfazer o cliente entregando rapidamente e com frequência, sistemas com algum valor. Desenvolvimento ágil, ou simplesmente agilismo, é um rótulo genérico para todos os métodos de desenvolvimento de *software* baseados no manifesto ágil (BECK, 2001).

Para Cohn (2006), um plano ágil:

É aquele que nós não só iremos, como estamos ansiosos para mudar. Nós não queremos mudar o plano apenas pela mudança, nós queremos mudar porque a mudança significa que nós aprendemos algo ou evitamos um erro.

O termo desenvolvimento ágil identifica metodologias de desenvolvimento que adotam os princípios do manifesto ágil. Estes princípios são (TELES, 2004): “Indivíduos e interação entre eles mais que processos e ferramentas; *Software* em funcionamento mais que documentação abrangente; Colaboração com o cliente mais que negociação de contratos; Responder a mudanças mais que seguir um plano”.

A Figura 4 ilustra o processo iterativo e em espiral, adotado pelos projetos no desenvolvimento ágil.



**Figura 4 - Desenvolvimento iterativo e em espiral (Fonte: DEVMEDIA, 2010)**

Neste processo (Figura 4), as fases são executadas diversas vezes ao longo do projeto, produzindo ciclos curtos que se repetem ao longo de todo o desenvolvimento. E ao final de cada ciclo, têm-se um *software* funcional, testado e aprovado. Estes ciclos são denominados iterações e crescem de acordo com as funcionalidades a cada repetição que é executada (TELES, 2004).

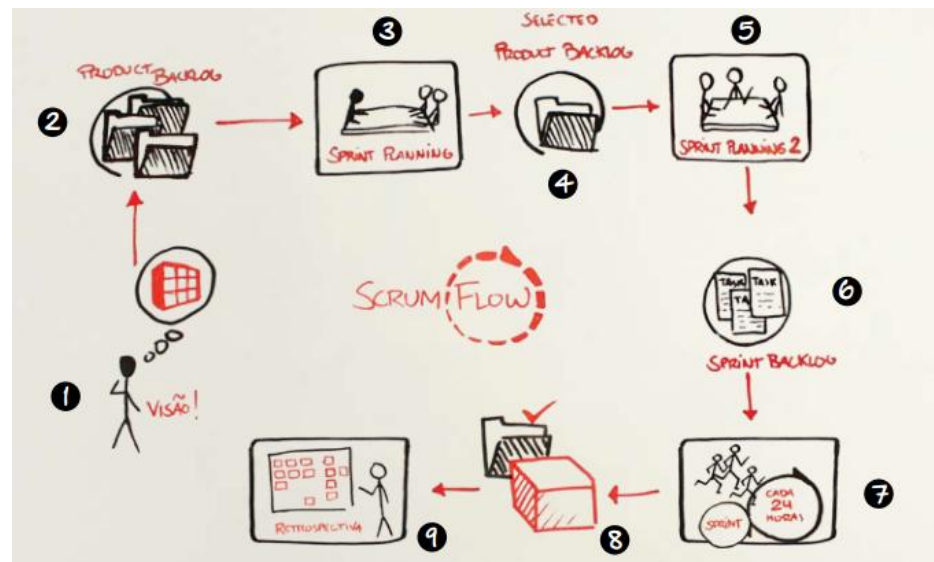
Este método permite que ao longo de todo o ciclo de vida podem ser fornecidas evidências concretas para os envolvidos sobre o estado do projeto. E, além disso, o *feedback* é bem mais presente no sistema, de forma que os requisitos possam ser bem mais facilmente

encontrados e os problemas identificados são colocados em evidência, contribuindo para reagir da forma mais adequada nos projetos.

### 2.2.2.1. Scrum

O Scrum é uma metodologia ágil para gestão e planejamento de projetos de *software*. “É fundamentado na teoria de controle de processos empíricos, emprega uma abordagem iterativa e incremental para otimizar a previsibilidade e controlar riscos (SCHWABER, 2009).”

Esta metodologia tem como conteúdo o conjunto de Times Scrum (projetados para otimizar flexibilidade e produtividade); Time-Boxes (cria regularidade) e Regras (fazem o elo entre os time-boxes e os artefatos do Scrum).



**Figura 5 - Ciclo do Scrum (Fonte: BARDUSCO, 2010)**

O ciclo Scrum (Figura 5) começa com o *Product Owner* definindo uma visão (1) de produto compartilhada com o time. Essa visão é transformada em *Product Backlog* (2), que contém uma lista de requisitos de todos os entregáveis para que aquele produto faça sentido. Essa lista deve estar sempre priorizada por valor de negócio. Requisitos podem ser adicionados ou removidos a qualquer tempo, assim como a prioridade também pode mudar. Ou seja, o *Backlog* precisa ser continuamente mantido pelo *Product Owner*, visando maximizar o retorno sobre o investimento.

É feito o *Sprint Planning* 1 (3), onde será definido “o quê” será feito durante o *Sprint*. Essa é a fase de “documentação” do projeto. O *Selected Product Backlog* (4), que é o

resultado do *Sprint Planning 1*, e define a quantidade de trabalho com a qual o time se comprometeu a entregar naquele *Sprint*. Ele se mantém inalterado durante todo o *Sprint*. O *Sprint Planning 2* (5) é a hora de definir “como” a solução será implementada. É a fase de “modelagem” do projeto. Nela, o time deve identificar as melhores soluções para resolver cada um dos problemas, identificando as tarefas necessárias para atingir o objetivo.

O resultado do *Sprint Planning 2* é o *Sprint Backlog* (6), uma lista de atividades necessárias para entregar a versão final das funcionalidades que serão aceitas pelo cliente. Começa o *Sprint* (7), que é a fase de “construção”, onde diariamente o time se reúne para sincronizar o trabalho que está sendo feito. O ciclo menor representa a primeira iteração, em geral de duas a quatro semanas, onde todas as fases do projeto são exercitadas para que seja possível concluir um incremento de funcionalidade pronto para ser utilizado. Dentro deste, o ciclo maior representa as iterações diárias onde o time se planeja novamente para achar soluções aos problemas que surgem durante o desenvolvimento.

Ao final do *Sprint*, uma reunião de *Review* (8) acontece, onde são apresentadas ao cliente as funcionalidades que estão disponíveis para o uso. O cliente pode aceitar ou rejeitar as funcionalidades nesse momento, além de sugerir melhorias ou novas ideias.

Finalmente, para encerrar o ciclo de melhoria contínua, acontece a Retrospectiva do *Sprint* (9), onde será identificado “o que deu certo” e “o que deu errado” no *Sprint*. São identificados pontos de melhoria no processo e levantados impedimentos que atrapalham o melhor desempenho do time.

O Scrum possibilita que as equipes sejam auto-organizáveis e muitas vezes se adequa em engenharias prescritas. O Scrum pode contribuir para a obtenção dos resultados esperados nos projetos executados em ambientes onde a agilidade possa ser empregada, mas para que isso aconteça é necessário que seus eventos sejam realizados com êxito.

Além destas metodologias supracitadas, existe também a metodologia *Lean* que foi desenvolvida primeiramente para as indústrias automobilísticas e posteriormente, para desenvolvimento de sistemas. A mesma será abordada na próxima seção.

### **2.3. Metodologia *Lean***

Nesta seção serão listadas algumas das abordagens que a metodologia *Lean* possui, cada uma delas direcionada para um contexto diferente. Posteriormente será citada e explanada a abordagem *Lean Software Development*, que é a metodologia *Lean* voltada exclusivamente para o desenvolvimento de *software*.

### 2.3.1. Abordagens *Lean*

Em uma visita aos Estados Unidos, o engenheiro chefe da Toyota – Taiichi Ohno percebeu o quanto as fábricas de desenvolvimento de automóveis estavam adiantadas em relação às fábricas japonesas, no que tange à eficiência e qualidade. Isso se dava, em grande maioria, pela grande produção na qual os americanos investiam, e também, pela baixa variedade de seus produtos oferecidos. Percebendo que tais processos não se adequariam ao Japão, por conta das baixas demandas e com um processo econômico em crise, Ohno sugeriu ao sistema Toyota de produção uma maior variedade de produtos, evitando a superprodução e buscando ao máximo a redução de desperdícios (IMAI, 1990).

A partir disto surgiu a necessidade de se criar um novo modelo gerencial, nascendo, assim, o Sistema Toyota de Produção ou *Lean Manufacturing* (Manufatura Enxuta). O conceito de manufatura enxuta acabou se disseminando no mundo dos negócios e com muitas definições, conforme apresentado a seguir:

A eliminação de desperdícios e elementos desnecessários a fim de reduzir custos; a ideia básica é produzir apenas o necessário, no momento necessário e na quantidade requerida (OHNO, 1997).

A busca de uma tecnologia de produção que utilize a menor quantidade de equipamentos e mão de obra para produzir bens sem defeitos no menor tempo possível, com o mínimo de unidades intermediárias, entendendo como desperdício todo e qualquer elemento que não contribua para o atendimento da qualidade, preço ou prazo requeridos pelo cliente. Eliminar todo o desperdício através de esforços concentrados da administração, pesquisa e desenvolvimento, produção, distribuição e todos os departamentos da companhia (SHINOHARA, 1988).

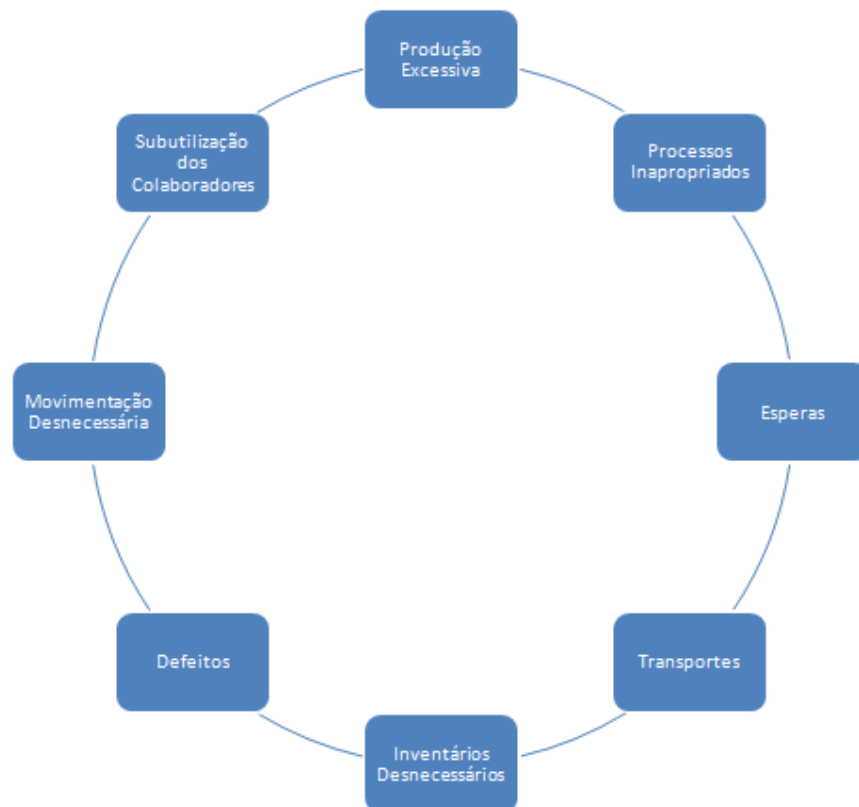
A estratégia *Lean* se encaixa perfeitamente na gerência de negócios que procuram aumentar a satisfação de seus clientes através da melhor forma de aplicar e utilizar os recursos disponíveis. De acordo com o *Lean Institute* Brasil (1999):

A gestão *Lean* procura fornecer consistentemente valor aos clientes com os custos mais baixos (Propósito) através da identificação de melhoria dos fluxos de valor primários e de suporte (Processos) por meio do envolvimento das pessoas qualificadas, motivadas e com iniciativa (Pessoas). O foco da implementação deve estar nas reais necessidades dos negócios e não na simples aplicação das ferramentas *Lean*.

Segundo Womack (2003), “o valor do *Lean* deve ser definido em termos de produtos e serviços específicos, com determinados atributos, com um determinado preço ou custo para clientes específicos”. Com isso, percebe-se que o valor é determinado pelas características e especificações impostas pelos clientes. Uma vez que este fluxo de valor é bem definido, o desperdício tão visado pela metodologia *Lean* pode ser então eliminado.

### 2.3.1.1. Fontes de Desperdício – Produção *Lean*

Toda organização possui fatores que contribuem para alguma forma de desperdícios. Estes desperdícios também serão abordados na próxima seção, porém voltados para o desenvolvimento *Lean* de *Software*. O autor Shingo (1989) identificou sete fontes destes desperdícios e o autor Penfield (2006) a oitava, conforme exibido pela Figura 6.



**Figura 6 - Fontes de Desperdícios Lean**

A Produção Excessiva consiste em produzir muito mais do que o cliente necessita. Esta produção excessiva é proveniente, muitas vezes, da: utilização de lotes de elevada dimensão



ou produto em massa; antecipação da produção; impossibilidade de se efetuar mudança de ferramenta em um curto espaço de tempo; necessidade de criar um estoque (GAIÃO, 2012). As ferramentas *Lean* (Seção 2.4.) podem suprir este desperdício através do trabalho programado; produção em fluxo contínuo; mudança rápida de ferramentas; ou, conforme o autor Gaião (2012), “através do nivelamento da produção e balanceamento dos postos de trabalho”.

Os Processos Inapropriados, segundo o autor Gaião (2012), “referem-se à complexidade desnecessária de alguns processos de produção, que, quando convertidos em processos mais simples, conduzem a resultados mais eficientes”. Isto pode ser claramente visível no manuseamento incorreto de ferramentas; utilização indevida de recursos; falta de formação dos funcionários da equipe, entre outros. Para eliminar esta fonte de desperdício, a metodologia *Lean* propõe a formação dos envolvidos no projeto, análise minuciosa dos processos, trabalhos padronizados, automatização de processos e análises de modo a aplicar sempre a forma mais simplificada na execução do projeto.

As Esperas são representadas pelos períodos inativos de máquinas ou pessoas que trabalham no desenvolvimento de projetos. Os autores Amaro e Pinto (2006) sugerem alguns métodos que podem ser aplicados com a metodologia *Lean* para minimizar estes desperdícios, são eles: “nivelamento da produção; balanceamento dos postos de trabalho; mudanças rápidas de ferramentas”.

Os Transportes segundo Gaião (2012), “dizem respeito à deslocação de materiais de um posto de trabalho para outro”. Esta fonte pode ser causada pela movimentação incorreta dos produtos, gerando danos aos mesmos. A metodologia *Lean* identifica algumas diretrizes para minimizar este desperdício, são elas (GAIÃO, 2012): “flexibilidade operacional; produtos modulares; operadores flexíveis; células de produção”.

Os Inventários Desnecessários referem-se à produção desnecessária de materiais, que acabam por fim, sendo acumulados em estoques. Demeter e Matyusz (2009) afirmam que: “a melhor maneira de encontrar desperdícios é procurar pontos de acumulação, que são os estoques. Estoques elevados representam problemas relacionados à produção”. A metodologia *Lean* propõe modos da diminuição dos estoques, são eles (GAIÃO, 2012): “nivelamento da produção; sincronização da produção; melhoria contínua da qualidade dos processos; mudanças rápidas”.

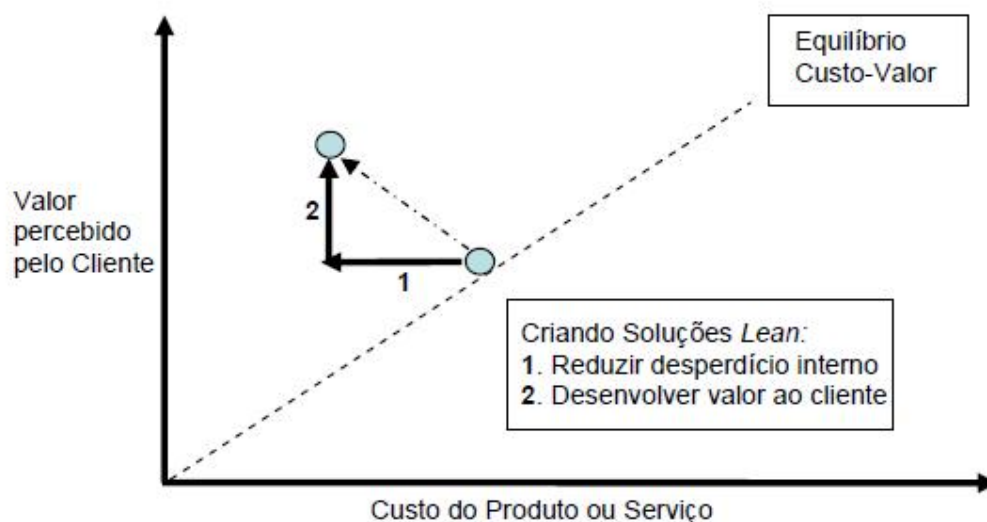
Os Defeitos segundo Gaião (2012) são compostos por “defeitos ou problemas que geram incertezas nos processos, influenciando a capacidade de produção”. A metodologia *Lean* sugere algumas ações para minimizar esta fonte de desperdício, são elas (GAIÃO,

2012): “presença de dispositivos de detecção de erros; automatização de atividades; implementação do trabalho padronizado; eliminação da movimentação de peças e materiais”.

A Movimentação Desnecessária está relacionada às movimentações que não geram valor ao produto. A metodologia *Lean* fornece indicações para a eliminação desta fonte de desperdício, são elas: “garantir fluxo contínuo de produção; formação e treino dos colaboradores; trabalho padronizado; identificação de movimentações que não acrescentam valor e eliminação das mesmas”.

Por fim, a Subutilização dos Colaboradores, que conforme afirmou o autor Ohno (1997) “um dos objetivos do Toyota *Production System* era criar pessoas pensantes”. Esta fonte de desperdício mais nova reconhecidamente retrata o não reconhecimento das empresas perante seus colaboradores. A metodologia *Lean* impõe que o colaborador possa ter voz ativa no projeto, onde o intelecto do colaborador seja fundamental (GAIÃO, 2012).

A Figura 7 apresenta a relação entre valor e custo e mostra como produtos e serviços podem ser representados em relação à sua oferta relativa de custo-valor para o cliente. “O equilíbrio custo-valor denota a situação onde o produto provê exatamente o valor que o cliente está disposto a pagar e o seu custo (FORTES, 2010)”.



**Figura 7 - Relação de Valor, Custo e Desperdício (Fonte: FORTES, 2010)**

A abordagem *Lean* possibilita muitos benefícios dentro das organizações, dentre eles o aumento na rentabilidade, satisfação do cliente em relação ao produto ou serviço adquirido,

redução de prazos na entrega, eliminação de desperdícios, melhor produto final, entre diversas outras vantagens.

Na engenharia de produção, os métodos de redução de perdas são conhecidos como *Lean Manufacturing* (Manufatura Enxuta). Na área de serviços, esses métodos podem ser encontrados como *Lean Service* (Serviço Enxuto). Segundo Nascimento (2004), o *Lean Service* é constituído apenas por atividades que geram valor para o cliente, visando a qualidade e o menor preço. A Tabela 1 exhibe algumas das terminologias e abordagens *Lean*:

**Tabela 1 - Evolução da terminologia *Lean*. (Fonte: Fortes, 2010)**

<b>Autor(es)</b>	<b>Ano</b>	<b>Termo</b>	<b>Conceito</b>
<b>Krafcik, Womack, Jones e Roos</b>	1990	<i>Lean Manufacturing; Lean Production</i>	Eliminação de desperdícios para alcançar vantagens competitivas (indústria automobilística).
<b>Womack e Jones</b>	1994	<i>Lean Enterprise</i>	Encadeamento de atividades que criam valor, gerando um fluxo de valor.
<b>Womack e Jones</b>	1996	<i>Lean Thinking, Lean Principles</i>	Sintetiza o conceito <i>Lean</i> em cinco categorias: valor, fluxo de valor, fluxo contínuo, produção puxada e perfeição.
<b>Bowen e Youngdahl</b>	1998	<i>Lean Service</i>	Princípios <i>Lean</i> voltados para a área de serviços.
<b>Womack</b>	2005	<i>Lean Consumption</i>	Foco no atendimento do cliente, sem desperdiçar seu tempo.
<b>Womack</b>	2008	<i>Lean Management, Lean Process</i>	Liderança <i>Lean</i> , buscando alinhamento de propósito, processos e pessoas. Valor corretamente especificado, de modo que a organização provê o que o cliente realmente deseja.

Existe também a abordagem *Lean Thinking* (Pensamento Enxuto), que consiste em um conjunto de ferramentas que buscam tornar as empresas mais competitivas, por meio da eliminação de atividades que não agregam valor (SENF, 2006). Existem ainda os conceitos *Lean Enterprise* (Empresa Enxuta) e *Lean Consumption* (Consumo Enxuto), ambos visam aproveitar o tempo do cliente, fornecer o que o cliente deseja, onde e quando ele quer. Outro conceito importante é o *Lean Management* (Gerenciamento Enxuto), onde a liderança *Lean* é fundamental para o alinhamento de três elementos (WOMACK, 2008):

1. Propósito: ajudar os clientes a resolverem problemas através da especificação correta de valor, de modo que a empresa possa prosperar;
2. Processos: através de processos *Lean* tão simples e visíveis quanto possível;
3. Pessoas: engajando pessoas profundamente capacitadas na repetida resolução de problemas e padronização.

O *Lean* como um todo é baseado em princípios e no nível mais alto de abstração eles podem ser apenas quatro (PASA, 2004):

1. A produção é programada em função do tempo, de forma a atender a demanda no momento exato;
2. Os estoques devem ser reduzidos continuamente, conforme o espaço físico, oportunidade e ocultamento de ineficiências;
3. A necessidade de mão-de-obra deve ser sempre minimizada;
4. Informações, pessoas, procedimentos e equipamentos devem estar sempre sob controle sistemático e rigoroso.

A Tabela 2 apresenta o crescimento e transformação que o desenvolvimento *Lean* de produtos possibilita, auxiliando e colaborando com uma excelência operacional e uma gestão de negócio mais eficaz.

**Tabela 2 - Desenvolvimento e Excelência *Lean***

<b>Desenvolvimento <i>Lean</i> de Produtos</b>	<b>Excelência Operacional <i>Lean</i></b>
<b>Inovação de serviços, produtos e processos</b>	Reduzir custo unitário dos serviços
<b>Variedade sem complexidade</b>	Melhorar a qualidade dos serviços

**Criar diferenciação**

Tornar mais fácil fazer negócio com a TI

A abordagem *Lean* pode ser definida como uma boa prática e os autores Bhasin e Burcher (2006) defendem a ideia de que “*Lean* tem uma significação estratégica”. Isso se dá pelo fato de que as ferramentas *Lean* aplicadas nas organizações devem ser implantadas em conjunto com a mudança na cultura organizacional das empresas. Caso a filosofia *Lean* não seja implantada com seriedade existem chances de se tornar uma implementação mal sucedida.

O autor Fortes (2010) cita alguns passos para uma implementação satisfatória da metodologia *Lean*, são elas:

1. Aplicar simultaneamente cinco ou mais técnicas ou ferramentas;
2. Encarar o *Lean* como uma jornada de longo prazo;
3. Implantar uma visão de melhoria contínua;
4. Realizar as mudanças culturais referentes à delegação de poder e patrocinar os princípios *Lean* através da cadeia de valor.

Estes passos acima poderão ser facilmente implantados nas organizações e às vezes poderá ser um processo demorado, já que dependerá das estratégias de trabalho e da equipe desenvolvedora. A visão de melhoria contínua poderá ser empregada utilizando a técnica *Kaizen*, que satisfaz o passo 4, onde o ser humano deve ser bem valorizado dentro da empresa. Já a mudança cultural tão visada na implantação da metodologia *Lean* é alcançada com o tempo porque o projeto é envolvido por pessoas, onde cada uma possui sua forma de aplicação e modo de trabalho.

Conforme Kotter (1997), esta mudança cultural tão visada na implementação *Lean* dependendo do seu desenvolvimento pode demorar de cinco a dez anos para se consolidar. Kotter (1997) propõe ainda alguns passos para auxiliar e encaminhar para o sucesso, são eles:

1. Estabelecer um senso de urgência;
2. Formar uma aliança forte para liderar a mudança;
3. Criar uma visão;
4. Comunicar a visão;
5. Distribuir poder para alcançar a visão;
6. Gerar pequenos ganhos;
7. Consolidar as melhorias;

## 8. Institucionalizar o novo comportamento.

Estes passos acima podem ser adaptados de acordo com as estratégias de trabalho de cada organização, todos eles podem e devem ser inseridos e adequados juntamente com a metodologia *Lean*. Estes oito passos servirão para encaminhar de forma satisfatória a mudança cultural, que é implantada de forma demorada, porém eficaz.

Os desperdícios citados na metodologia *Lean* podem ser qualquer atividade humana que absorve recursos e não gera valor. Ohno (1997) afirmou que um antídoto para desperdícios seria o pensamento enxuto, ou seja, fazer mais com menos. Além das sete fontes de desperdícios encontradas há ainda a oitava perda nos serviços, identificada por Francischini, Miyake e Giannini (2006) como: o excesso de capacidade, onde são utilizados parcialmente os recursos e estrutura da empresa de serviços. A Tabela 3 apresenta os princípios da produção enxuta e sua adaptação para serviços.

**Tabela 3 - Princípios da Produção Enxuta e sua adaptação para Serviços (Fonte: PASA, 2004 - adaptado)**

<b>Princípios da produção Enxuta</b>	<b>Princípios <i>Lean</i> aplicados a serviços</b>
<b>Produção programada em função do tempo.</b>	<ol style="list-style-type: none"> <li>1. Fazer atividades que criam valor;</li> <li>2. Fazer somente o que é necessário ao cliente (produção puxada);</li> <li>3. Melhorar o fluxo de valor;</li> <li>4. Envolver-se com os fornecedores.</li> </ol>
<b>Estoques reduzidos.</b>	<ol style="list-style-type: none"> <li>5. Eliminar perdas;</li> <li>6. Eliminar <i>loopings</i> entre atividades;</li> <li>7. Redução do excesso de capacidade.</li> </ol>
<b>Necessidade de mão-de-obra deve ser reduzida.</b>	<ol style="list-style-type: none"> <li>8. Focalizar perdas que não acrescentam valor;</li> <li>9. Equipes multifuncionais.</li> </ol>

<b>Controle sistemático e rigoroso.</b>	10. Descentralizar responsabilidades; 11. Planejamento de produção; 12. Gerenciamento de capacidade; 13. Gestão do conhecimento; 14. Padronização de procedimentos.
---	---

Os quatro princípios *Lean* exibidos na Tabela 3 consistem no gerenciamento de vários fatores envolvidos, como: o conceito *Just-in-time* e estoques e perdas minimizados. O sistema de produção enxuta foi planejado para trabalhar sempre próximo dos limites, afinal, o controle de produção deve estar sempre sob sistemática gerência.

Uma visão geral sobre os conceitos *Lean* abordados até o momento será construída na forma de um mapa mental básico, que é um diagrama de fácil utilização para representar elementos relacionados a uma ideia central (FORTES, 2010). A Figura 8 apresenta uma estrutura básica desse mapa.



**Figura 8 - Mapa mental básico Lean**

A Figura 8 ilustra o mapa mental básico *Lean*, no qual foram abordados conceitos sobre este tema, como: origem de conceito; objetivo; evolução da sua terminologia, já que a metodologia *Lean* foi direcionada em vários contextos e abordagens, e em cada uma delas foi reconhecida em diferentes conceitos; aplicação em serviços; características *Lean*; entre outros conceitos.

Nesta seção foram exibidas algumas das diversas abordagens *Lean* e estas possuem aplicações dependendo do contexto, ou seja, cada uma delas é direcionada para um foco distinto e na próxima seção será abordada a *Lean Software Development*, que é a abordagem *Lean* para desenvolvimento de *software*.



### **2.3.2. Desenvolvimento *Lean de Software* – *Lean Software Development***

Os conceitos de manufatura enxuta e desenvolvimento *Lean* de produtos tiveram início no Sistema Toyota de produção e no Sistema Toyota de Desenvolvimento de Produto, e de acordo com Shore e Wardem (2008), as ideias das quais são baseadas o desenvolvimento *Lean de software* tiveram sua origem a partir destes conceitos.

Segundo Poppendieck e Poppendieck (2011), o desenvolvimento de *software* é uma forma de desenvolvimento de produto. Foram estes autores que deram início a disseminação do conceito de desenvolvimento *Lean de software*, em 2003. O propósito destes autores era identificar os conceitos de *Lean* de forma em que pudessem ser empregados ao desenvolvimento de *software*.

Os conceitos *Lean* deram inspirações ao desenvolvimento ágil e ao desenvolvimento *Lean de software* e segundo Gustavsson (2011), os métodos ágeis são aplicados apenas ao desenvolvimento de *software*, enquanto *Lean* é um conceito muito mais amplo. A filosofia *Lean* não é apenas um conjunto de ferramentas, ela afeta todos os setores da empresa, desde os recursos humanos até o *marketing*.

Os sete princípios do desenvolvimento *Lean de software*, (POPPENDIECK; POPPENDIECK, 2011) são apresentados a seguir.

#### **1. Eliminar o desperdício**

O Sistema Toyota de Produção tem como objetivo principal a eliminação total do desperdício, onde tudo que não agrega valor para o cliente deve ser removido do processo, Ohno (1988). Segundo Hibbs, Jewett e Sullivan (2009), os desperdícios indicados pelo Sistema Toyota de Produção podem ser identificados no ambiente de desenvolvimento de *software*, a partir de uma série de conceitos que devem ser observados e analisados para que seja possível compreender os desperdícios, como exibido pela Figura 9.



**Figura 9 - Fontes de Desperdícios Lean de Software**

A Figura 9 ilustra as possíveis fontes de desperdícios, são elas: Defeitos: São representados pelos defeitos em si. O desenvolvimento *Lean de software* tem como foco principal prevenir os defeitos, já que os defeitos causam o retrabalho custoso e demorado, não agregando valor ao produto final. Superprodução: É representada pelo excesso de funcionalidades desnecessárias. O custo de um *software* está além do código fonte, pois este código precisa ser mantido, documentado, repassado a novos membros da equipe, entre outros. Por este motivo, todas as funcionalidades inseridas no *software* devem agregar valor ao produto final e devem satisfazer as necessidades reais do usuário.

Segundo Hibbs, Jewett e Sullivan (2009), 64% de todas as funcionalidades de um sistema não são usadas ou são usadas raramente. Isto compõe um grande desperdício de recursos ao longo do tempo. Estoque: É representado pelas tarefas e atividades concluídas parcialmente, como por exemplo, requisitos analisados e não implementados, erros que não foram corrigidos, trechos de códigos que não foram testados, entre outros.

A filosofia *Lean* não admite o acúmulo deste estoque e, para isso, é adotado o fluxo unitário onde a tarefa deve ser concluída o quanto antes. Movimentação: É representada pela

alternância entre tarefas, como por exemplo, as interrupções. Qualquer interrupção faz com que processos sejam reiniciados, prejudicando a produtividade. No ambiente de trabalho, as pessoas devem ter tempo para se ambientar aos problemas e compreender requisitos passados, colocando em prática o fluxo unitário, conhecido por sua produtividade e eficácia. Processamento adicional: É representado por processos desnecessários, como por exemplo, criação de documentação que não será utilizada por ninguém. Outro exemplo é a execução manual de tarefas que poderiam ser realizadas automaticamente. Estes tipos de processos atrapalham a produtividade e não agregam valor algum ao produto final.

A Espera é representada pelos atrasos, como por exemplo, a necessidade de se comunicar com outros integrantes do projeto de desenvolvimento de *software*, como programadores, entre outros. Caso o integrante não esteja disponível haverá atrasos na entrega ou a implementação será feita sem os devidos esclarecimentos, gerando na maioria das vezes, retrabalho.

## **2. Integrar qualidade**

Segundo Ohno (1988 apud BOSNIC, 2012) não é possível inspecionar a qualidade de um produto ao fim da linha de produção. Segundo Hibbs, Jewett e Sullivan (2009), “as metodologias tradicionais de desenvolvimento cometem exatamente esse erro: permitir que os defeitos sejam detectados tardiamente pela equipe de garantia de qualidade”.

A filosofia *Lean* destaca a eliminação total de defeitos e para alcançar esse grau de maturidade no processo só é possível com a utilização de recursos como testes unitários e integração contínua, entre outros. (BOSNIC, 2012).

## **3. Criar conhecimento**

Segundo Bassi (2008), “as lições devem ser extraídas das experiências vividas pela equipe”. Hibbs, Jewett e Sullivan (2009) destacam que o conhecimento deve ser armazenado de forma tal que da próxima vez que este conhecimento for necessário ele seja facilmente localizado. Isso corrobora para que as pessoas não percam tempo aprendendo algo que já foi estudado e até mesmo executado por outros membros das equipes de desenvolvimento do *software*.

## **4. Adiar comprometerimentos**

Segundo Poppendieck e Poppendieck (2011), este princípio se aplica principalmente à tomada de decisões irreversíveis. Já as decisões reversíveis devem ser tomadas antes, porque elas podem ser facilmente modificadas (BOSNIC, 2012).

Se uma decisão não precisa ser tomada imediatamente, o melhor a se fazer é aguardar até que se tenha um melhor e maior conhecimento sobre o assunto. Hibbs, Jewett e Sullivan (2009) afirmam “que as melhores decisões são tomadas quando dispomos de maior quantidade de informação possível”.

## **5. Entregar rápido**

Kniberg (2011) afirma que o essencial para dar início em grandes projetos é ter o conhecimento necessário sobre aquilo que gera e agrega valor ao cliente. Depois de percebidas as necessidades do cliente, o essencial a ser feito é criar fluxos de trabalho, onde seja possível fazer entregas rápidas e frequentes do *software* funcionando.

De acordo com Hibbs, Jewett e Sullivan (2009), a importância de entregar rápido está em obter o retorno do cliente o quanto antes. Isso impossibilita, também, que os requisitos tenham que ser editados apenas por demora na entrega.

## **6. Respeitar as pessoas**

Este princípio prioriza e valoriza o respeito às pessoas envolvidas no projeto, fornecendo-lhes a confiança necessária para que atuem diante das situações da maneira em que acharem mais conveniente e útil para o desenvolvimento e melhoramento do *software* e seus processos. Segundo Poppendieck e Poppendieck (2011), “pessoas pensadoras e engajadas no projeto são a maior e a mais sustentável vantagem competitiva que uma empresa pode ter”.

## **7. Otimizar o todo**

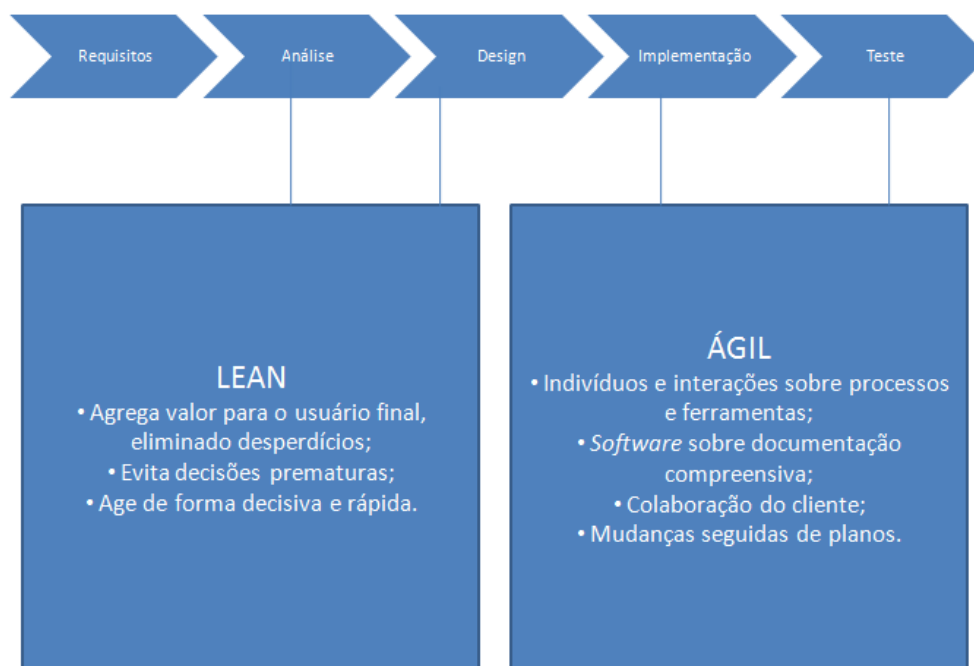
Segundo Bosnic (2012), “aperfeiçoar um processo local quase sempre é feito à custa do fluxo de valor no processo como todo”. As organizações que implementam o *Lean* devem tentar evitar o ato de executar mudanças sem levar em consideração todo o processo.

As ferramentas *Lean* auxiliam na identificação e eliminação de desperdícios, identificando e eliminando o tempo e custos de um produto ou serviço. Na engenharia de

*software*, segundo o autor Hayata (2012), “os desperdícios podem ser definidos em muitos meios, em cada fase do ciclo de desenvolvimento. Programação ágil para acelerar a produção pode gerar desperdícios, devido a abstração prematura do domínio do problema proposto”.

A ideia que se tem de aplicar os princípios *Lean* no desenvolvimento de *software* é antiga, isto pode ser afirmado com o estudo de Charette (1990), onde o autor utilizou o termo “desenvolvimento *Lean*” para se referir a uma estratégia de gestão que pode trazer estabilidade dinâmica nas organizações, contribuindo com a agilidade e resistência da empresa. Segundo Charette (1990), “o único propósito de um negócio é criar e servir o cliente. Neste espírito, o desenvolvimento *Lean* não foca somente no processo de desenvolvimento por si só, mas sobre como usá-lo para criar valor para o cliente”. Assim, o desenvolvimento *Lean* de *software* pode ser considerado um conjunto de práticas, princípios e filosofias para a construção de sistemas de *software* para uso de clientes.

A Figura 10 ilustra o processo de que enquanto a metodologia *Lean* é utilizada para análise de *software* e exigência nos processos arquitetônicos, o Ágil foca muitas vezes na implementação e testes de *software*.



**Figura 10 - Lean e Métodos Ágeis**

Apesar das distinções exibidas na Figura 10, as abordagens possuem algumas características em comum, como por exemplo: artefatos leves; evita-se o retrabalho; e, o pensamento adaptativo (HAYATA, 2012).

Esta seção abordou a metodologia *Lean* direcionada para o desenvolvimento de *software*. Na próxima seção será abordado o fluxo que esta metodologia possui, ilustrando os passos e etapas que devem ser seguidos para uma implantação satisfatória na gerência de projetos.

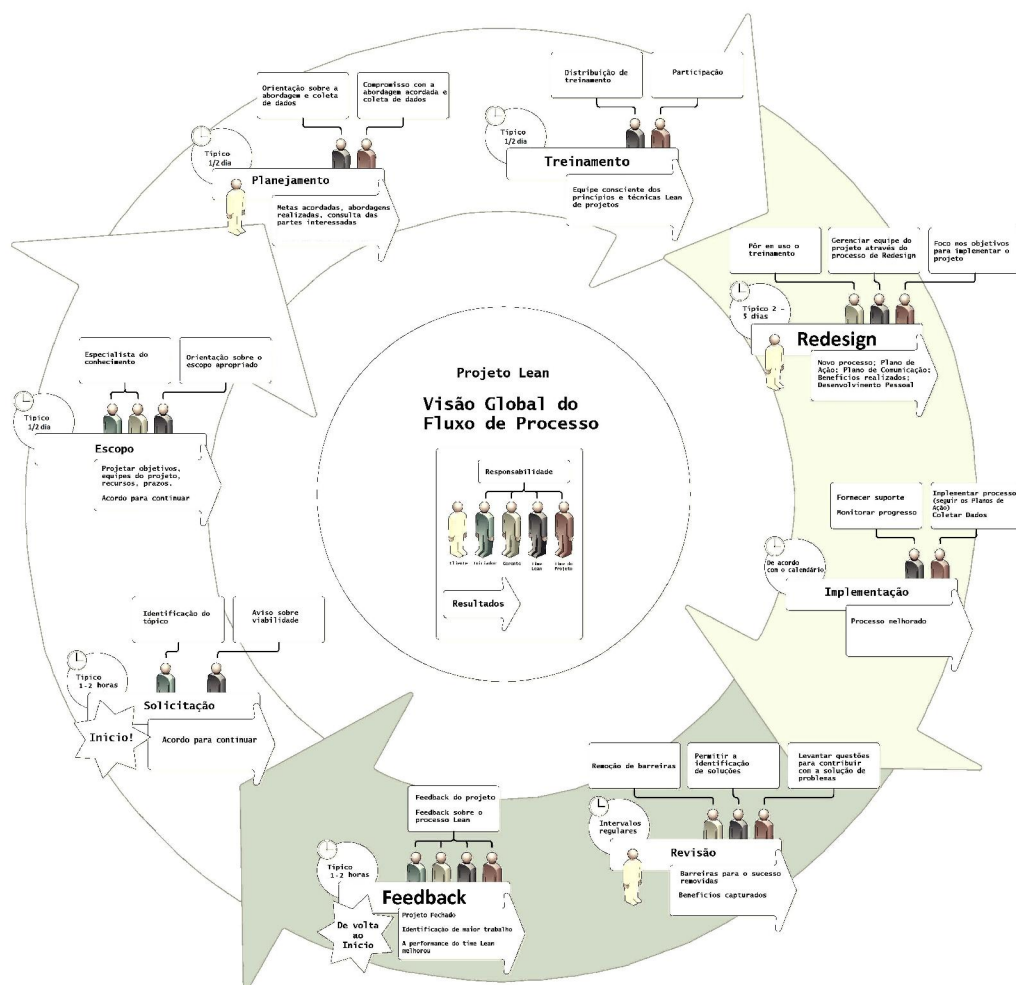
### **2.3.3. Fluxo da metodologia *Lean***

A metodologia *Lean*, quando implantada, segue um fluxo que é composto por processos, papéis, cronogramas e treinamentos. Vale ressaltar que existem três itens essenciais na implantação deste fluxo, que são: envolver a liderança no desenvolvimento de projetos para alcançar uma mudança cultural; realizar a verificação dos benefícios que a metodologia *Lean* trouxe no desenvolvimento do projeto; e, as experiências obtidas que podem criar uma cultura *Lean*.

É possível verificar o quanto a metodologia *Lean* elimina os desperdícios, aumenta o valor do produto, melhora a forma como o trabalho é realizado pelas equipes e aumenta a satisfação do cliente. A metodologia *Lean* é implantada para causar essa mudança cultural dentro da organização.

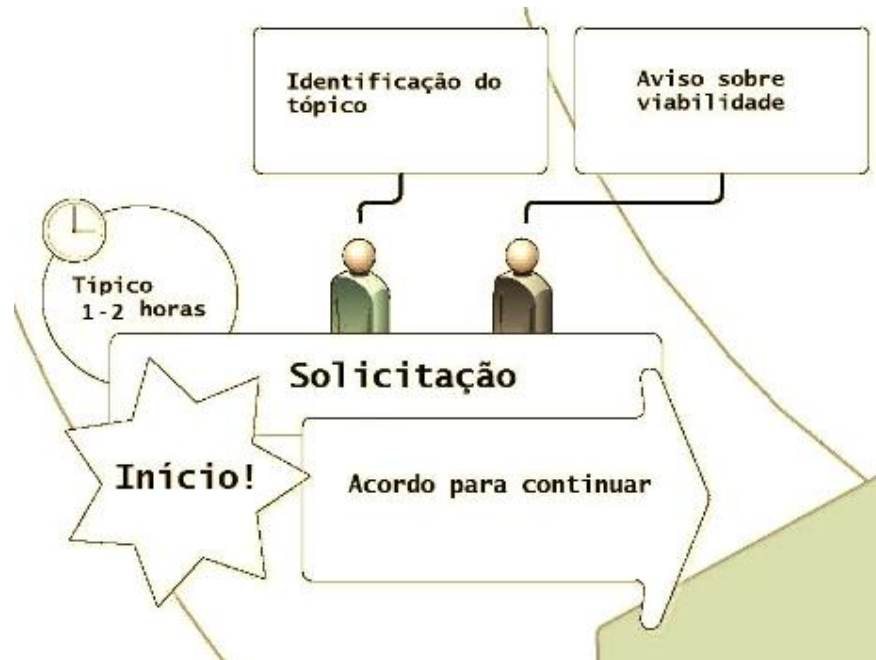
Entre os benefícios encontrados quando se aplica *Lean* está o entusiasmo da equipe desenvolvida, pois o *Lean* proporciona o incentivo pessoal da equipe através do princípio Respeitar as Pessoas, onde acredita-se que os membros das equipes são os bens mais valiosos das organizações. Com a equipe motivada, a qualidade do produto se eleva, além disso, a metodologia *Lean* fornece oficinas de treinamento *Lean*, nas quais sempre são debatidas as experiências vividas em cada projeto. Isso auxilia na verificação de prazos e custos antes e depois do *Lean*.

As experiências que possibilitam criar uma cultura *Lean* são necessárias para que cada membro da equipe possa aplicar a melhoria contínua, de forma contínua. Isto gerará uma melhor aplicação dos recursos investidos e conseqüentemente, equipe e cliente satisfeitos. A Figura 11 ilustra uma visão global de todo este fluxo de processos.



**Figura 11 - Fluxo Lean (Fonte: UNIVERSITY OF ST ANDREWS, 2011 - Adaptado)**

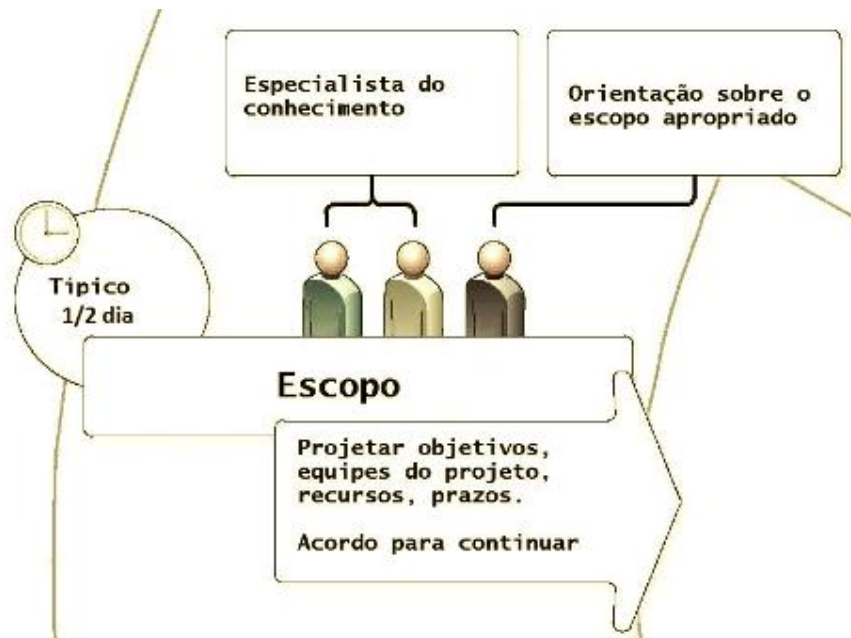
Conforme exibido pela Figura 11, o fluxo da metodologia *Lean* engloba oito passos, são eles: Solicitação; Escopo; Planejamento; Treinamento; *Redesign*; Implementação; Revisão e *Feedback*. Cada um destes processos será explicado a seguir.



**Figura 12 - Solicitação**

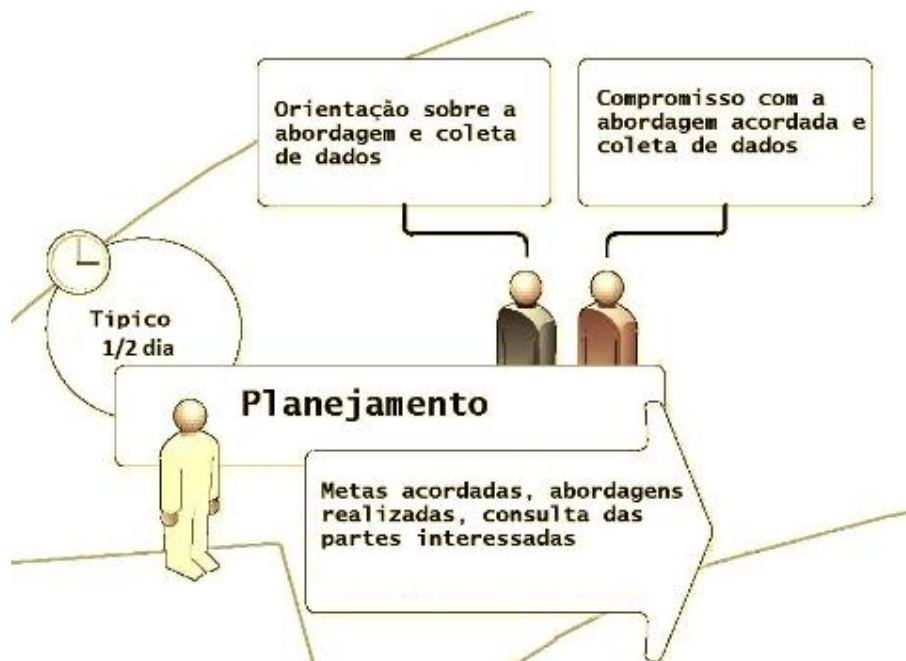
O fluxo *Lean* inicia-se com a solicitação, ou seja, quando existe um projeto a ser desenvolvido. As funcionalidades desejadas e os tópicos necessários são identificados e o responsável pelo projeto verifica a viabilidade de desenvolvê-lo. Se existe um acordo entre funcionalidades e tópicos, e, viabilidade, o processo continua e vai para a próxima etapa, que é o escopo. Vale ressaltar que na etapa de solicitação tudo é definido de forma informal, pois as variáveis analisadas são apenas pensadas como um todo e não detalhadamente.





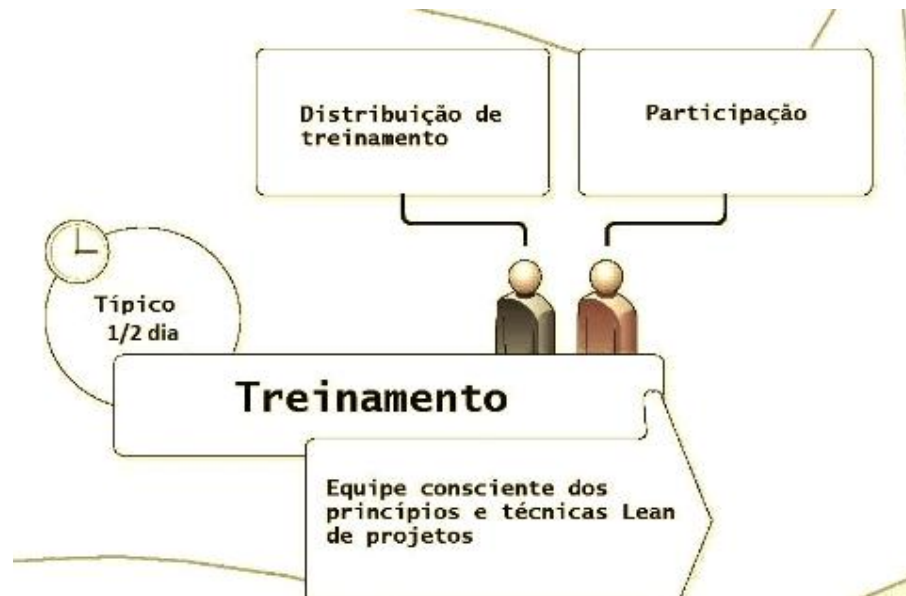
**Figura 13 - Escopo**

A segunda etapa do processo, que geralmente acontece no segundo dia do projeto, é o escopo, onde são projetados os objetivos, quais serão as equipes que irão desenvolver o projeto, quais serão os recursos e os prazos necessários para concluir o projeto. Para desenvolver o escopo, a metodologia *Lean* indica dois especialistas do conhecimento (número variável de acordo com o projeto), pois eles são responsáveis em definir os itens a serem colocados no escopo. Além dos especialistas do conhecimento, existe um responsável em orientar qual escopo é mais apropriado de acordo com as funcionalidades e viabilidade pré-definidas.



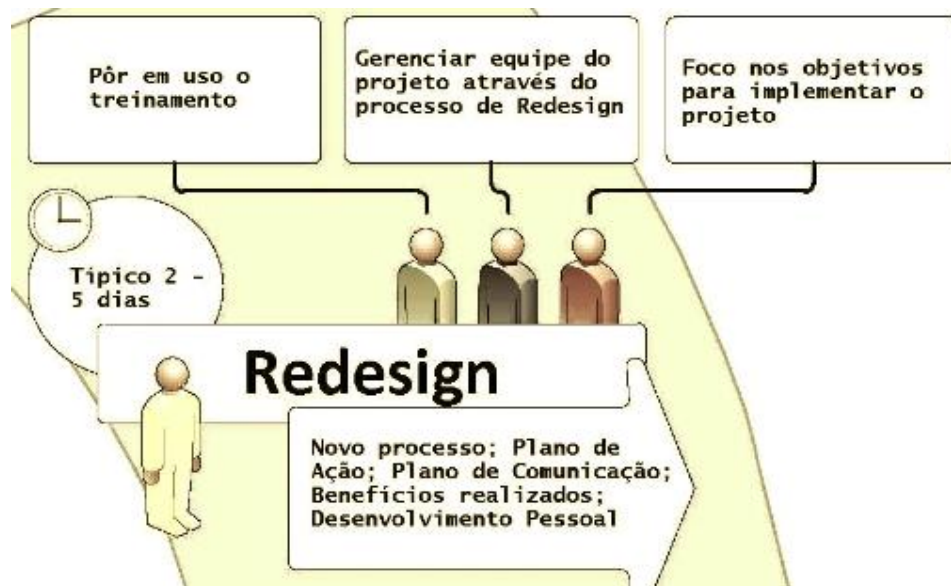
**Figura 14 - Planejamento**

Já feita a solicitação de um novo projeto e já definido o escopo, é iniciada a terceira etapa do processo, que é a etapa de planejamento. Nesta etapa são planejadas as metas acordadas entre os especialistas, quais serão as abordagens que serão utilizadas, é definido o calendário de atividades e o plano de coleta de dados. É nesta etapa, também, que é realizada a consulta entre as partes interessadas no desenvolvimento do projeto, pois assim que os dados são coletados é feito um acordo entre o que se obteve durante a coleta e a abordagem definida.



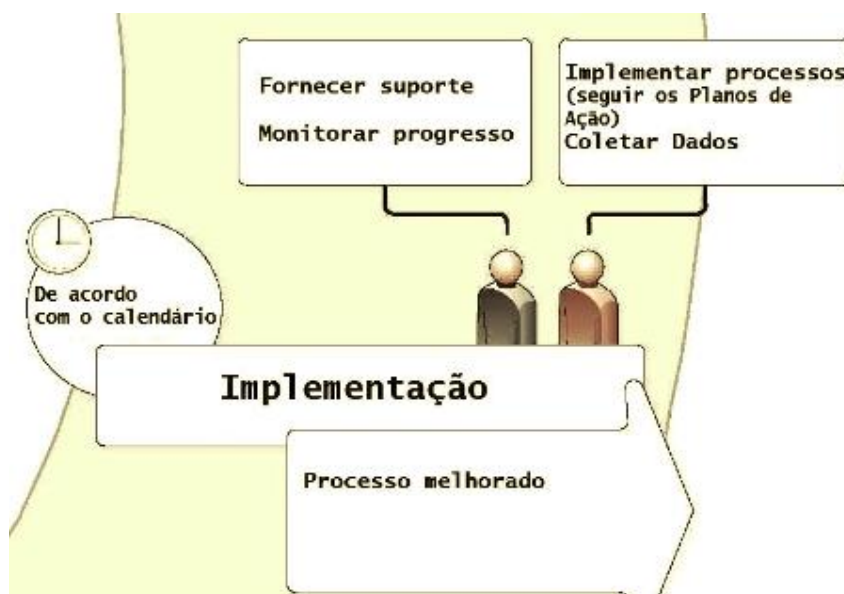
**Figura 15- Treinamento**

A quarta etapa do processo é a etapa de treinamento. Nesta etapa, como o próprio nome já diz, as equipes são treinadas e devem estar conscientes sobre os princípios e técnicas da metodologia *Lean* de projetos. Existe um responsável que irá distribuir esse treinamento de acordo com as responsabilidades de cada membro e claro, haverá a participação de todos os membros envolvidos na execução do projeto.



**Figura 16 - Redesign**

A quinta etapa do processo é a etapa de *redesign*, onde o que foi aprendido na fase anterior, fase de treinamento, é colocado em prática. O *redesign* é considerado um processo que auxilia o gestor a gerenciar o projeto, onde são abordados: o novo processo, o plano de ação, o plano de comunicação, os benefícios realizados; e, o desenvolvimento pessoal da equipe. Esta etapa é de suma importância para o projeto como um todo, pois é nela que os objetivos a serem implementados ganham um foco para serem iniciados.



**Figura 17 - Implementação**

A sexta etapa do processo é a etapa de implementação, onde os planos de ação são seguidos e os processos são implementados. Também acontece a coleta de dados, de modo a melhorar os processos. Um item bastante importante nesta etapa são as atualizações regulares sobre o progresso do projeto que está sendo desenvolvido. Este item refere-se ao andamento do projeto, se o mesmo está cumprindo os prazos e custos estabelecidos, se a viabilidade realmente está positiva, se o tempo atual onde se encontra o projeto está de acordo com o calendário, ou seja, monitora-se o progresso.

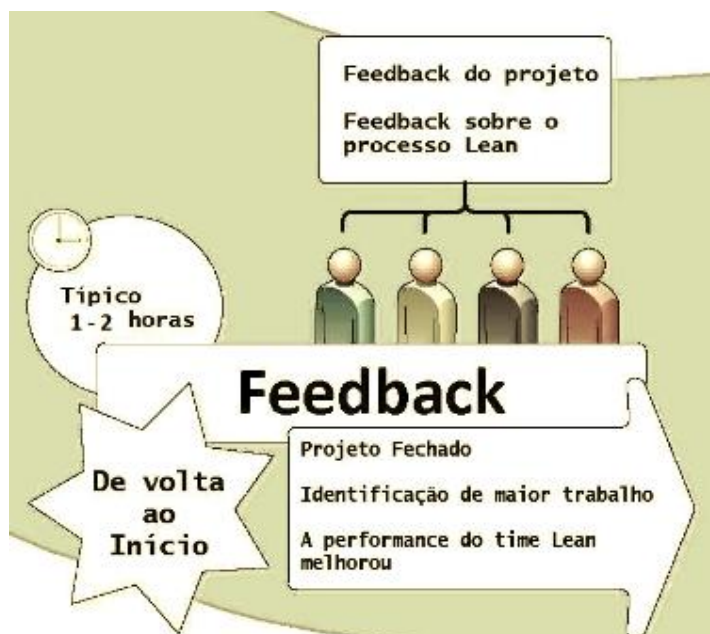
Nesta etapa é importante colocar todas as funcionalidades em um quadro *Kanban* dividido, inicialmente, por três colunas, sendo a primeira coluna a listagem das atividades a serem desenvolvidas, a segunda coluna as atividades em andamento (essa coluna facilita a verificação dos responsáveis por cada atividade e se a mesma se encontra nos prazos estabelecidos) e, a terceira coluna as atividades já concluídas. Esta última coluna é essencial

para que todos os envolvidos no projeto vejam o que já foi feito e tenham uma visão geral do sistema desenvolvido.



**Figura 18 - Revisão**

Como a etapa anterior foi a etapa de implementação, surgem muitos problemas em sua execução. A sétima etapa, que é a etapa de revisão, é responsável em levantar sugestões para contribuir com a solução de problemas, ou seja, estas soluções são identificadas e possíveis barreiras que até então contribuíam para o atraso do projeto, são removidas. Os dados gerados são processados e analisados e os benefícios *Lean* são capturados, ou seja, são evidenciados. Esta etapa acontece frequentemente, com intervalos regulares, para auxiliar o andamento do projeto.



**Figura 19 - Feedback**

A última etapa do projeto, que é a etapa de *Feedback*, acontece com o projeto concluído, identificação dos trabalhos desenvolvidos, identificação da atividade que mais ajudou e/ou que mais demorou para o andamento do projeto, também é verificada se a performance do time *Lean* melhorou a execução do projeto, entre outros. Esta etapa acontece como um retorno sobre tudo que aconteceu antes, durante e após o término do projeto, ou seja, ela foca nas vantagens e desvantagens da implantação da metodologia *Lean*, nas experiências vividas pelas equipes e na satisfação dos envolvidos com a aplicação da fórmula mágica *Lean*, que objetiva reduzir custos, eliminando desperdícios e aplicando os recursos da melhor forma possível.

Esta seção abordou a metodologia *Lean Software Development*, conforme suas práticas e princípios. No entanto, existem diversas ferramentas que possibilitam chegar de forma satisfatória nos focos almejados, algumas delas são aplicadas juntos com a metodologia *Lean* e estas serão abordadas na próxima seção.

## **2.4. Ferramentas de Produção**

Nesta seção serão listadas algumas das ferramentas e métodos que a metodologia *Lean* recorre frequentemente. Estas ferramentas são as práticas usadas para se atingir os objetivos identificados (GAIÃO, 2012). As ferramentas que serão abordadas nas seções seguintes serão

*Kanban*, *Kaizen* e *Just-in-time*, todas elas são ferramentas de apoio à metodologia *Lean*, ou seja, elas completam e auxiliam a implantação da metodologia.

#### **2.4.1. Kanban**

A definição geral do *Kanban* segundo Moura (1999), é que “*Kanban* é um método que reduz o tempo de espera, diminuindo o estoque, melhorando a produtividade e interligando todas as operações em um fluxo uniforme ininterrupto”. O principal objetivo do *Kanban* é eliminar todo o tempo, material, estoque, produtos, entre outros, que interrompem e atrasam o desenvolvimento dos projetos.

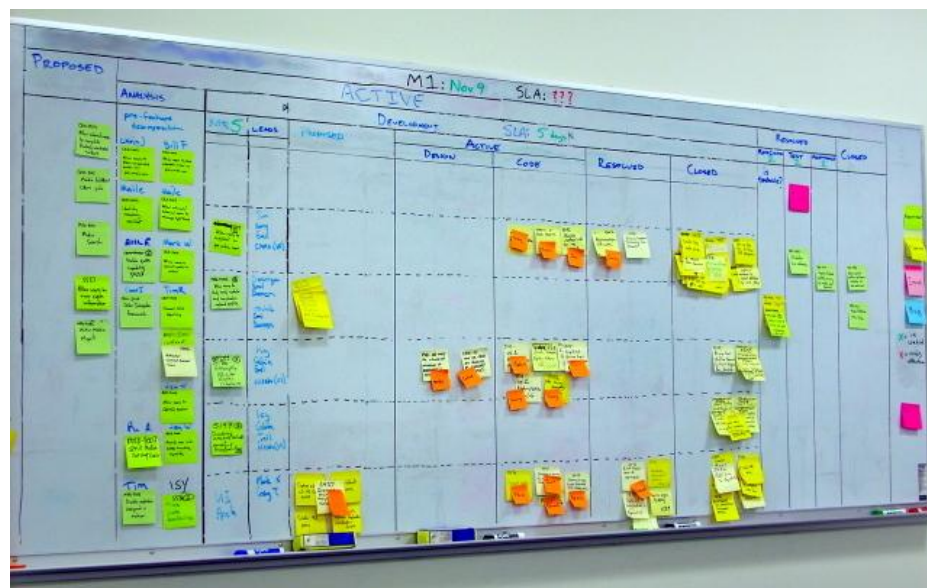
Ainda conforme Moura (1999), “o *Kanban* é uma técnica de programação em curto intervalo de tempo que usa cartões ou *containers* para acionar o puxar de materiais de um processo para outro”. Por ser um método que foca no término de uma fase para dar início a outra fase, acaba por ser mais viável a sua utilização em projetos que exigem um controle maior em cada parte isolada dos projetos.

Segundo Monden (1984), “*Kanban* é uma ferramenta para se obter produção no tempo exato. *Kanban* é um cartão normalmente colocado em um envelope retangular de vinil”. O *Kanban* resultou do Sistema Toyota de Produção com o objetivo e propósito de aumentar a eficiência e qualidade da produção juntamente com a eliminação de desperdícios.

Moura (1992) possui uma definição onde ele cita a movimentação e produção *Kanban*:

O sistema *Kanban* emprega um cartão para sinalizar a necessidade de movimentar mais peças e um cartão idêntico ou semelhante para sinalizar a necessidade de produzir mais peças. Na maioria dos sistemas *Kanban*, todos os componentes ou peças possuem o seu próprio contenedor especial, projetado para manter quantidades precisas da peça, de preferência uma quantidade muito pequena.

“O sistema *Kanban* controla a produção dos produtos necessários nas quantidades necessárias e no tempo necessário, em todo o processo de uma fábrica e também entre companhias (MONDEN, 1984)”. O *Kanban* focaliza o trabalho sempre rumo ao progresso, apresentando a melhor forma visual da evolução dos projetos, tornando os problemas mais evidenciados e favorecendo a melhoria contínua (PEDROSA, 2012).



**Figura 20 - Exemplo da utilização do quadro Kanban (Fonte: PUPO, 2011)**

Esta técnica é um meio de controlar as informações, pois ela separa as informações julgadas desnecessárias das necessárias, alcançando um resultado satisfatório com o mínimo de informações possíveis. Segundo Moura (1999), “os empregados que utilizam o *Kanban* se sentem mais motivados e empolgados para ativar as metas através de meios inovadores”, já que as metas visíveis de desempenho são expostas para todos que fazem parte, direta ou indiretamente, do projeto.

O *Kanban* possui algumas funções, evidenciadas por Sacomano e Fusco (2007), são elas:

- Prover coleta ou transporte de informação;
- Prover informação de produção;
- Impedir a produção e o transporte em excesso;
- Prevenir a saída de produtos defeituosos, identificando os processos que levam à sua produção;
- Revelar problemas existentes e manter o controle dos estoques.

Estas funções citadas acima poderão ser evidenciadas assim que a técnica *Kanban* for implantada no projeto, já que a mesma foca na visualização das etapas que necessitam ser desenvolvidas, atividades em andamento e atividades já concluídas. Facilitando assim



minimizar o excesso e gerar um melhor direcionamento de recursos, revelando problemas e responsáveis pelas atividades.

Segundo a *Productivity Press Development Team* (2002), o *Kanban* atua como um sistema de informação que integra o *layout*, liga todos os processos entre si e liga toda a cadeia de valor gerada harmoniosamente pela procura do cliente. O *Kanban*, segundo o autor Gaião (2012) é:

Um método de puxar a produção (sistema *pull*) a partir da procura, ou seja, o ritmo de produção é determinado pelo ritmo de circulação de *kanban's*, o qual, por sua vez, é determinado pelo ritmo de procura dos produtos, por um cliente interno ou externo à organização.

O *Kanban* é essencial na produção *Just-in-time*, pois permite controlar o fluxo de produção, mantendo-o contínuo, permite balancear a carga de trabalho e, também, nivelar a produção (CHEN e SUBRAMANIAM, 2012).

Existem diversos tipos de cartões *Kanban*, entre eles pode-se citar o *Kanban* de Requisição e o *Kanban* de Ordem de Produção. Um *Kanban* de Requisição “especifica o tipo e a quantidade do produto que o processo subsequente deverá retirar do processo precedente (MONDEN, 1984)”, enquanto que o *Kanban* de Ordem de Produção “especifica o tipo e a quantidade do produto que o processo precedente terá que produzir (MONDEN, 1984)”. Além destes dois tipos, o *Kanban* possui alguns demais tipos que serão descritos a seguir, sendo cada um deles adequado a diferentes situações, podendo variar de empresa para empresa.

- O *Kanban* de Requisição contém as seguintes informações (MONDEN, 1984):
  - Descrição da peça;
  - Capacidade do *container*;
  - Número de emissão do *Kanban*;
  - Centro de trabalho precedente e seu local de estocagem;
  - Centro de trabalho subsequente e seu local de estocagem.

Estas informações poderão variar de acordo com o contexto aplicado o cartão *Kanban*, ou seja, os dados preenchidos nos cartões dependerá do cenário aplicado. Nas informações acima, existe toda uma preocupação com origem e destino do produto.

○ O *Kanban* de Ordem de Produção contém as seguintes informações (MONDEN, 1984):

- Descrição da peça, com identificação do código e nome desta;
- Descrição do processo e do centro de trabalho onde a peça é fabricada;
- Capacidade do *container*, o que indica quantas peças devem ser produzidas para este *container* específico;
- Local para estocagem, indicando o local onde as peças devem ser colocadas uma vez fabricadas;
- Número de emissão do *Kanban*, que indica o *container* onde o *Kanban* foi anexado e a relação do total de *containers* em uso no centro de trabalho específico, isto é, a quantidade total de peças em processo.

Estas informações poderão variar de acordo com o contexto aplicado o cartão *Kanban*, ou seja, os dados preenchidos nos cartões dependerá do cenário aplicado. Nas informações acima, existe toda uma preocupação com a quantidade de produtos produzidos.

○ *Kanban* de Fabricação: Segundo Ribeiro (1989), “é o cartão que acompanha as peças nos *containers* durante o processo de fabricação até as linhas de montagem”. Este possui informações para facilitar e orientar os fluxos de processos, através das informações do cotidiano e de rotina que estarão contidos nele.

○ *Kanban* de Matéria-Prima: Segundo Ribeiro (1989), “é o cartão usado entre os setores de início de fabricação e os almoxarifados de matéria-prima”. A utilização deste implica em uma requisição de matéria-prima para produção de novos produtos.

○ *Kanban* de Montagem: Segundo Ribeiro (1989), “é o cartão usado entre o setor de montagem e o supermercado de peças, representando a sequência de programação da montagem”. Este é extremamente importante dentro das organizações porque pelo modo como ele é utilizado é que dá para produzir as quantidades exatas programadas.

○ *Kanban* de Fornecedor: Segundo Ribeiro (1989), “é o cartão usado entre os setores de montagem e o fornecedor”. Este possibilita que os *containers* com as peças sejam repostos diretamente no local onde estão sendo utilizados, sem necessidade de inspeção referente a quantidade e/ou qualidade.

Estes modelos de acionamento via *Kanban*, segundo Tubino (2000), têm nivelamento de estoques calculados e podem ser controlados por meio de pontos de pedidos, que consistem em estabelecer uma quantidade de itens em estoque. Quando estes atingem esse ponto de reposição, dá-se a partida ao processo de reposição do lote consumido.

O fato percebido por Kniberg e Skarin (2009) é que o *Kanban* é uma abordagem para mudança gerencial, e ainda segundo eles:

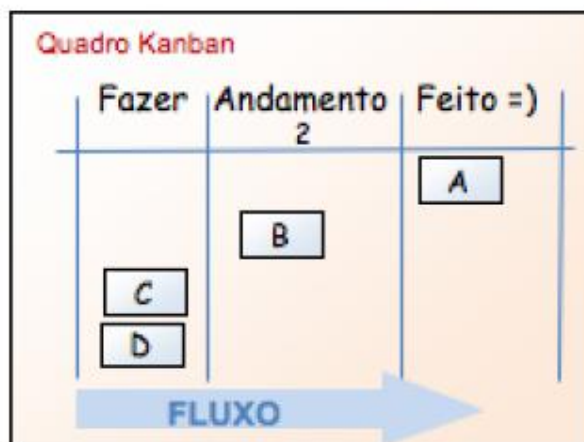
*O Kanban não é um processo ou ciclo de vida de gerenciamento de projetos ou de desenvolvimento de software. O Kanban é uma abordagem para introduzir mudanças em um ciclo de desenvolvimento de software. O Kanban é uma abordagem para introduzir mudanças em um ciclo de desenvolvimento de software ou metodologia de gerenciamento de projetos.*

Os cartões sinalizadores, juntamente com a utilização do gerenciamento *Kanban* possibilitam iniciar todo o processo de gerência de qualquer ponto em que a organização esteja e com ele é possível mapear todo o fluxo de valor e entender o que se passa entre os processos. O *Kanban* tem sido útil em equipes ágeis de desenvolvimento de *software*, mas por sua utilidade eficiente vem sendo utilizado por equipes que usam abordagens mais tradicionais. Ele está sendo introduzido como parte de uma iniciativa *Lean* para moldar a cultura das organizações e encorajar a melhoria contínua.

O fluxo de valor é controlado pelo *Kanban* através das várias etapas que o fluxo contém, já que o *Kanban* possibilita um melhor controle visual. Como supracitado, para utilização do *Kanban*, tipicamente utiliza-se um quadro branco com *post-its*, ou um sistema de cartões eletrônicos (KNIBERG e SKARIN, 2009).

Utilizar estes dois métodos facilita a visualização da mudança cultural, já que estes métodos contribuem com a transparência. O *Kanban*, no entanto, vai um passo além desta transparência, já que ele expõe os gargalos, filas, variabilidade e desperdício. Essa exposição proporciona uma melhor visibilidade aos membros de equipes sobre os efeitos de suas ações (ou falta delas). A visibilidade incentiva na melhoria dos processos, já que tudo que acontece nas organizações é bem mais difundido entre os funcionários. Kniberg e Sharin (2009) afirmam que “o *Kanban* muda o comportamento e incentiva uma maior colaboração no trabalho”.

No *Kanban* as atividades em andamento são limitadas pelo fluxo de trabalho, conforme exibido na Figura 21.



**Figura 21 - Exemplo básico de quadro Kanban (Fonte: KNIBERG e SKARIN, 2009)**

A Figura 21 ilustra o processo de atividades em andamento, sendo que no máximo duas podem estar no estado “Andamento” ao mesmo tempo, independente do seu status. Ao utilizar o quadro *Kanban* é necessário escolher qual limite aplicar para cada estado dentro do fluxo de trabalho, mas a ideia geral é limitar as atividades em andamentos de todos os estados e suas fases, iniciando o quanto antes e terminando o mais tarde possível dentro dos limites estabelecidos.

É necessário colocar todos os ciclos de *feedback* acima dos processos, para isso o *Kanban* oferece algumas métricas, conforme os autores Kniberg e Skarin (2009):

- Tempo médio de execução, atualizado a cada vez que um item atinge o *Done* (ou o que for colocado na coluna mais à direita no quadro *Kanban*);
- Gargalos, um sintoma típico é que a coluna X está repleta de itens, enquanto a coluna X+1 está vazia.

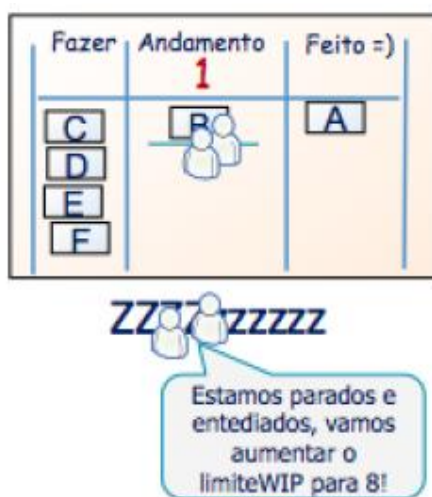
A vantagem em se utilizar as métricas em tempo real, é que isso possibilita definir o comprimento do ciclo de *feedback*, com base na frequência com que você está disposto a analisar as métricas e fazer mudanças. Ciclos longos são sinônimos de melhorias com certa lentidão, em contrapartida, os ciclos curtos podem causar perdas, por falta de tempo para estabilizar cada mudança.

Por ser uma metodologia mais adaptativa e flexível, o *Kanban* possui poucas restrições, como Visualização do Fluxo de Trabalho e Limitação de Atividades em Andamento. Isso é extremamente trivial em desenvolvimento de projetos, e apenas com a utilização destas restrições já é possível verificar uma maior satisfação no gerenciamento de projetos. O

*Kanban* não prescreve papéis, isso irá variar de acordo com as estratégias de trabalho dos gestores. Esse fator não significa que não se devem ter papéis, mas que apenas não é necessário, já que um raciocínio muito comum quando se utiliza o *Kanban* é que “menos é mais”.

Um cenário será ilustrado nas próximas páginas, onde existe um projeto em desenvolvimento com 14 atividades a serem desenvolvidas, duas equipes, um servidor de integração (que servirá apenas para exemplificar o cenário) e três status: fazer (que serão as atividades que deverão ser feitas); andamento (que serão as atividades em andamento); e, feito (que serão as atividades já desenvolvidas).

Como supracitado uma das restrições do *Kanban* é a limitação do trabalho em progresso e isso variará de acordo com as equipes, projetos, atividades e estratégias pré-definidas. A Figura 22 ilustra o ócio entre membros da equipe de desenvolvimento, que por conta de uma limitação mal direcionada, faz com que o projeto tenha um prazo maior de entrega do que realmente deveria.



**Figura 22 - Mau direcionamento da limitação do trabalho em progresso (Fonte: KNIBERG e SKARIN, 2009)**

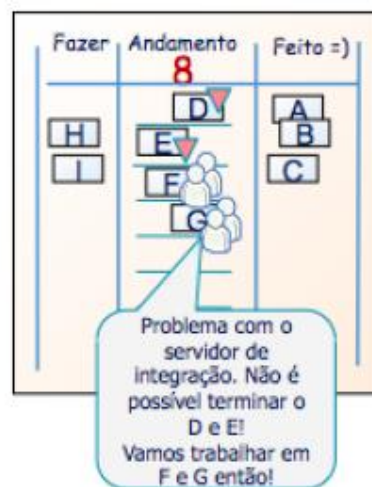
A Figura 22 aborda o seguinte caso: os funcionários ociosos não podem dar continuidade ao projeto, já que a limitação do trabalho em progresso foi apenas um, ou seja, os membros de fora do quadro *Kanban* devem esperar a atividade B ser concluída, para então iniciarem a atividade C, sendo que elas podem ser desenvolvidas em paralelo, mas por um descuido na limitação isso não é possível.

Na Figura 23 a limitação do trabalho em progresso foi aumentada para oito, como sugerido na Figura 22, apenas para verificar se com o aumento no limite (que anteriormente era 1) era possível não deixar funcionários ociosos. Este número 8 foi apenas um exemplo, podendo ser qualquer número até então que não seja 1. Porém, existem menos atividades no projeto do que a limitação imposta.



**Figura 23 - Aumentando a limitação do trabalho em progresso. (Fonte: KNIBERG e SKARIN, 2009)**

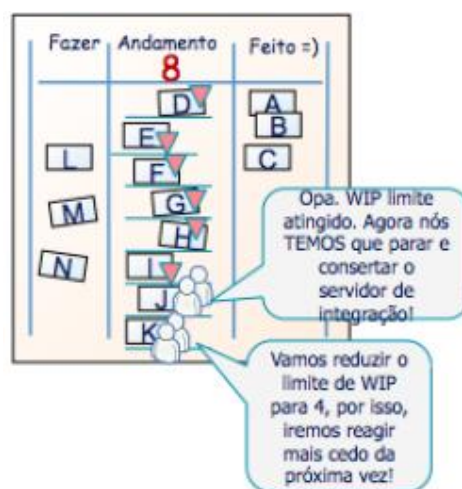
A limitação do trabalho em progresso estar superior às atividades não é problema, ou seja, é possível ter menos itens em Andamento em comparação com o limite definido. Porém, com essa limitação acima das atividades é possível que existam alguns problemas, dependendo do contexto. Um exemplo pode ser exibido na Figura 24.



**Figura 24 - Problemas com a limitação do trabalho em progresso (Fonte: KNIBERG e SKARIN, 2009)**

A Figura 24 mostra um possível problema com o servidor de integração (este servidor é apenas um exemplo de um possível problema que pode surgir nos projetos), o que impossibilita a conclusão das atividades D e E. Com isso, os membros da equipe preferiram iniciar outras atividades (F e G), ao invés de resolver o problema com o servidor de integração, já que o limite do trabalho em progresso viabiliza as quatro atividades serem desenvolvidas ao mesmo tempo. Portanto, o limite definido possibilita iniciar oito atividades mesmo que algumas delas estejam paradas por algum problema.

Na Figura 25, as atividades D, E, F, G, H e I estão com os problemas relacionados ao servidor de integração, e como visto na Figura 24, os membros preferiram iniciar outras atividades, até que o limite seja alcançado, ao invés de resolver o problema das atividades já em andamento.



**Figura 25 - Identificação do problema (Fonte: KNIBERG e SKARIN, 2009)**

Portanto, como exibido na Figura 25, o limite do trabalho em progresso foi atingido, ou seja, os membros das equipes iniciaram as oito atividades permitidas. Porém, das oito, seis estão paradas por conta de um problema com o servidor de integração. E até então, esse problema não foi resolvido porque o limite possibilitava iniciar outra atividade. No entanto este limite foi atingido e nenhuma outra atividade pode ser iniciada, restando aos membros das equipes corrigirem o tal problema com o servidor de integração e finalmente, concluírem as atividades em andamento.

O último par na Figura 25 percebeu ainda a necessidade de reduzir o limite do trabalho em progresso de oito para quatro, porque assim será possível resolver possíveis problemas mais rapidamente, sem acumular atividades em andamento.

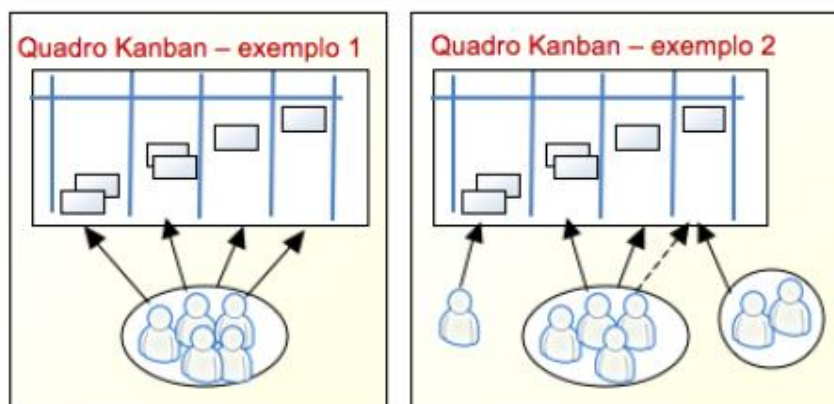
A Figura 26 mostra essa redução do limite do trabalho em progresso para quatro:



**Figura 26 - Limite do trabalho em progresso alterado (Fonte: KNIBERG e SKARIN, 2009)**

Com o limite quatro, era possível que as equipes reagissem mais rapidamente na resolução de problemas. Ou seja, existe a necessidade de equilibrar o tempo médio de execução das atividades.

O *Kanban* possibilita que equipes possam ter várias atividades ao mesmo tempo, ou que atividades possam pertencer a várias equipes, como exibido na Figura 27:



**Figura 27 - Equipes e Atividades no Kanban (Fonte: KNIBERG e SKARIN, 2009)**

O exemplo 1 na Figura 27 mostra o quadro *Kanban* e suas atividades desenvolvidas por uma única equipe, ou seja, uma equipe para executar todas as atividades. Por outro lado, o exemplo 2 ilustra várias equipes executando suas atividades independentemente de outras equipes; apenas uma equipe desenvolvendo mais de uma atividade em paralelo; e, uma atividade sendo executada por mais de uma equipe.

Estes fluxos conforme Kniberg e Skarin (2010) podem ser definidos de acordo com o fluxo de trabalho, onde as regras estabelecidas irão variar de acordo com o quadro *Kanban* de atividades.



O *Kanban* como supracitado é um sistema puxado que corresponde ao modo de gerência *Lean*. Esse modo puxado é conforme o andamento do projeto, ou seja, as equipes de desenvolvimento só irão iniciar uma atividade quando estiverem prontos.

Para iniciar um quadro *Kanban* é necessário levantar algumas questões, como (KNIBERG e SKARIN, 2010):

- Que colunas são necessárias?

As colunas representam os fluxos de trabalho, e conforme o necessário é possível adicionar ou remover colunas.

- Quais devem ser os limites do *Kanban*?

Os limites devem ser impostos de acordo com as atividades e o tempo de desenvolvê-las. O limite de *Kanban* baixo colabora com a má produtividade, onde pode haver membros da equipe ociosos. O limite de *Kanban* alto pode significar tarefas inativas, onde o tempo imposto para execução de atividades não é o suficiente.

- Quão estritos são os limites de *Kanban*?

Isso dependerá das estratégias de trabalho definidas pelos gestores, podendo ser estritas, ou seja, não pode haver quebra de regras, ou pode ser uma forma de diretriz, onde é permitido quebrar regras conforme o motivo para tal finalidade.

O *Kanban* lida com um melhor controle visual dos projetos e possui algumas características, são elas:

- Ele é ágil;
  - O *Kanban* é uma ferramenta utilizada pela metodologia *Lean* e é considerada ágil por conta de seus fluxos e visualizações.
- Utiliza controle de cronograma;
  - O *Kanban* é composto por colunas que ilustram um cronograma sucinto do projeto como um todo.
- Limita as atividades em andamento;
  - As atividades em andamento são limitadas de acordo com as estratégias de trabalho e com as equipes desenvolvedoras. Este limite impõe quantas atividades poderão ser realizadas simultaneamente.
- Utiliza a transparência para direcionar a melhoria de processos;

- O *Kanban* possui como objetivo principal o melhor controle visual de atividades e com isso, é possível observar possíveis problemas e melhorias.
- Concentra-se na entrega de *software* funcionando, de forma rápida e freqüente;
  - O *Kanban* quando utilizado em conjunto com ferramentas como a *Kaizen* e *Just-in-time* possibilita um maior *feedback* e aplicação de recursos.
- Utiliza equipes auto-organizáveis;
  - As equipes *Kanban* podem variar de acordo com a atividade, podendo uma equipe ou mais desenvolverem uma mesma atividade, ou uma equipe desenvolver mais de uma atividade simultaneamente.
- Todo o trabalho e as atividades são divididos em partes menores;
  - O *Kanban* parte do princípio em que menos é mais e que a partir do momento que se divide o problema em partes menores é mais fácil solucionar problemas e garantir a qualidade do produto final.
- O planejamento é continuamente otimizado.
  - O quadro *Kanban* vive em constante mudança, já que todas as vezes que as atividades alteram de status os cartões são movidos de lugar, possibilitando uma melhor visão de problemas e otimizações.

Um aspecto interessante ao utilizar o *Kanban* é que ele fornece uma evolução gradual do processo cascata para o modelo de desenvolvimento ágil de *software*. A Tabela 4 apresenta algumas das características peculiares do *Kanban*.

**Tabela 4 - Características específicas do *Kanban***

Item	Observação
<b>Iterações <i>Timeboxed</i> opcionais</b>	Pode ter cadência separada para o planejamento, <i>release</i> e melhoria de processos. Pode ser orientada para eventos, em vez de <i>Timeboxed</i> .
<b>Compromisso</b>	Os compromissos no <i>Kanban</i> são opcionais, possibilitando aguardar os melhores momentos para tomar decisões e executar compromissos que necessitam de uma responsabilidade maior.

<b>Lead Time</b>	Utiliza o <i>Lead Time</i> como padrão métrico para o planejamento e melhoria de processos.
<b>Equipes</b>	Equipes multifuncionais opcionais; Equipes de Especialistas autorizadas.
<b>Itens</b>	Nenhum tamanho especial de item é prescrito.
<b>Gráficos e Diagramas</b>	Nenhum tipo específico de diagrama é prescrito.
<b>Limitação do trabalho em progresso</b>	É definido de acordo com a situação do fluxo de trabalho.
<b>Estimativas</b>	As estimativas são opcionais.
<b>Capacidade</b>	Pode adicionar novos itens sempre que houver capacidade disponível.
<b>Quadros</b>	Quadros <i>Kanban</i> podem ser compartilhados por várias equipes ou individualmente.
<b>Papéis</b>	Não discrimina nenhum papel.
<b>Visualização</b>	Um quadro <i>Kanban</i> contínuo.
<b>Priorização</b>	As priorizações das atividades podem ser opcionais.

O *Kanban* foi desenvolvido embasado e focado no controle visual ao longo das etapas de fabricação dos produtos. Deste modo, é provável identificar mais facilmente e mais rapidamente possíveis irregularidades e suas correções imediatamente (RIBEIRO, 1989).

Cada organização possui uma maneira específica de utilizar o *Kanban*, segundo Ribeiro (1989), “algumas fábricas possuem painéis luminosos instalados nas seções de trabalho, onde uma luz verde significa trabalho transcorrendo corretamente, a luz amarela significa atraso em relação ao programado e a luz vermelha indica problema com a operação interrompida”.

Existe, também, a utilização do quadro branco, onde são fixados pequenas partes do processo desenvolvido e seu funcionamento se dá por filas. Estas filas constata as datas, *status* e as atividades, isso dependerá da aplicação da empresa em relação ao *Kanban*.

Segundo Oliveira (2008) e Cunha (2009), o *Kanban* possui as seguintes vantagens:

- Redução de custos: A necessidade de reduzir custos pode ser um reflexo de uma gerência ineficaz. Uma forma de redução de custos é ter um melhor controle visual do que é negociado e os procedimentos aplicados no projeto, como a telefonia e informática.
- O número de cartões *Kanban* em circulação limita o estoque máximo: No sistema *Kanban*, devido à redução dos estoques, os problemas tendem a aparecer.
- Os setores produtivos são mais bem aproveitados: O *Kanban* é um método motivacional, aproveitando sempre da melhor forma possível o que gera valor ao produto final, identificando e valorizando tais setores.
- Identificação rápida das flutuações da demanda e resposta rápida: Todo o processo *Kanban* é baseado na demanda real do projeto.
- A eficiência do sistema é a medida pela diminuição do número de cartões *Kanban* em circulação: Existem o(s) cartão(ões) que auxiliam no controle de eficácia e ineficácia do projeto.
- As necessidades de reposição são identificadas visualmente: Por ter o controle visual de tudo o que acontece com o projeto, é fácil identificar benfeitorias e necessidades de reposição.
- Não há programação de produção para os itens controlados pelo *Kanban*: Não se programa a produção, a mesma é produzida de acordo com o que é visualizado pelo *Kanban*, levando em consideração a necessidade e o cronograma.
- Trata de um item isoladamente: Esta vantagem do *Kanban* leva em consideração o autocontrole que esta técnica impõe no desenvolvimento do projeto.
- Repõe a quantidade consumida: De acordo com o que é visualizado nos cartões *Kanban*, o estoque é repostado. Se for visto que o estoque ainda está contendo itens necessários em uma boa quantidade, não é repostado mais material no estoque, evitando desperdícios.
- Simplicidade: O *Kanban* é uma técnica simples de controle visual onde o foco principal é observar todas as atividades desenvolvidas.

Em contrapartida, algumas das desvantagens presentes no sistema *Kanban* são, segundo Oliveira (2008) e Cunha (2009):

- Demanda instável: A demanda é influenciada pelos clientes e é controlada pelos cartões *Kanban*.
- Emergência afeta o funcionamento do sistema: Quando existe a necessidade de mudar algo rapidamente, são necessários mudar, muitas das vezes, todos os cartões *Kanban* de posição e ordem.
- Indisciplina operacional: Falta de comprometimento dos operários.
- Não integra informações do futuro: O *Kanban* se preocupa com o controle visual do momento, apenas com as atividades desenvolvidas naquele momento, sem preocupações com o futuro do projeto (diretamente).
- Utiliza cobertura vertical: Quando o consumo for zero, o estoque será reabastecido até o limite indicado.
- Flexibilidade limitada ao dimensionamento de estoque: Só é inserida no estoque a quantidade que realmente foi retirada.

O *Kanban* se divide em um sistema *kanban* e um método *Kanban*, conforme exibido na Figura 28:



**Figura 28 - Sistema kanban e Método Kanban**

A Figura 28 apresentou duas distinções do *Kanban*, o sistema e o método. O primeiro é basicamente um sistema puxado com limites na geração de valor, já o método *Kanban* funciona com o sistema *kanban* focado em transição, *kaizen* e gestão. O método *Kanban* possui os seguintes princípios:

- Comece com o que você já faz hoje;

- Concorde em buscar uma abordagem evolucionária para mudança;
- Inicialmente respeite papéis, responsabilidades e cargos estabelecidos.

É necessário que ao implantar o método *Kanban* esteja sempre concordando com uma melhoria contínua, ou seja, a partir do estado atual é necessário buscar mudanças evolucionárias.

A Tabela 5 exibe algumas das características contidas nas metodologias *Lean* e *Kanban*:

**Tabela 5 - *Lean* e *Kanban***

<i>Lean</i>	<i>Kanban</i>
Tem como filosofia a busca pela perfeição.	Implementa o <i>feedback</i> organizacional.
Pensamento sistêmico.	É projetado um sistema que se permita trabalhar bem.
Pensadores <i>Lean</i> acreditam na construção de sistemas e que há classes de pessoas que operam estes sistemas.	Equilibra a demanda com a capacidade da equipe.
Possui áreas de aplicação, como as atividades que agregam valor, interfaces entre operações e o fluxo de atividades.	Sistema puxado que contribui para limitar o trabalho em progresso e adiar o comprometimento até o último momento responsável.

O *Kanban*, assim como toda metodologia, recebe algumas características que não são reais, os chamados mitos. Estes mitos surgem muitas vezes com o desconhecimento sobre as metodologias, gerando situações surreais. Serão descritos os mitos e verdades encontradas no *Kanban*, como por exemplo, o *Kanban* não é um processo com iterações, na verdade, a utilização de iterações no *Kanban* é opcional, ou seja, este recurso pode ser utilizado quando julgado necessário, conforme o contexto.

Outro mito descoberto foi que o *Kanban* não utiliza estimativas, este mito, assim como o anterior, não é real pelo fato de que as estimativas no *Kanban* também são opcionais. Um mito extremamente longe da realidade é citar que o *Kanban* é melhor que Scrum, XP, RUP e demais metodologias, já que o *Kanban* é apenas mais uma ferramenta do processo, logo, não dá para presumir que ela é melhor que as demais ferramentas e/ou metodologias.

Por fim, o mito que cita o *Kanban* como modo de substituição das metodologias tradicionais ágeis, sendo que a metodologia *Kanban* é apenas um recurso que interfere sobre o gerenciamento do fluxo de trabalho, portanto, sua proposta não é substituir nenhuma ferramenta, e sim, implementar os conceitos de mudança de unidade, aplicando o modelo visualização, limites do trabalho em progresso e evolução de resultados.

De acordo com as metodologias abordadas até o momento, na Tabela 6 são apresentadas as correlações e disparidades entre elas. As metodologias são: RUP, *Scrum*, *Lean* e *Kanban*.

**Tabela 6 - Correlação e disparidade entre as metodologias**

<b>RUP</b>	<b>Scrum</b>	<b>Lean</b>	<b>Kanban</b>
Mais de 30 papéis	Planejamento	Ágil	Quadro de Cartões
20 atividades	Revisão	Eliminação do desperdício	Ciclo de tempo
70 artefatos	Retrospectiva	Integração da qualidade	Controle do <i>work in progress</i>
Complicado aplicar na prática	Reunião diária	Conhecimento	Aberto a tudo
Mais prescritivo	<i>Product Backlog</i>	Entregas rápidas	Limite o trabalho em progresso
Concepção; Elaboração; Construção e Transição	<i>Sprint Backlog</i>	Otimização do todo	Visualize o <i>workflow</i>
Iterativo	<i>Burndown charts</i>	Respeito às equipes	Mais adaptativo
Flexível	Adaptabilidade mediana	Adiar decisões desnecessárias	Simples

A Tabela 6 exibiu algumas características em comuns e distintas entre algumas metodologias. Vale ressaltar que as mesmas podem ser utilizadas em conjunto ou separadamente, isso será definido de acordo com as estratégias de trabalho do grupo desenvolvedor dos projetos.

As ferramentas podem ser trabalhadas em conjunto e/ou separadamente, na próxima seção será abordada a ferramenta *Kaizen*, comumente utilizada em conjunto com o *Kanban*, já que as duas ferramentas possuem características em comum e se completam, tornando a visualização dos projetos ainda melhor e eficaz.

### 2.4.2. *Kaizen*

O criador desta técnica, Ohno (1997), afirma que ela é uma metodologia baseada na identificação contínua de oportunidades de melhoria, na sua análise e implementação imediata em toda a organização. O que difere esta metodologia das demais, é que para ela, conforme Gaião (2012), “o ser humano é visto como o bem mais valioso das organizações”, o que leva aos colaboradores uma motivação maior e auxilia para que os mesmos estejam sempre motivados e em constante melhoria pessoal e profissional dentro das empresas.

Os autores Chen e Shady (2008) afirmam que as tarefas seguem três categorias, são elas: “trabalho que acrescenta valor; trabalho que não acrescenta valor, mas que é necessário executar; e, desperdício”.

Para identificar a necessidade da aplicação da metodologia *Kaizen* é importante utilizar o mapeamento do fluxo de valor, pois através deste fluxo é possível encontrar as fontes de desperdícios (CICONELLI, 2007). O sistema Toyota de Produção (*Lean Manufacturing*) também utilizou os princípios *Kaizen*. A Figura 29 ilustra alguns passos chaves para a implementação dessa ferramenta.





**Figura 29 - Kaizen**

Esta metodologia possui dez mandamentos de boas práticas para colaborar com o sucesso dos projetos desenvolvidos pelas empresas, são eles (GAIÃO, 2012):

- Não procure desculpas e justificativas;
- Não se preocupe antecipadamente;
- Discuta os problemas sob fatos concretos;
- Aproveite os problemas para encontrar ideias melhores;
- Não queira 100% de perfeição;
- Faça as coisas acontecerem imediatamente;
- Use o raciocínio;
- Transforme o desnecessário em produtivo;
- Destrua os mitos;
- *Kaizen* é infinito.

*Kaizen* é utilizado na metodologia *Lean* para indicar mudanças de melhoria menores e contínuas. *Kaizen* tem sido mais bem aceito pelos gestores por sua facilidade de implementação. É possível fazer inúmeras analogias com a metodologia *Kaizen*, já que se trata de uma mudança para melhor, independente do contexto.

Conforme o autor Mariotti (2010),

Na abordagem *Kanban*, aplicando a cultura *Kaizen*, antes de mudar qualquer coisa devemos entender o ambiente de trabalho. Se filas, bloqueios, gargalos ou problemas entre áreas estão nos prejudicando, primeiro, vamos visualizar isso, convencer o grupo dos problemas e usar o *Kaizen* para a melhoria do ambiente com pequenas mudanças incrementais e constantes.

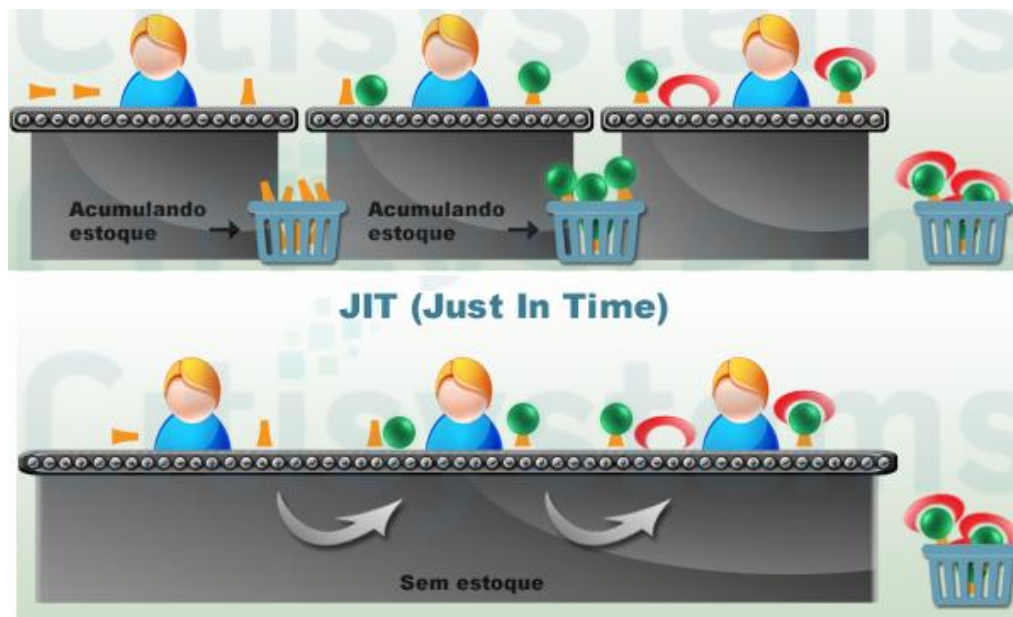
Um dos capacitadores do princípio Busca da Perfeição da metodologia *Lean* é o *Kaizen*, já que para o *Kaizen* é sempre possível fazer o melhor, nenhum dia deve passar sem que alguma melhoria tenha sido implantada, seja ela na estrutura da empresa ou no indivíduo.

Como supracitado, as ferramentas podem trabalhar em conjunto e na próxima seção será abordada a ferramenta *Just-in-time* que é comumente utilizada em conjunto com o *Kanban* e *Kaizen*.

### **2.4.3. *Just-in-time***

Esta técnica significa que devem ser produzidos apenas os produtos necessários na quantidade certa, num período de tempo adequado. Amasaka (2007) afirma que tendo isto como premissa “o resultado é que os excessos nos estoques e mão-de-obra serão naturalmente diminuídos, conseguindo-se assim, uma redução dos custos de produção e um aumento na capacidade produtiva”.

No entanto, apesar da redução de custos ser um fator primordial no sistema de produção, Pinto (2009) afirma que existem outros três objetivos secundários que auxiliam ao objetivo final, são eles: “controle da qualidade; garantia da qualidade; e, respeito pelo trabalhador”. A Figura 30 ilustra o processo *Just-in-time*, onde é essencial o não acúmulo de estoque.



**Figura 30 - Processo Just-in-time (Fonte: SILVEIRA, 2012)**

O conceito da produção *Just-in-time* assenta no sistema *pull* deixando que seja o cliente a puxar a produção. Assim sendo, os processos são realizados nos momentos certos, na quantidade pedida e no local combinado, “recorrendo ao *Kanban* para controlar e disciplinar o fluxo de materiais, pessoas e informação (JUNIOR e FILHO, 2010)”.

Segundo Holweg (2007), “a implementação do sistema *Just-in-time* ajuda a reduzir a produção excessiva, o tempo de espera, o transporte, o tempo de ciclo, os níveis de estoques, a movimentação excessiva dos recursos e o número de produtos defeituosos”.

A metodologia *Lean* possui o princípio: determinar valor para o cliente, identificando a cadeia de valor e eliminando desperdícios, e para isso, o capacitador é o *Just-in-time*, como forma de fornecimento de produtos e serviços. Este processo evita o acúmulo de estoque ao longo do processo de desenvolvimento, como se faz no ciclo de vida em cascata. Nesse caso, o *software* em estoque são todos os requisitos que ainda não foram liberados para o cliente usar (MARIOTTI, 2010).

Por reduzir custos e agilizar processos, o *Just-in-time* contribui para otimizar espaço e tempo. Como vantagens ao utilizar esta ferramenta podem ser relacionadas:

- Rápida conversão dos materiais;
- Redução do trabalho em processo;
- Redução do espaço e manuseio;
- Rápida resposta aos problemas;

- Redução de agendamentos e rastreamentos;
- Maior responsabilidade;
- Melhor qualidade;
- Menos desperdícios e retrabalhos;
- Melhor resposta ao mercado;
- Melhoria de atitude.

Em contrapartida, o *Just-in-time* inviabiliza sua utilização em produtos com demanda pouco previsível e com grandes oscilações. E sua prática se torna viável em um número menor de fornecedores, nos quais tenham estabilidade no fornecimento dos materiais necessários.

A Tabela 7 mostra algumas distinções entre as ferramentas *Kanban* e *Just-in-time*.

**Tabela 7 - *Kanban* e *Just-in-time***

<b><i>Kanban</i></b>	<b><i>Just-in-time</i></b>
<b>Ferramenta de trabalho utilizada para o controle de produção.</b>	Técnica de gestão de produção.
<b>Utiliza cartões que controlam o estoque e a produção de forma visual.</b>	Envolve conceitos de administração e produtividade.
<b>Colabora com a eliminação ou redução de desperdícios.</b>	Aumenta a satisfação do cliente através de um desempenho de entrega muito superior e com menores custos.

Mesmo com algumas disparidades, as ferramentas *Kanban* e *Just-in-time*, na prática, trabalham em conjunto, uma complementando a outra, já que o *Kanban* é uma ferramenta de redução e eliminação de desperdícios, o que contribui para que os processos atinjam um nível de competitividade maior, seguindo o conceito *Just-in-time*, onde existem entregas na medida das necessidades dos clientes.

Nesta seção foram apresentados conceitos de Gerência de Projetos, que possibilitou o estudo das Metodologias de Gerenciamento de Projetos, como o RUP, Modelo em Cascata, *Scrum* e *Lean de Software*. Esta última possui várias abordagens, sendo descritas suas

filosofias, práticas e terminologias. A metodologia *Lean* possui ferramentas que possibilitam e facilitam sua prática, sendo os utensílios apresentados o *Kanban*, o *Kaizen* e o *Just-in-time*. Nas próximas seções serão abordados os materiais e métodos que auxiliaram o desenvolvimento do trabalho e os resultados obtidos de acordo com a proposta definida.

### **3. MATERIAIS E MÉTODOS**

Nesta seção serão apresentados os recursos e métodos utilizados durante o desenvolvimento deste trabalho.

#### **3.1. Local e Período**

O desenvolvimento deste trabalho deu-se nas dependências do complexo de informática do CEULP/ULBRA, bem como em residência própria, o primeiro localizado na cidade de Palmas e o segundo na cidade de Paraíso do Tocantins. O período de desenvolvimento deste trabalho ocorreu durante o primeiro e o segundo semestre de 2013, como parte das disciplinas “Trabalho de Conclusão de Curso I” e “Trabalho de Conclusão de Curso II”.

#### **3.2. Materiais**

Para que fosse possível a realização deste trabalho foram utilizados recursos próprios e outros adquiridos através da internet e livros. Dentre os materiais utilizados no referencial teórico e nos resultados obtidos estão: artigos; tutoriais; trabalhos de conclusão de curso; resumos; imagens; dissertações de mestrado e doutorado; e, livros que abordam os assuntos da proposta.

#### **3.3. Métodos**

A metodologia deste trabalho é composta por seis fases distintas e está relacionada à execução das etapas apresentadas na Figura 31.



**Figura 31- Metodologia do Projeto**

A primeira parte do projeto consistiu na coleta de material sobre os temas propostos (Passo 1). A seqüência de estudos foi iniciada com a elaboração do referencial teórico baseado nos materiais coletados, isto se fez necessário para se obter um entendimento geral do contexto do trabalho (Passo 2). Os conceitos estudados foram: Projetos e Gerência de Projetos; Metodologia *Lean* e Metodologia *Kanban*; *Kaizen* e *Just-in-time*.

A metodologia *Lean* possui diversas abordagens, que vão desde o foco na manufatura de negócios até a implementação de projetos de *software*. O Passo 3 consistiu na análise destas abordagens. O Passo 4 envolveu o estudo sobre a metodologia *Kanban*, suas características e modelos de aplicação. Dentre as abordagens da metodologia *Lean*, foi dado o foco na metodologia *Lean Software Development* (LSD), que foi desenvolvida especificamente para os projetos de *software* (Passo 5).

A fase de elaboração do projeto ocorreu no período letivo do semestre 2013/1, sendo que os primeiros procedimentos consistiram em um estudo conjunto de dois temas propostos no trabalho, *Lean* e *Kanban*. Durante essa fase, foram relacionadas as abordagens das duas técnicas voltadas para computação. Posteriormente, foi formulada uma proposta de utilização das técnicas *Lean* e *Kanban* em conjunto (Passo 6) para uma melhoria no controle e gerenciamento de atividades dos projetos de *software*.

O conhecimento adquirido a partir dos estudos realizados permitiu realizar uma relação entre os temas, através da intersecção de seus processos, papéis e artefatos. Esta relação foi mensurada após a elaboração da proposta e com isso, foi possível fazer a

verificação das chances das técnicas serem utilizadas em conjunto. A segunda fase do projeto, que é referente à elaboração da metodologia de gerência de projetos, foi realizada no segundo semestre letivo do ano de 2013.

Esta fase de elaboração da proposta consistiu em: representação gráfica do processo das duas abordagens, com modificações nas duas visando a adequação de ambas; criação de uma representação gráfica única para a proposta; explicação de cada item da proposta; definição textual de papéis, atividades, artefatos, princípios, métodos e processos. Estes itens foram representados através de um *workflow* (fluxo de trabalho) a partir das representações gráficas específicas, podendo ser, por exemplo, bonecos e quadros para papéis e processos, respectivamente.

É importante ressaltar que o *Kanban* foi escolhido por conta da rápida eliminação de questões que prejudicam o desempenho dos projetos, já que o mesmo atua fornecendo visibilidade nos processos, deixando explícitos os problemas e prendendo o foco das equipes no objetivo final com agregação de qualidade. Ao aplicar o *Kanban* e respeitando suas exigências, o sistema tende a contribuir para a maturidade da equipe, podendo até afetar na cultura organizacional das empresas. Já a metodologia *Lean* foi escolhida porque agrega valores ao produto final, elimina desperdícios e aumenta a rentabilidade dos negócios. A mesma possui princípios e técnicas para melhor direcionamento de recursos e foca totalmente na gerência de *software*.

Na próxima seção serão abordados os resultados atingidos de acordo com a proposta inicial do trabalho e conseqüentemente, as considerações finais sobre tudo que foi desenvolvido para a proposta da metodologia de gerenciamento de *software*.



## 4. RESULTADOS E DISCUSSÃO

A partir da proposta de utilização do *Kanban* e *Lean* em conjunto para melhoria do gerenciamento e monitoramento de *software*, foram definidos os princípios, os componentes essenciais, as etapas e processos, os papéis, os artefatos e os itens necessários para o entendimento do projeto. É importante ressaltar que o objetivo do presente trabalho é fazer a união do *Kanban* e *Lean* através da intersecção dos conceitos supracitados, fazendo a adequação de processos, eliminação ou adição de novos itens.

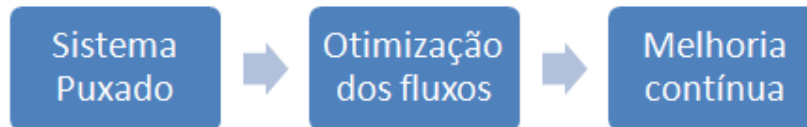
Os princípios são filosofias e estratégias para aumentar a satisfação do cliente, através de custos menores e melhor direcionamento de recursos. Os componentes essenciais são aqueles itens que permitem a análise e melhoria dos fluxos de trabalho, por meio do envolvimento de pessoas qualificadas e com a execução das práticas pré-definidas. As etapas e processos formam a metodologia proposta, ou seja, as etapas representam cada passo a ser implantado no projeto e os processos formam a etapa. Os papéis representam as responsabilidades dos envolvidos no projeto. E, por fim, os artefatos representam todo o auxílio ou o resultado gerado conforme a execução de uma atividade.

As seções a seguir apresentam os passos produzidos durante o desenvolvimento do projeto, assim como os resultados obtidos durante este desenvolvimento e os possíveis trabalhos futuros. Conforme a proposta do trabalho, que a partir de processos existentes é criada uma nova metodologia, existem um conjunto estruturado de práticas, que serão explanados a seguir.

### 4.1. Os Princípios da Metodologia

Os princípios estabelecidos para a metodologia proposta são baseados nos princípios do *Lean Thinking* (ou Mentalidade Enxuta). O *Lean Thinking* é uma filosofia estratégica de negócios que visa aumentar a satisfação dos clientes através da melhor utilização de recursos. Segundo o *Lean Institute Brasil* (2013), “A gestão *lean* procura fornecer, de forma consistente, valor aos clientes com os custos mais baixos (propósito), identificando e sustentando melhorias nos fluxos de valor primários e secundários (processos), por meio do envolvimento de pessoas qualificadas, motivadas e com iniciativa (pessoas)”. O foco da implementação deve estar nas reais necessidades dos negócios e não na simples aplicação das ferramentas *Lean*.

As práticas da metodologia proposta envolvem criação de fluxos contínuos e sistemas puxados que, possibilitam resultados com aumento da capacidade de oferecer os produtos que os clientes querem, na hora que eles precisam, aos preços que estão dispostos a pagar, com custos menores e qualidade superior, garantindo maior rentabilidade para o negócio. Os princípios da metodologia proposta podem ser visualizados na Figura 32.



**Figura 32 - Princípios da Metodologia**

Os princípios (ou práticas) definidos para a metodologia proposta foram: Sistema Puxado; Otimização de Fluxos e Melhoria Contínua. Estas práticas foram evidenciadas para facilitar os processos de gerenciamento e monitoramento de atividades. Cada prática será abordada nos próximos itens.

- **Sistema Puxado**

Conforme apresentado no referencial teórico, este princípio é baseado no *Kanban*. O Sistema Puxado é quando se inverte o fluxo produtivo da situação cliente X empresa, ou seja, as empresas não mais empurram os produtos para o consumidor através de descontos e promoções. O consumidor passa a puxar o fluxo de valor, reduzindo a necessidade de estoques e valorizando o produto. No desenvolvimento de sistemas, o cliente é quem decide quais funcionalidades devem continuar ou não, isso é realizado conforme os *feedbacks* gerados entre cliente X desenvolvedores, pois quanto mais rápido e quanto menor os trechos apresentados ao cliente, menores os problemas e retrabalhos futuros.

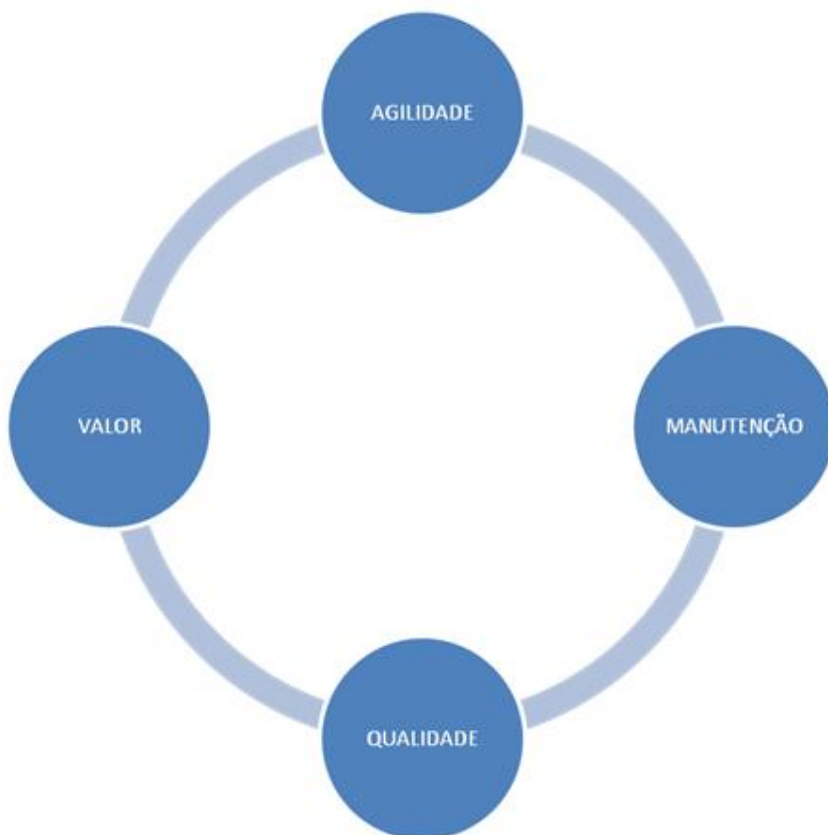
- **Fluxo Contínuo**

Este princípio é a permissão que os processos e atividades possuem para serem iniciados, ou seja, executar de forma contínua aqueles processos que de alguma forma auxiliam na geração

de valor. Este princípio envolve ter a capacidade de desenvolver, produzir e distribuir rapidamente as atividades desenvolvidas.

- **Melhoria Contínua**

Este princípio se dá pela busca contínua na direção da situação ideal. A metodologia proposta possui alguns componentes essenciais que auxiliam para uma entrega satisfatória e eficaz, são eles: agilidade; manutenção; qualidade e valor. A Figura 33 ilustra o ciclo desses componentes, que devem ser executados em paralelo e sem ordem de início e/ou fim.



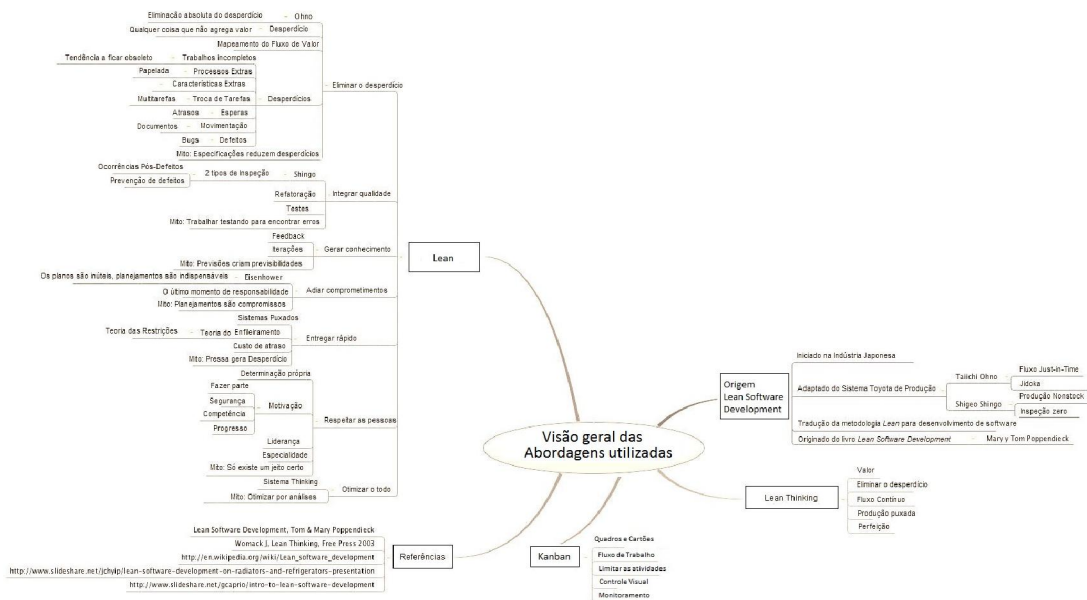
**Figura 33 - Componentes essenciais**

Conforme ilustrado pela Figura 33, os componentes essenciais da metodologia proposta são quatro, sendo eles a Agilidade, o componente que executa o processo Entrega Rápida, pois com este acontece um *feedback* e possíveis melhorias contínuas mais rapidamente. Além do processo Entrega Rápida, o processo Integrar qualidade também é ligado ao componente

Agilidade, pois agrega qualidade quando se tem respostas mais rápidas e mais curtas acerca das atividades executadas.

O componente Manutenção leva em consideração o fluxo de Revisão, onde os dados são analisados e possíveis barreiras que atrapalhavam o andamento dos processos são removidas e erros/defeitos são consertados ou removidos do sistema. A Qualidade engloba diversos processos, pois todos eles focam na entrega de valor para o cliente, e, este valor deve conter a qualidade esperada por ele. O último componente é o Valor, que é o motivo maior de todos os princípios, de todos os processos *Lean* e *Kanban*, pois tudo que é desenvolvido pela metodologia proposta foca em itens que agregam valor ao produto final, pois a qualidade e satisfação é medida pelo valor entregue ao cliente.

Dentre as abordagens existentes da metodologia *Lean*, as que foram importantes para o desenvolvimento do trabalho foram a *Lean Software Development* e a *Lean Thinking*. A primeira é a aplicação do *Lean* para o desenvolvimento de *software* e a segunda para os negócios. A Figura 34 exibe os principais itens que fazem parte destas abordagens e que auxiliaram o desenvolvimento do trabalho.



**Figura 34 - Visão geral das Abordagens utilizadas**

Conforme apresentado na Figura 34, a origem da metodologia *Lean Software Development* foi na indústria automobilística japonesa que logo em seguida, foi adaptado do sistema Toyota de Produção. Logo após essa adaptação, foi reconhecida uma abordagem que identificava os princípios *Lean* voltados para computação, surgindo a metodologia *Lean Software Development*.

Além da metodologia *Lean Software Development* outros elementos foram importantes para confecção da metodologia proposta, como os princípios *Lean Thinking*, as técnicas do *Kanban* e, princípios e práticas da metodologia *Lean*, mais precisamente da metodologia enxuta, que foca na eliminação dos desperdícios.

Um outro fator que auxiliou a origem desta abordagem foi o livro *Lean Software Development* de *Mary Tom Poppendieck*. Com isso, a metodologia *Lean Software Development* possui alguns princípios, como: Eliminar o Desperdício; Integrar Qualidade; Gerar Conhecimento; Adiar Comprometimentos; Entregar Rápido; Respeitar as pessoas; e, Otimizar o todo. Cada um destes princípios possuem processos e atividades que facilitam sua implantação dentro dos projetos.

Alguns dos itens que englobam a metodologia *Lean Software Development*; o *Kanban*; o *Lean Thinking*; serviram de base para a metodologia proposta. Cada processo e etapa de desenvolvimento foi adaptado, inserido ou removido para a nova metodologia, conforme foi apresentado na Figura 34.

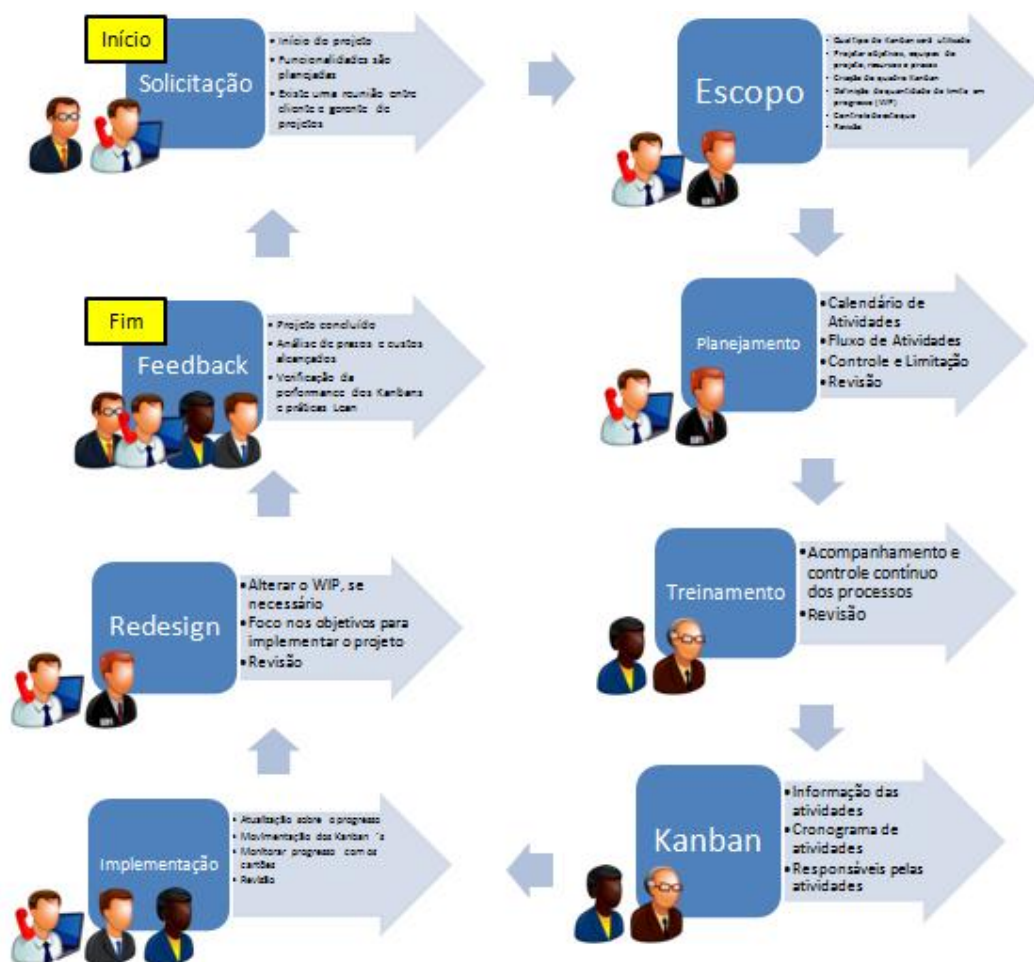
Nesta seção foram citados os princípios da metodologia proposta baseados nos princípios *Lean Thinking* e na metodologia *Lean Software Development*. Além disso, foi apresentada uma visão geral do que engloba a metodologia proposta. Já a próxima seção aborda os fluxos e processos desenvolvidos na metodologia proposta, que englobam o *Lean* e o *Kanban*.

#### **4.2. Etapas e Processos da Metodologia**

Com base nos estudos apresentados na Revisão de Literatura e na metodologia apresentada em Materiais e Métodos, iniciou-se o entendimento da realidade das metodologias de desenvolvimento de *software* existentes para que se planejasse a metodologia proposta, que trata-se de uma metodologia de gerenciamento de *software*.

Para que a metodologia seja implantada no gerenciamento de projetos, é necessário que se tenham princípios, práticas, processos e etapas a serem seguidos. Nesta seção serão ilustrados os processos que constituem a metodologia proposta, desde a solicitação (início) ao *feedback* (fim) do projeto.

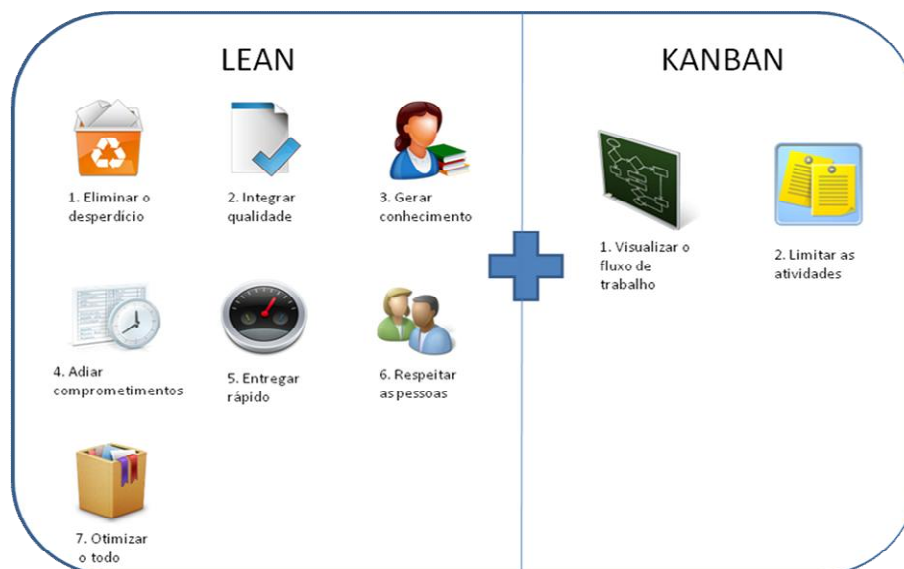
A Figura 35 apresenta as etapas e seus respectivos processos, são elas: Solicitação; Escopo; Planejamento; Implementação; *Redesign*; Treinamento; *Kanban*; *Feedback*. Estas etapas foram baseadas nas etapas contidas no *Lean Software Development* e no *Kanban*. Em cada etapa existe a figura do responsável por executar os processos descritos nas etapas, como pode ser visualizado a seguir.



**Figura 35 - Etapas e Papéis**

Conforme visualizado pela Figura 35, a metodologia proposta possui 8 etapas, que visam a implantação da metodologia nos projetos, possui 6 atores (ou papéis), que são os responsáveis em colocar em prática e executar todas as etapas, e 30 processos, que quando desenvolvidas em conjunto constroem a metodologia proposta.

A metodologia *Lean* foca em fazer o que, onde e como o cliente deseja, sem desperdícios. O *Kanban* foca na transparência e visibilidade de pequenas partes do projeto para que se possa otimizar o todo. Juntos, estes conceitos possuem processos e princípios para atingirem tais objetivos, que sempre é entregar o produto de forma satisfatória para o cliente, sem que haja desperdícios para a equipe desenvolvedora. Em conjunto, todos estes princípios colaboram para entregas rápidas, com mais qualidade e de baixo custo, tudo isso ao mesmo tempo.



**Figura 36 - Juntando os processos: *Lean* e *Kanban***

Conforme apresentado pela Figura 36, a metodologia *Lean* e a ferramenta *Kanban* possuem diversos processos. Estes processos são aplicados para que se obtenham resultados satisfatórios e melhor direcionamento de recursos, facilitando a execução de atividades e melhorando os prazos e custos dos projetos. Cada uma destas técnicas pode ser aplicada separadamente, porém, a proposta do presente trabalho foi executar todos estes processos em paralelo para um melhor monitoramento e gerenciamento das atividades contidas nos projetos.

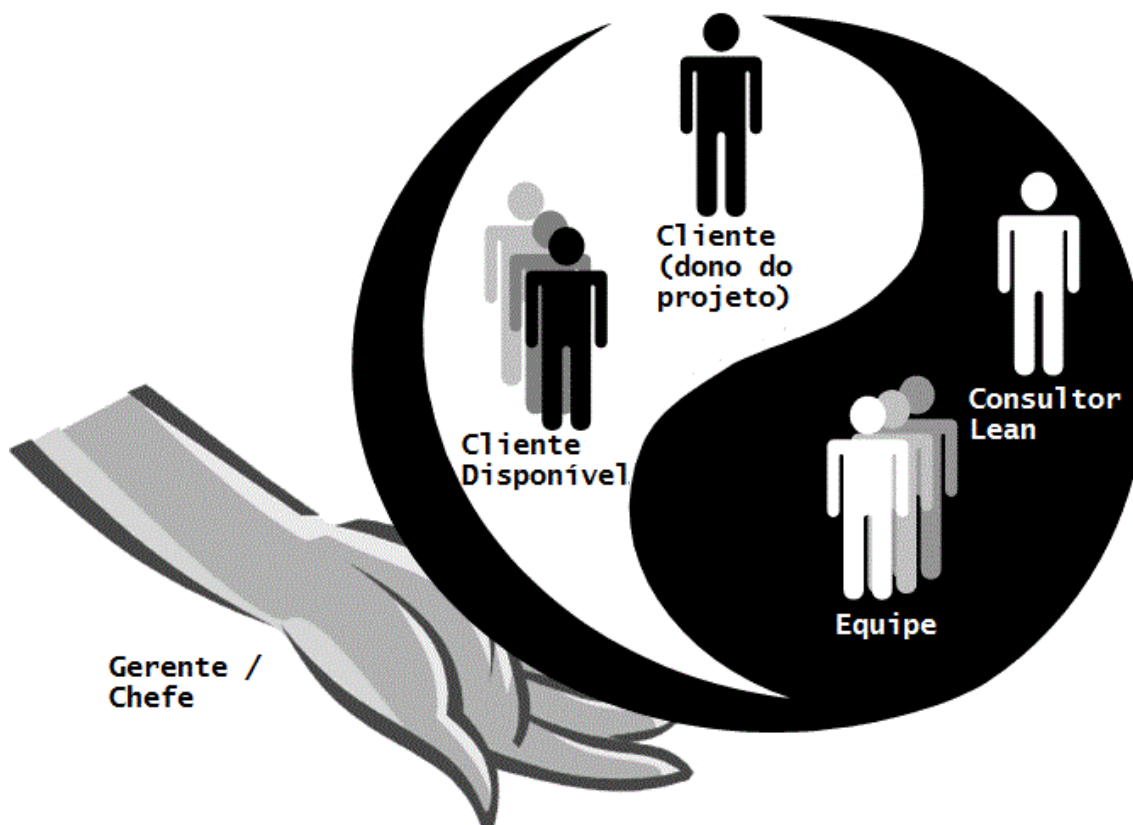
Na próxima seção, será apresentado cada um dos papéis desenvolvidos para implantar as etapas e processos da metodologia. Estes papéis foram definidos conforme a necessidade de execução, alguns foram baseados em outras metodologias existentes e outros foram adaptados para melhor adequação com o *Kanban* e *Lean*.

#### **4.3. Papéis da Metodologia**

No *Lean*, são definidos papéis e responsabilidades que cada membro da equipe deve assumir para otimizar o todo. Uma das características destes papéis, que se assemelham a papéis encontrados em várias outras metodologias de gerenciamento de projetos, é a existência de um líder para gerenciar as equipes. Nos próximos itens estes papéis serão descritos mais detalhadamente. A Figura 37 exibe os responsáveis que irão compor a metodologia proposta. Nesta ilustração, os papéis, conforme foram baseados na metodologia *Lean*, foram



acrescentados e retirados. Estas alterações foram necessárias para fazer as adequações que a nova metodologia propõe. Os papéis apresentados a seguir possuem responsabilidades de acordo com cada etapa e processos que necessitem de um ator para executá-los.



**Figura 37 - Papéis da Metodologia**

Conforme ilustrado pela Figura 37, o time da metodologia proposta é composto pelos seguintes membros: gerente/chefe; cliente disponível; cliente (dono do projeto); consultor *Lean*; e, a equipe do projeto. Cada papel será descrito mais detalhadamente a seguir.

- **Cliente**

O Cliente participa das tarefas relacionadas à definição da lista de funcionalidades do *software* que está sendo desenvolvido ou até mesmo melhorado. É ele quem elabora os requisitos e restrições que o sistema deverá ter. O papel do Cliente poderá ser de dois tipos: o cliente disponível (que é o usuário do sistema, ou seja, a pessoa que irá mexer de fato no sistema) e o cliente dono do projeto (aquele que fez a solicitação do sistema; aquele que irá

tomar as decisões mais importantes a respeito do sistema; aquele responsável pelos pagamentos; entre outros). A Figura 38 ilustra estes tipos de clientes.



**Figura 38 - Tipos de Cliente**

Os tipos de cliente diferem de acordo com as necessidades da empresa, pois o cliente dono do projeto poderá ser o analista de sistemas, o administrador da empresa, entre outros. É ele quem solicita e compra o projeto (ou sistema). Já o cliente tipo usuário é quem utiliza as funcionalidades solicitadas pelo cliente dono do projeto. Muitas vezes é o próprio cliente usuário quem solicita estas funcionalidades, isto irá variar com as estratégias de trabalho de cada organização.

- **Gerente**

O Gerente é quem toma as decisões finais, muitas vezes ele utiliza os resultados esperados por meio de gráficos, tabelas e listas, para tomada de decisão. Ele é responsável por conversar com o Cliente para decidirem o objetivo e os requisitos do sistema. Dependendo das estratégias de trabalho definidas para o projeto, o papel do Gerente é o mesmo do Chefe, chefiar decisões administrativas acerca dos membros das equipes e do projeto.

- **Equipe**

A Equipe é quem possui autoridade para executar e decidir ações rotineiras do sistema. Geralmente as equipes são compostas por vários membros (a quantidade irá variar de acordo com o projeto) e elas são responsáveis por organizar e tentar atingir os objetivos pré-estabelecidos do projeto.

- **Consultor *Lean***

O Consultor *Lean* é aquele que fez todos os treinamentos necessários acerca da metodologia *Lean*, estudou seus princípios e práticas, para poder auxiliar no desenvolvimento do projeto. Ele é responsável em aplicar a metodologia e auxiliar no seu monitoramento até o *software* ser desenvolvido. É ele quem ministra os treinamentos para as equipes desenvolvedoras dos projetos.

- **Chefe**

O Chefe é quem cuida de todo o gerenciamento do projeto e da Equipe, faz as comunicações entre os papéis existentes no projeto, garante as reuniões entre membros desenvolvedores do projeto. Ele também é responsável por auxiliar que impedimentos não atrapalhem o desenvolvimento do projeto. Dependendo das estratégias de trabalho definidas para o projeto, este papel será o mesmo exercido pelo Gerente do projeto, com funções administrativas a respeito das equipes e projeto.

A Figura 39 apresenta a legenda dos papéis que serão utilizados nas etapas desenvolvidos para a metodologia proposta. A descrição dos processos e suas aplicações para desenvolvimento de atividades serão explanadas na próxima seção.



**Figura 39 - Legenda dos Papéis**

Nesta seção foram abordados os papéis que constituem a metodologia proposta, ou seja, as funções dos responsáveis pela execução de todos os processos. A próxima seção exhibe as etapas e os processos para a metodologia proposta, sendo alguns deles existentes nas metodologias *Lean* e *Kanban* e os demais inseridos e modificados de forma a adequá-los da melhor maneira.

#### 4.4. Etapas e Processos da Metodologia

As etapas são os guias que devem ser seguidos para implantação da metodologia proposta, estas etapas são compostas por processos e papéis. Na seção 4.2. (Figura 35) foram abordadas as etapas e processos de forma geral, ilustrando a metodologia proposta como um todo, dando uma visão geral do trabalho.

Os próximos itens ilustrarão cada uma das etapas separadamente, para melhor visualização e entendimento dos processos e responsáveis. A Figura 40 apresenta a etapa de Solicitação, como pode ser visualizado a seguir.



**Figura 40 - Início do projeto**

A etapa de solicitação é baseada na metodologia *Lean Software Development*, porém foram inseridos os processos apresentados na Figura 40, mantendo apenas a questão de viabilidade do projeto. Esta viabilidade é conforme a visão do CLIENTE, ou seja, se o GERENTE acha viável a ideia da execução do projeto que o CLIENTE está solicitando. Esta etapa é realizada pelo CLIENTE e pelo GERENTE, onde ambos se reúnem para o

planejamento de funcionalidades, estudam a viabilidade do que foi sugerido com o que o cliente realmente necessita. Existe a comunicação entre eles para que se possa iniciar o projeto, este início só será autorizado caso exista um consenso e um acordo entre os atores responsáveis. Os processos contidos nesta etapa são:

- **Início do projeto**

Processo em que se dá o passo inicial, onde o que foi definido pelo CLIENTE e pelo GERENTE deve ser autorizado e repassado para os demais membros do projeto. Neste início, as informações estão sucintas e objetivas para o repasse.

- **Funcionalidades são planejadas**

Este processo, como o próprio nome já diz, é de planejamento de funcionalidades. Quando o CLIENTE solicita o projeto, ele tem em mente o que deseja, quais requisitos seriam necessários para o sistema, quais funcionalidade são importantes, entre outros. Na reunião são definidas estas funcionalidades e a melhor forma de serem implementadas. É importante ressaltar que todo o sistema deve ser desenvolvido de forma com que seus itens agreguem valor para o produto final.

- **Reunião entre cliente e gerente de projetos**

Essa reunião é solicitada pelo CLIENTE quando o mesmo deseja um sistema a ser implementado. Ela é realizada para que seja conversado com o GERENTE, o melhor para ser desenvolvido pela empresa e que o produto desenvolvido agregue valor ao CLIENTE, satisfazendo suas necessidades com melhor distribuição de recursos e eliminação de desperdícios.

A Figura 41 apresenta a etapa de Escopo. No Escopo estão contidas as informações sobre recursos que podem e devem ser utilizados durante o desenvolvimento do projeto. No Escopo da metodologia proposta estão contidos os seguintes processos: Tipo de *Kanban*; Criação do Quadro *Kanban*; WIP; Controle do Estoque; e, Revisão.



**Figura 41 - Escopo do projeto**

A fase de Escopo foi baseada no *Kanban*, acrescentando e modificando processos da metodologia *Lean Software Development*. Como apresentado pela Figura 41, a fase de Escopo possui os seguintes processos:

- **Tipo de *Kanban***

Neste processo são definidos quais tipos de *Kanban's* serão utilizados no decorrer do projeto. Estas definições poderão variar de acordo com as estratégias de cada empresa, por exemplo, existem *Kanban's* de fornecedor, *Kanban's* de requisição, entre outros (ver mais na Página 39, Tipos de cartões *Kanban*). Os *Kanban's* são cartões que contêm informações a respeito do tipo escolhido. O princípio do *Kanban* valoriza a ideia de que todos acompanham todo o processo, pois se trata de uma ferramenta de controle visual, onde tudo que é desenvolvido é descrito nos cartões para conferência de prazos, recursos, custos e responsabilidades. Os responsáveis pela definição do tipo de *Kanban* são o GERENTE e o CHEFE.

- **Projetar objetivos, equipes do projeto, recursos e prazos**

Este processo é baseado na metodologia *Lean Software Development*, o mesmo está presente na etapa de Escopo. Neste processo os objetivos são definidos como viáveis e são projetados, definindo as equipes do projeto, quais recursos estarão disponíveis durante a execução do projeto e quais prazos devem ser seguidos. Este processo é de suma importância para o

projeto como um todo, pois é ele que vai dar a base a ser seguida e respeitada. O responsável por estas definições é o GERENTE e o CHEFE.

- **Criação do Quadro *Kanban***

Este processo abrange a criação do quadro *Kanban*, ou seja, definição de colunas, de atividades, de papéis e de limites para os fluxos de trabalho. Esse processo é de suma importância no projeto como um todo, por ser o responsável pelas configurações iniciais que tendem a garantir a satisfação ao fim do produto gerado. Os responsáveis pela criação do quadro *Kanban* são o GERENTE e o CHEFE.

- **Definição de quantidade de limite em progresso (WIP)**

Este processo, como o próprio nome já diz, é a definição de quantidade de limite em progresso, também conhecido como *Work In Progress* (WIP). Na etapa de Escopo se tem apenas uma visão superficial de que será necessário se definir este limite, porém, é na fase de Planejamento que um número para este limite é definido. É importante defini-lo e editá-lo conforme for o andamento do projeto. Cabe ao GERENTE de projetos fazer esta verificação e adequação.

- **Controle de Estoque**

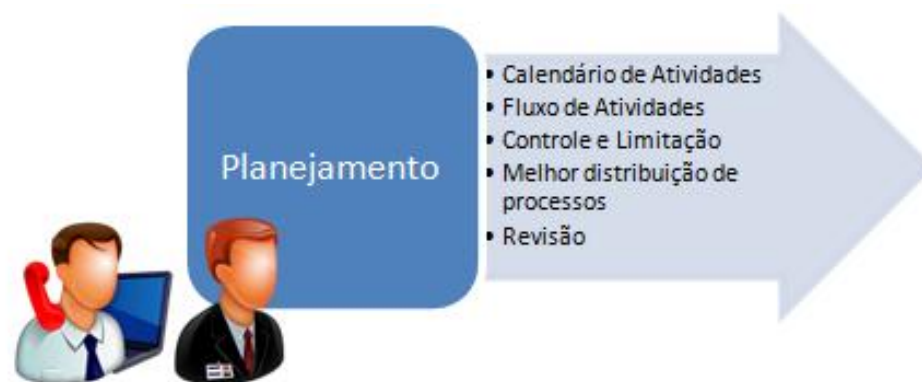
É na fase de Escopo que se têm a visão de Controle de Estoque. Este estoque é tudo que auxilia o desenvolvimento do projeto, podendo ser um código pronto desenvolvido em projetos similares, uma documentação, entre outros. É importante ressaltar que o que irá surgir de novo no projeto a ser desenvolvido seja armazenado de forma que, quando for necessário em projetos futuros se tenha uma boa base de conhecimento e experiências adquiridas. O responsável pelo controle de estoque é o CHEFE das equipes desenvolvedoras.

- **Revisão**

É na etapa de Escopo que se iniciam os processos de Revisão. Na metodologia proposta, a revisão está presente em quase todos os processos e etapas. Neste processo, é de

responsabilidade do GERENTE de projetos rever o que foi definido até agora, fazendo a verificação de viabilidade do que foi pré-definido com o que se deve ter pronto ao final do desenvolvimento.

A próxima etapa da metodologia proposta consiste em Planejamento, que é desenvolvida pelo CHEFE e pelo GERENTE de projetos. Esta etapa é de suma importância para o projeto como um todo, pois é nela que estão contidas as informações que servem de guias para o andamento do projeto. A Figura 42 apresenta esta fase, que foi baseada na etapa de Planejamento da metodologia *Lean Software Development*.



**Figura 42 - Planejamento**

A etapa de Planejamento, diferentemente da etapa de Solicitação - onde as funcionalidades são planejadas - é voltada para os especialistas, ou seja, na fase de Planejamento, o GERENTE e o CHEFE se reúnem para planejar a melhor forma para que as funcionalidades sejam implementadas, na etapa de solicitação este planejamento é feito com a visão do usuário. Nesta fase estão contidos os seguintes processos:

- **Calendário de Atividades**

O Calendário de Atividades é o planejamento realizado que possibilita prevê quanto tempo demorará para executar determinada atividade. Este processo é de suma importância devido



ao quadro *Kanban* utilizado, que possibilita o controle visual das atividades para todas as partes envolvidas no desenvolvimento. Com isso, todos os membros se tornam cientes dos prazos definidos na etapa de Planejamento e qual status - que seriam as colunas do quadro *Kanban* - a atividade deveria estar. Este calendário se torna personalizável de acordo com a quantidade de envolvidos para execução do projeto e de acordo com a quantidade de atividades, de modo a equilibrar estas execuções, para que nenhuma equipe se torne ociosa ou sobrecarregada em relação às demais.

- **Fluxo de Atividades**

Este processo é referente ao fluxo de atividades que o projeto contém. Quando é utilizado o quadro *Kanban*, é necessário que se defina uma quantidade para este fluxo, ou seja, esta quantidade significa quantas atividades poderão ser executadas em paralelo. O cálculo da melhor definição para o fluxo de atividades irá variar de acordo com o projeto, podendo ser, por exemplo, oito o número do fluxo de atividades, que significa que oito atividades poderão ser iniciadas simultaneamente, sem a necessidade da conclusão de outra atividade, gerando atrasos e conseqüentemente, desperdícios para o projeto. O responsável em planejar o melhor número para o fluxo de atividades é o CHEFE. É importante ressaltar que este número poderá ser alterado conforme a necessidade e o calendário de atividades (processo anterior).

- **Controle e Limitação**

Este processo é referente ao controle visual que se tem quando aplicado o *Kanban* em alguns dos processos da metodologia *Lean* e da metodologia proposta. Este controle é conforme os dados definidos para prazos e custos e como andam estes dados na execução das atividades. Cabe ao GERENTE de projetos fazer este controle juntamente com o quadro *Kanban* e os envolvidos no desenvolvimento do projeto. Este processo está ligado com os processos anteriores da etapa de Planejamento, já que possibilita a limitação das atividades e prazos.

- **Melhor distribuição de processos**

A melhor distribuição de processos garante que eles sejam interligados de forma complementar, ou seja, que cada processo seja desenvolvido em paralelo, interligando e completando outro processo. Os processos distribuídos para esta etapa foram:

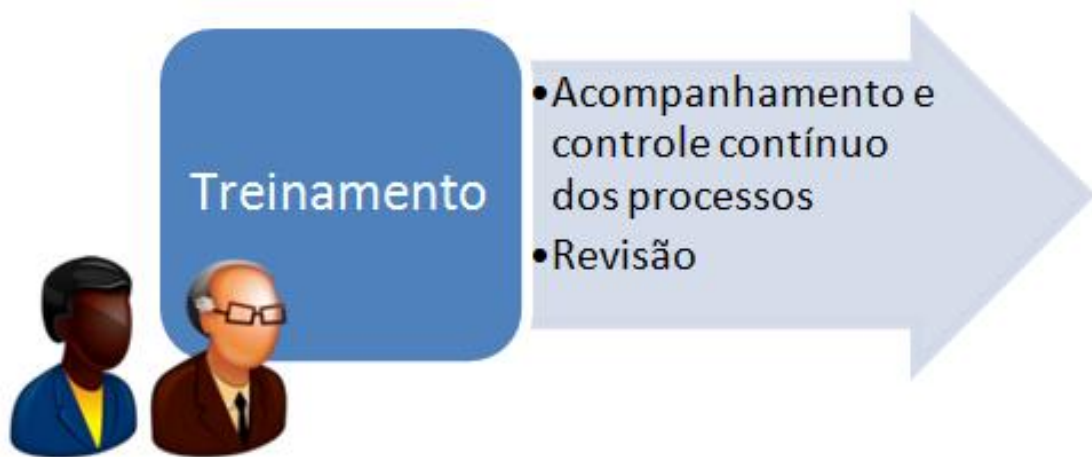
- ✓ **Adiar Comprometimentos**
- ✓ **Limitação de Atividades**
- ✓ **Entregas Rápidas**
- ✓ **Integração de Qualidade**

Estes processos garantem um melhor direcionamento de atividades e responsabilidades. Com isso, através da limitação de atividades, as entregas serão mais freqüentes e mais rápidas, integrando qualidade e agregando valor ao produto final, já que possíveis barreiras serão adiadas conforme a base de conhecimento acerca daquele problema se torne maior. Estes itens farão com que a metodologia proposta se adéque ao controle visual oferecido pelo *Kanban*, para um melhor monitoramento do que foi planejado pela metodologia *Lean*. Os responsáveis por este processo são o GERENTE e o CHEFE.

- **Revisão**

O processo de revisão não mais é executado a partir do que foi implementado, ou seja, na metodologia *Lean* o processo de revisão é único e só acontece após a implementação do que foi planejado. Esta teoria está ultrapassada nas metodologias atuais, já que existem testes de verificação e validação que auxiliam a revisão do que foi executado, para evitar o retrabalho e desperdício de se revisar somente após a execução de alguma atividade ou etapa do fluxo. Na metodologia proposta, este processo de Revisão está presente em quase todas as etapas, possibilitando um planejamento mais eficiente e mais colaborativo com os demais processos.

A próxima etapa da metodologia proposta é a etapa de Treinamento, na qual os FUNCIONÁRIOS são treinados por um TREINADOR/ESPECIALISTA da metodologia *Lean*. A Figura 43 apresenta esta etapa e seus respectivos processos. Vale ressaltar que esta etapa foi baseada na metodologia *Lean Software Development*, porém, modificando seus processos para melhor adequá-los na metodologia proposta.



**Figura 43 - Treinamento Lean**

Esta é uma das etapas primordiais da metodologia proposta, onde os FUNCIONÁRIOS são treinados e se tornam cientes dos princípios e práticas da metodologia *Lean*, aprendendo sobre a eliminação dos desperdícios - tudo aquilo que não agrega valor - e o melhor direcionamento de recursos. Nesta etapa é contratado um ESPECIALISTA da metodologia *Lean* para fornecer o treinamento e o suporte necessário durante o desenvolvimento do projeto. Da metodologia *Lean Software Development*, o processo que foi mantido é o processo responsável por treinar as equipes e torná-las conscientes dos princípios e práticas *Lean* de projetos. Esta etapa, além do treinamento, oferece:

- **Acompanhamento e controle contínuo dos processos**

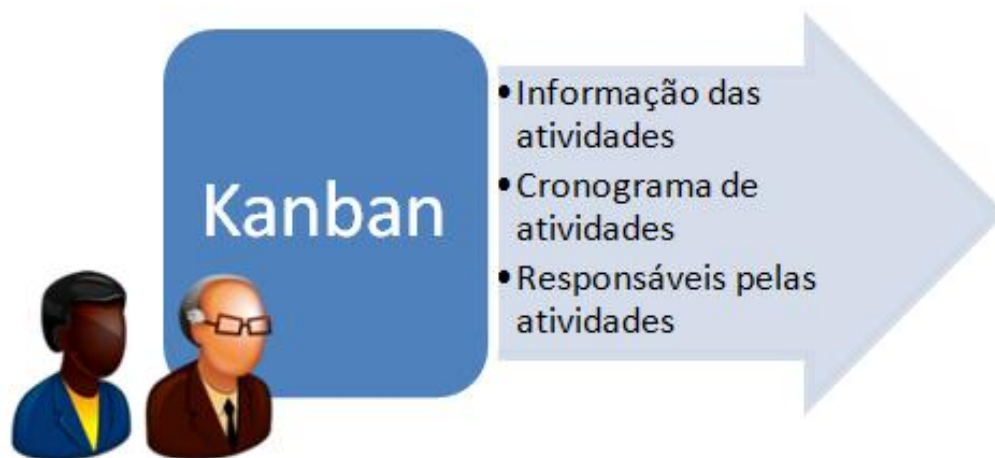
Este acompanhamento e controle contínuo dos processos é realizado pelo ESPECIALISTA em relação ao desenvolvimento e aplicação dos princípios e práticas *Lean* que os FUNCIONÁRIOS executam. Cabe ao ESPECIALISTA fazer a verificação e orientação sobre o melhor a se fazer e direcioná-lo para implantação correta da metodologia *Lean* sobre a metodologia proposta.

- **Revisão**

A revisão nesta etapa consiste em fazer a verificação e validação do que foi desenvolvido pelos FUNCIONÁRIOS, ou seja, verificar se o que foi desenvolvido por eles realmente foi

baseado nos princípios e práticas *Lean*, respeitando o que foi imposto pela mesma. O responsável por este processo é o ESPECIALISTA.

Além do treinamento *Lean* oferecido pelo ESPECIALISTA, existe o treinamento *Kanban* oferecido para os FUNCIONÁRIOS. Neste treinamento não é necessário que todos os membros das equipes envolvidas no projeto participem. É viável que apenas os líderes das equipes participem e repassem o conhecimento aos demais. Esta etapa é importante para que os FUNCIONÁRIOS de adéquem as práticas *Kanban*, desde o manuseio dos cartões no quadro *Kanban* ao direcionamento seguido para cumprir o WIP pré-definido. A Figura 44 ilustra melhor esta etapa, que foi inserida na metodologia proposta para adequar os processos *Kanban's* no gerenciamento de *software*.



**Figura 44 - Kanban**

Conforme apresentado pela Figura 44, a etapa de *Kanban* consiste no treinamento oferecido pelo ESPECIALISTA aos FUNCIONÁRIOS. Este treinamento é de suma importância devido às informações e experiências trocadas entre os líderes das equipes de desenvolvimento. Nesta etapa estão contidos os seguintes processos:

- **Informação das atividades**

Neste processo são repassadas as informações referentes às atividades aos FUNCIONÁRIOS. Estas informações consistem em como descrever as atividades nos cartões; qual *Kanban* será utilizado para cada atividade; qual ordem a lista de atividades deve seguir; como identificá-las caso existam atividades que necessitem de outras atividades; entre outras.

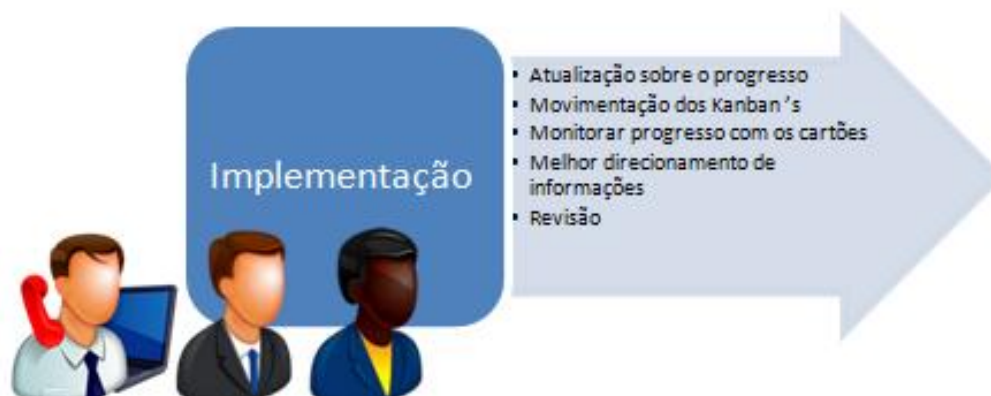
- **Cronograma de atividades**

O Cronograma de atividades é o mesmo da etapa de Planejamento, porém, é nesta etapa que o cronograma é repassado aos FUNCIONÁRIOS e como eles devem segui-lo na forma de cartões sobre o quadro *Kanban*, já que o cronograma e prazos de atividades são identificados a partir da utilização das colunas e cartões *Kanban's*.

- **Responsáveis pelas atividades**

Neste processo o ESPECIALISTA repassa aos FUNCIONÁRIOS como lidar com o quadro *Kanban* referente a responsabilidade de cada um. No *Kanban*, uma atividade pode ser desenvolvida por um único membro ou por uma equipe; ou, um único membro ou uma equipe pode executar mais de uma atividade ao mesmo tempo. Tudo isto deve ser controlado visualmente via cartões e quadro *Kanban*. É neste processo que os FUNCIONÁRIOS aprendem a lidar com a responsabilidade emitida pelos cartões e aprendem a ter um controle visual maior. Isto possibilita o cumprimento de prazos de forma mais eficaz, já que todos os envolvidos acompanham o que cada um está fazendo no projeto como um todo, obrigando-os a executar suas atividades de forma correta e dentro dos prazos estabelecidos.

A próxima etapa da metodologia proposta é a etapa de Implementação, onde será colocado em prática tudo que foi definido na fase de Escopo e Planejamento. Os responsáveis pela implementação do projeto são o CHEFE; o GERENTE; e o FUNCIONÁRIO. A Figura 45 apresenta a etapa de Implementação.



**Figura 45 - Implementação**

A Figura 45 apresenta a etapa de Implementação, que tem como objetivo a execução do projeto, onde as funcionalidades são desenvolvidas, as interfaces e códigos são gerados, entre outros. Esta etapa possui os processos:

- **Atualização sobre o progresso**

Este processo consiste na atualização sobre o progresso das atividades contidas no quadro *Kanban*. Este progresso é referente a movimentação dos cartões *Kanban*, sendo passado por exemplo, uma atividade da coluna 'Executando' para coluna de 'Prontas'. Esta atualização deve ser em tempo real, ou seja, assim que uma equipe conclui um trecho ou uma atividade, é necessário que seja exposto o status dessa atividade no quadro, para que todos percebam o andamento do projeto e o desenvolvimento das atividades e seus respectivos responsáveis. Essa fiscalização é importante que seja feita pelo GERENTE de projetos e por todos os envolvidos.

- **Movimentação dos *Kanban's***

O processo de movimentação de *Kanban's* consiste em mover os cartões que contém atividades, neles descritos, de posição (colunas). Cada coluna ilustra um status que servirá para todas as atividades. A quantidade de colunas poderá variar de acordo com o projeto e cabe ao GERENTE e o CHEFE das equipes limitarem essa quantidade. É importante ressaltar que cada cartão *Kanban* deve ser movimentado por todas as colunas existentes no quadro.

- **Monitoramento do progresso com cartões**

Este processo faz parte dos processos anteriores contidos na etapa de Implementação. Ele é baseado no controle visual que os cartões *Kanban's* possibilitam, ou seja, as colunas existentes no quadro *Kanban* servem para ilustrar as fases e etapas que cada parte do projeto deve percorrer, desde o início - que geralmente é a lista de atividades a serem executadas -, ao fim - que geralmente é a lista de atividades executadas-. O progresso dessas atividades deve ser visualizado conforme a coluna que a atividade está no momento da visualização. Esta fiscalização é realizada pelo GERENTE e as movimentações necessárias são realizadas pelos FUNCIONÁRIOS desenvolvedores das atividades.

- **Melhor direcionamento de informações**

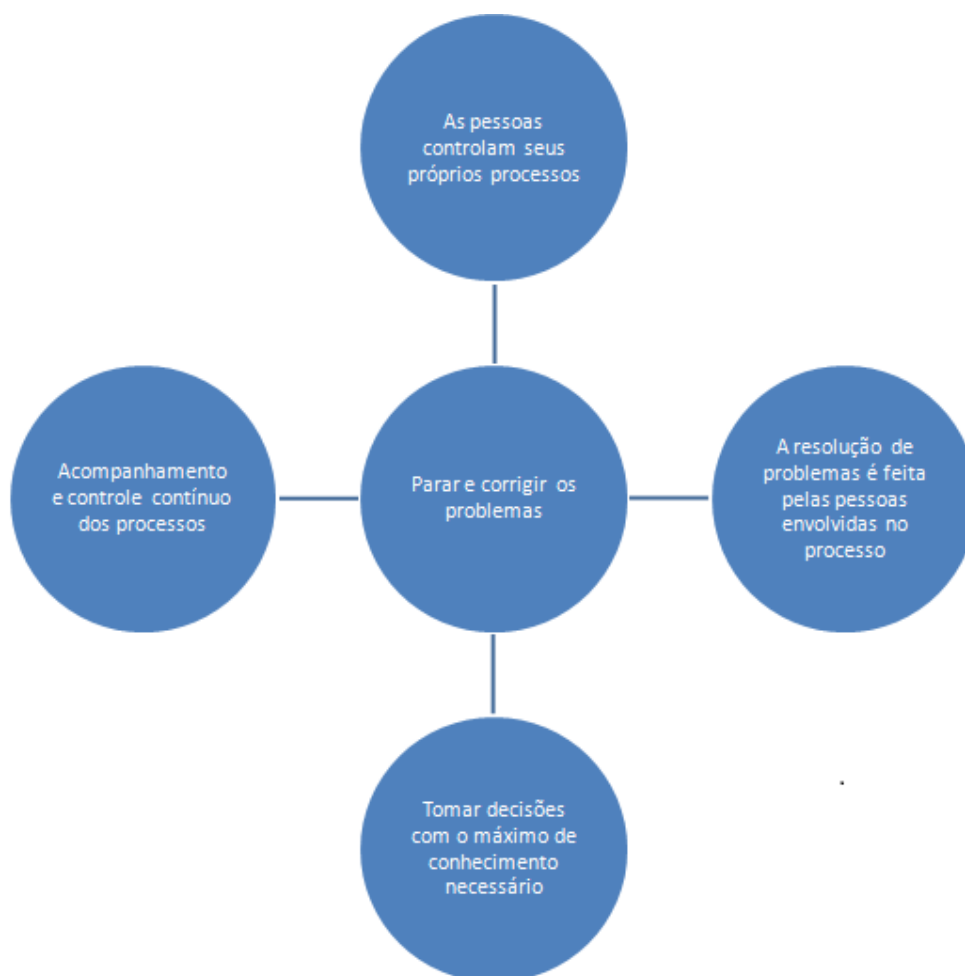
As informações serão melhores direcionadas conforme a visualização do fluxo de trabalho, ou seja, o *Kanban* prioriza o controle visual, gerando um melhor monitoramento sobre tudo que acontece no projeto. Isto faz com que todo o projeto se adéque aos prazos e custos estabelecidos nas etapas iniciais em seu desenvolvimento. Os processos interligados nesta etapa foram:

- ✓ **Eliminação de Desperdício**
- ✓ **Geração de Conhecimento**
- ✓ **Respeito às Pessoas**
- ✓ **Otimização do Todo**
- ✓ **Visualização do Fluxo de Trabalho**

Através dos itens listados, foi possível observar na metodologia proposta a necessidade de visualizar todo o projeto, para que se tenha a eliminação de desperdícios e otimização do todo. Conforme visualizado no quadro *Kanban*, o fluxo de trabalho faz com que haja respeito com os envolvidos, já que o quadro monitora as atividades e seus respectivos responsáveis, gerando uma base de conhecimento e experiências adquiridas, possibilitando a melhoria contínua nas atividades e no projeto. Os responsáveis por este processo são o GERENTE, o CHEFE e o FUNCIONÁRIO.

Para melhoria contínua dos processos na metodologia proposta, existem alguns pontos primordiais, conforme é apresentado pela Figura 46. Entre eles estão: o acompanhamento e

controle contínuo dos processos, ou seja, quando se tem o controle de algo, é possível melhorar esse algo; as pessoas controlam seus próprios processos, ou seja, a metodologia proposta acredita que pessoas motivadas geram uma maior qualidade, então elas mesmas controlam seus processos, sem interferência de outros envolvidos, a não ser que seja necessário. Parar e corrigir os problemas também é um ponto de suma importância, só que se deve levar em consideração a experiência que se tem sobre determinado problema, pois as vezes é mais viável esperar um pouco mais para tomada de decisão, do que tomá-la precipitadamente, sem o conhecimento suficiente.



**Figura 46 - Pontos de Melhorias Contínuas**

Conforme exibido pela Figura 46, existem pontos que auxiliam na execução dos processos. Estes pontos são interligados, sendo facilitador e complemento de outros pontos. Por

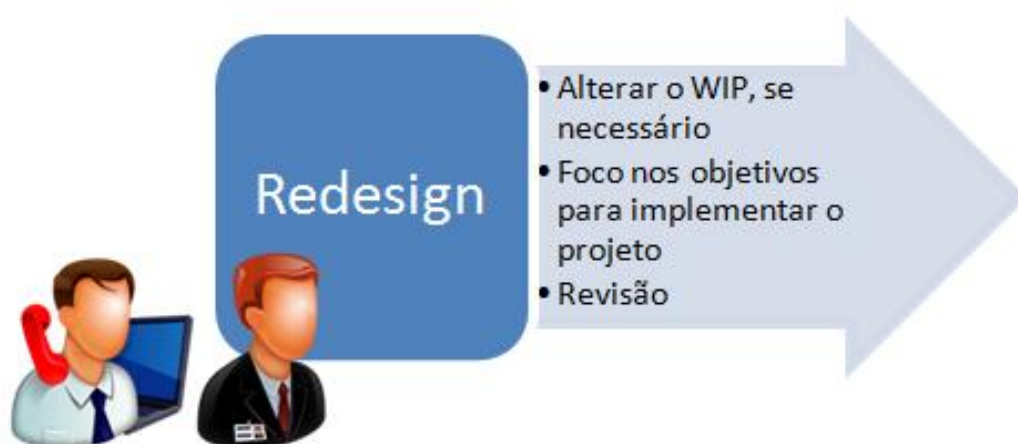


exemplo, para resolução de problemas, o ideal é tomar decisões com o máximo de conhecimento necessário, ou seja, dois pontos foram envolvidos. Assim como a resolução de problemas e o controle dos processos são executados pelos envolvidos de cada fase.

- **Revisão**

A Revisão é feita pelo CHEFE e pelo GERENTE, que são os responsáveis em analisar os progressos das atividades como um todo e pela revisão dos cartões *Kanban's*. Esta revisão é necessária para fazer a verificação se os cartões estarão de acordo com a realidade da atividade.

A próxima etapa da metodologia proposta é a etapa de *Redesign* (ou Reformulação), onde qualquer modificação evidenciada como necessária é realizada. Os responsáveis por estas execuções são o GERENTE e o CHEFE. A Figura 47 ilustra esta etapa e seus respectivos processos e desenvolvedores.



**Figura 47 - Redesign**

Nesta etapa estão contidos os processos referentes ao alcance do objetivo final, já que se trata das adequações e/ou modificações nas barreiras encontradas que possibilitaram o impedimento da entrega do produto final. Os processos contidos nesta etapa são:

- **Alterar o WIP, se necessário**

O *Work In Progress* (ou WIP) é o limite para o fluxo de trabalho, ou seja, é o limite definido para que determinada quantidade de atividades a serem executadas em paralelo seja respeitado. Como supracitado, esse valor deve ser fixo para todo o projeto e pode ser alterado conforme as necessidades, como por exemplo, se observado que equipes estejam ociosas aguardando as demais equipes concluírem alguma atividade, este WIP pode ser aumentado. Porém, este valor é válido para todo o projeto e não para uma ou duas atividades. Caso seja percebida a necessidade de aumentar ou diminuir o valor do WIP, é nesta etapa que este valor é modificado. Cabe ao GERENTE controlar o trabalho dos FUNCIONÁRIOS e os resultados esperados com os resultados até o momento, ou seja, fazer a verificação de prazos entre o esperado e a realidade do projeto.

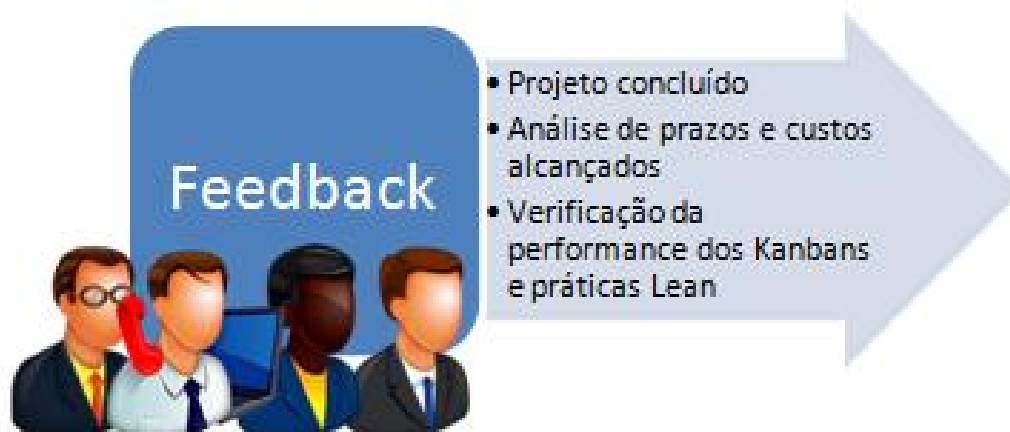
- **Foco nos objetivos para implementar o projeto**

É na etapa de *Redesign* que as pequenas modificações são realizadas, pois o projeto já foi implementado na fase anterior e deve conter apenas pequenos detalhes, seja ele um problema que está interrompendo o desenvolvimento do projeto ou até mesmo uma alteração no planejamento de algum item ou funcionalidade. Neste processo, todos os focos são para conclusão do projeto, ou seja, os mínimos detalhes são resolvidos e adaptados da melhor maneira possível e as barreiras são removidas.

- **Revisão**

Como toda revisão é para conferir e validar o que já foi desenvolvido, é nesta etapa que são feitas: a verificação da remoção de barreiras; a verificação dos prazos com o calendário de atividades definido, com o WIP e com os resultados (obtidos até o momento). É de suma importância que o GERENTE e o CHEFE se reúnam nesta etapa para conversar sobre tudo que já foi desenvolvido, pois se deve ter o projeto praticamente pronto, com todos os problemas resolvidos.

A próxima etapa da metodologia proposta consiste na etapa de *Feedback* (Retorno), onde tudo que foi desenvolvido gera análises e resultados. Estas análises podem ser emitidas pelos FUNCIONÁRIOS, pelos CLIENTES ou pelo GERENTE. A Figura 48 apresenta esta etapa e seus respectivos processos.



**Figura 48 - Fim do projeto**

A etapa de *Feedback* retrata o fim do projeto, onde tudo que foi desenvolvido pelos FUNCIONÁRIOS é entregue ao GERENTE, o GERENTE fica responsável em fazer as revisões necessárias e repassa o que foi feito para o CLIENTE. Esta etapa é a conclusão e o fechamento de tudo que foi planejado pelos envolvidos. Nesta etapa estão contidos os seguintes processos:

- **Projeto é concluído;**

Neste processo as atividades já estão concluídas, bastando apenas a entrega das mesmas ao GERENTE, que ficou responsável em fazer os testes de verificação e validação. Caso as atividades estejam de acordo com o que foi definido nas etapas de Solicitação, Escopo e Planejamento, o projeto é dado como finalizado.

- **Análise de prazos e custos alcançados**

Esta análise de prazos e custos é a relação entre o planejado com o que foi desenvolvido. Se houve o cumprimento das normas, se o que foi definido no escopo foi respeitado, entre outros. Este processo é de suma importância por conta das análises e experiências adquiridas, que servirão para projetos que ainda serão executados. Por isso é importante a análise para verificar o que o projeto concluído teve de negativo ou positivo, para que se possa aprimorar em projetos futuros.

- **Verificação da performance dos *Kanban's* e práticas *Lean***

Este processo é de suma importância para a metodologia proposta. Ele é responsável em fazer a verificação da performance do *Kanban* e do *Lean* na execução do projeto. Se foi viável a execução de ambos no desenvolvimento do projeto; se o controle visual foi eficaz; se a performance das equipes em relação ao início do desenvolvimento do projeto melhorou em relação ao esperado, entre outros. Esta verificação de performance pode ser cada vez mais aprimorada com o decorrer do desenvolvimento de outros projetos, pois as equipes se tornarão cada vez mais maduras e conscientes das práticas e princípios *Lean* e da eficácia da ferramenta *Kanban*.

Nesta seção foram apresentados os processos gerados com a junção da metodologia *Lean Software Development* com a ferramenta de controle visual *Kanban*. Além disso, foram ilustrados os pontos considerados de suma importância para a melhoria contínua da metodologia proposta, que deverão ser implantados e respeitados no gerenciamento dos projetos.

Nas próximas seções são apresentados os artefatos dos processos criados para a proposta do presente trabalho. Vale ressaltar que alguns dos itens citados englobam as metodologias *Lean* e *Kanban* e os demais foram acrescentados de forma a gerar adequação em ambas as metodologias.

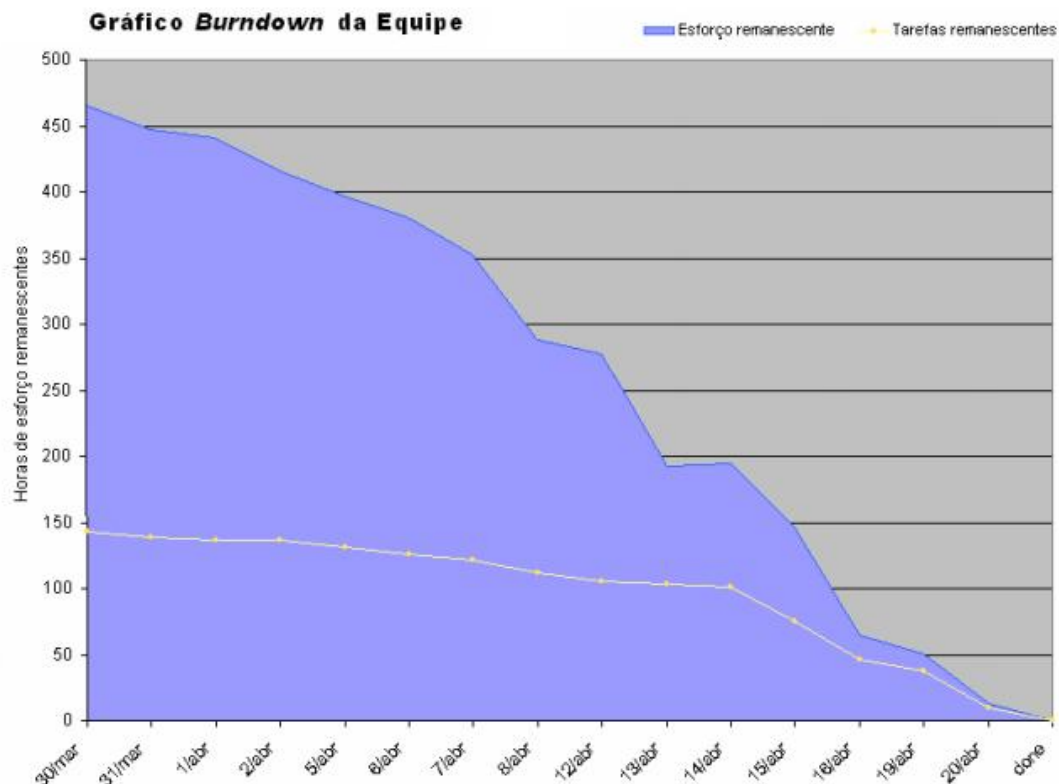
#### **4.5. Artefatos da Metodologia**

Os artefatos auxiliam no desenvolvimento do projeto, pois eles são os resultados de uma atividade que foi desenvolvida. Os artefatos podem ser: um modelo, um documento ou um

código produzido por uma atividade. Eles são interligados às etapas e aos processos da metodologia proposta e são apresentados a seguir.

- **Gráficos visuais**

Os Gráficos visuais são utilizados para uma sinalização local e podem ser direcionados para distribuição de funcionalidades, ilustração de resultados, análises de dados, entre outros. Este artefato é baseado na metodologia *Scrum* e está presente nas etapas de *Redesign* e *Feedback*. A Figura 49 exibe um Gráfico *Burndown* de uma equipe, que tem como objetivo focar na relação entre as tarefas desenvolvidas e o esforço aplicado para desenvolvê-las.

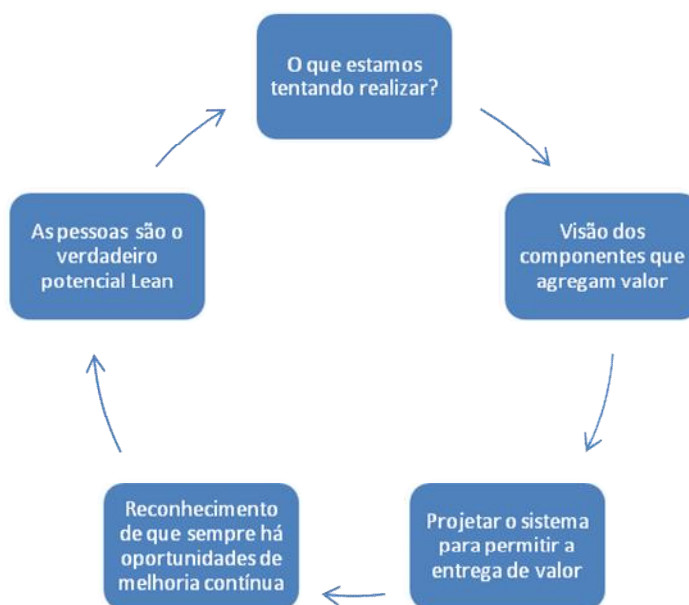


**Figura 49 - Exemplo de Gráfico Visual (Fonte: DENK, 2005)**

A Figura 49 ilustra um exemplo dos controles visuais utilizados para acompanhar a execução do projeto. A curva azul representa a estimativa de esforço necessário para finalizar as tarefas na iteração atual e a linha amarela corresponde à quantidade de funcionalidades para serem codificadas e incorporadas ao incremento do produto a ser entregue no fim da iteração.

- **Reuniões diárias**

As reuniões diárias são reuniões breves com todos os membros das equipes de desenvolvimento do projeto. Durante a reunião são debatidos e analisados os assuntos referentes ao projeto em desenvolvimento. Este artefato é baseado na metodologia *Scrum* e está presente nas etapas de Planejamento, Implementação e *Redesign*. A Figura 50 ilustra alguns dos pontos primordiais nas reuniões da metodologia proposta.



**Figura 50 - Exemplos de pautas para as Reuniões**

Conforme mostra a Figura 50, estas reuniões começam com “O que se pretende realizar”, diante desta interrogação, são englobados alguns pontos, como: sistema; fluxo; perfeição e pessoas. O sistema abrange uma visão geral sobre os componentes que agregam valor ao produto final, sendo obrigatório este conhecimento para todos. O fluxo debatido nas reuniões é referente ao *design*/projeto do sistema, para que o mesmo seja entregue com o valor dado como ideal sem o mínimo de interrupções durante seu desenvolvimento.

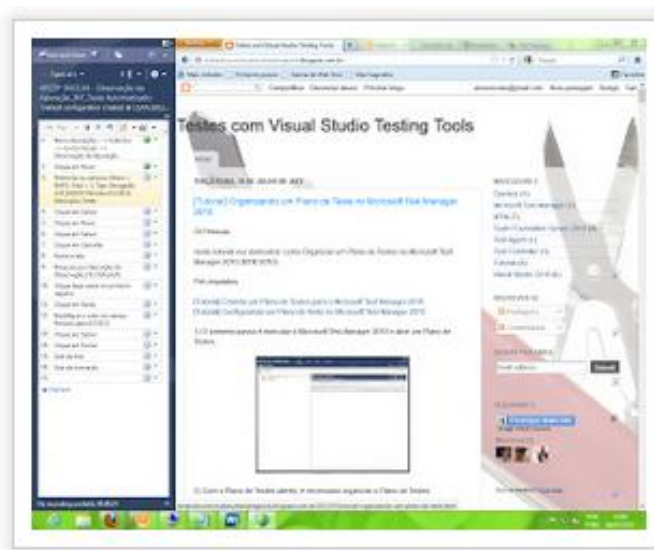
O terceiro ponto a ser debatido nas reuniões é a perfeição, onde é reconhecido que sempre existem oportunidades para uma melhoria contínua e, muitas vezes, diárias. Este ponto é crucial nas reuniões devido as mudanças que o mesmo gera no projeto como um todo, já que essa verificação sempre colabora para uma melhoria contínua.

O último ponto são as pessoas, que são consideradas os motores da metodologia *Lean*, já que são elas as pessoas responsáveis por todo o desenvolvimento dos projetos. Este ponto refere-se ao entusiasmo da equipe, o respeito entre os membros, o incentivo a criatividade na execução de atividades, ao companheirismo entre equipes, ao diálogo entre membros e supervisores, entre outros. A metodologia *Lean* defende que pessoas encorajadas e criativas são de suma importância para um projeto de sucesso e com o valor agregado esperado.

- **Testes automatizados**

O desenvolvimento de *software* é dirigido por testes. No fluxo *Lean*, após a implementação existe a revisão, onde códigos são testados e possíveis empecilhos são removidos. Já no fluxo da metodologia proposta os testes são executados com maior frequência, geralmente após a inserção, modificação e/ou exclusão de alguma atividade ou processo. Os testes são de suma importância, pois eles simulam o ambiente do cliente (ou usuário final), verificando funcionalidades e testando as iterações entre partes distintas do sistema. Este artefato está presente no processo de Revisão da etapa de Implementação.

A fase de testes é essencial no desenvolvimento de projetos, pois reduz o número de erros e possíveis retrabalhos posteriores. A fase de testes previne que barreiras surjam na fase de *Redesign*. Para executar testes automatizados existem inúmeras ferramentas, a Figura 51 exibe um exemplo, que é a ferramenta *Microsoft Test Manager*.



**Figura 51 - Microsoft Test Manager (Fonte: MICROSOFT, 2012)**

Conforme exibido pela Figura 51, a ferramenta da *Microsoft* é um exemplo bastante utilizado para executar testes automatizados. Ela permite o gerenciamento de alguns testes de sistema. A ferramenta *Test Manager* é baseada em planos de testes e casos de testes, e, executa os testes automatizados vindos de outras plataformas, como por exemplo, o *Visual Studio*.

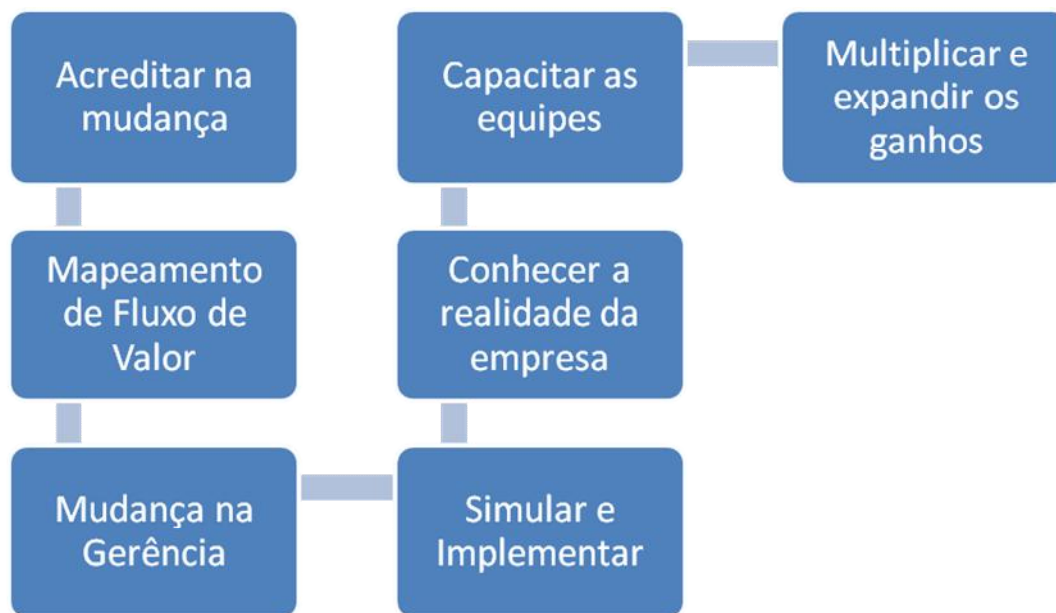
- **Integrações frequentes**

As integrações são quando partes do projeto que foram desenvolvidas independentemente de outras partes são juntadas, ou seja, quando existe a união entre as partes executadas. Este artefato está presente na etapa de Implementação e *Redesign*. É importante ter integrações frequentes no desenvolvimento de projetos, devido a análise do todo e devido as partes serem menores, mais fácil para controlar. Quando se integra com uma frequência maior é possível integrar partes menores, com isso, se surgir erros e/ou defeitos e/ou sugestões, essas adaptações são melhores inseridas ou retiradas de acordo com a parte integrada.

Integrar as informações dos sistemas pode ser uma tarefa complicada devido a eficiência proposta por cada parte do sistema. A frequência destas integrações pode variar de acordo com as estratégias de trabalho da equipe desenvolvedora, mas o importante é que toda a equipe entenda o processo como um todo e que, quando a integração acontecer, todas as funcionalidades estejam funcionando corretamente, agregando valor ao produto final.

Fazendo uma analogia a um jogo de quebra-cabeças, cada parte do jogo seria uma parte do projeto e o jogo montado seria a integração realizada, podendo ser de forma satisfatória, que é o objetivo final, ou, com possíveis melhorias. Estas melhorias são alcançadas através de guias e conforme a Figura 52, a metodologia proposta possui sete *Rounds*. Estes *Rounds* são guias de boas práticas, que quando implantados da forma correta, trazem vários benefícios para toda a equipe de desenvolvimento.





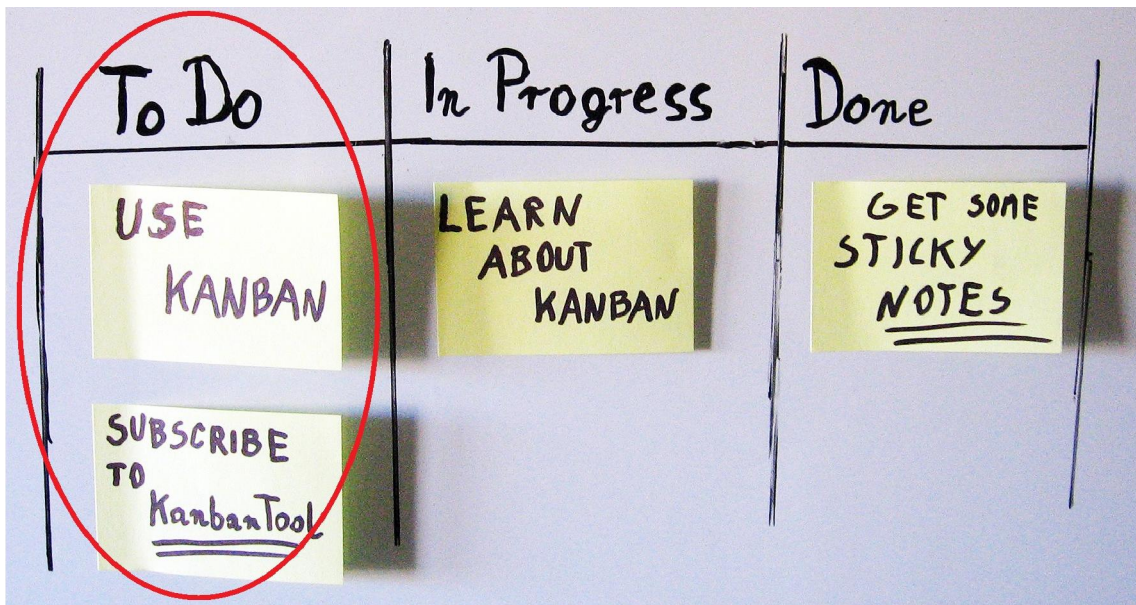
**Figura 52 - Rounds da Metodologia**

Conforme visualizado na Figura 52, a metodologia *Lean* possui *Rounds*. O passo Acreditar na Mudança refere-se a quebrar os paradigmas da empresa e conseguir aprender a identificar os desperdícios. O Mapeamento de Fluxo de Valor refere-se a desenhar o fluxo futuro, definir cronograma, calcular ganhos e retorno sobre o investimento, ou seja, avaliar diferentes tipos de soluções, custos e dificuldade. A mudança na Gerência envolve as mudanças como um todo, pois existem gestores que resistem às mudanças evidenciadas e as melhorias contínuas devem partir do nível gerencial do projeto. Capacitar as equipes envolve treinar o time do projeto, treinar todos os envolvidos tanto na teoria quanto na prática, seja com jogos, simulações, vídeos, etc. Conhecer a realidade da empresa significa conversar com o restante da hierarquia da empresa, como por exemplo, os funcionários, ou seja, envolver as pessoas que exercem de fato algum tipo de função no projeto. Significa também ir a um nível inferior e observar a realidade da empresa com diferentes visões.

Simular e Implementar é praticar as melhorias contínuas (*Kaizen*), ou seja, fazer tudo para garantir os resultados esperados, mesmo simulando o desenvolvimento no papel; marcando reuniões; debatendo os assuntos iniciais; tudo colabora pra uma entrega satisfatória. Por fim, Multiplicar e Expandir os Ganhos, que é a partir do treinamento nas equipes, fazer com que elas continuem desenvolvendo o que aprenderam, sempre replicando o conhecimento e melhorando os processos. Esse *Round* deve sempre estar expandindo dentro das organizações, seja expandindo para outros setores, fornecedores e clientes.

- **Lista de atividades a serem executadas**

O *Kanban* visa o melhor monitoramento das atividades e para isso, utiliza um quadro *Kanban* que possui várias colunas (a quantidade de colunas pode variar de acordo com as estratégias de trabalho do projeto). Porém, excepcionalmente para a metodologia proposta, existem três colunas: a coluna de lista de atividades a serem executadas, conforme ilustrado pela Figura 53, a lista de atividades em andamento e lista de atividades concluídas.



**Figura 53 - Coluna de atividades a serem feitas (Fonte: NICOLAY, 2013)**

A coluna de atividades a serem feitas, ou também descrita como A Fazer (na Figura 53 - *To Do*), contém a lista de todas as atividades a serem desenvolvidas. Geralmente, os gestores procuram colocar além das atividades, os responsáveis por cada atividade, pois isto facilita a visão global sobre todas as atividades e seus respectivos desenvolvedores.

- **Lista de atividades em andamento**

A lista de atividades em andamento, como o próprio nome já diz, contém as atividades que passaram da coluna anterior, que era de atividades a serem feitas, para atividades em andamento, ou seja, elas foram iniciadas e estão na fase de desenvolvimento. Geralmente, a

lista de atividades em andamento é a segunda coluna no quadro *Kanban*. A Figura 54 mostra um exemplo de coluna de atividades em andamento em um quadro *Kanban*.



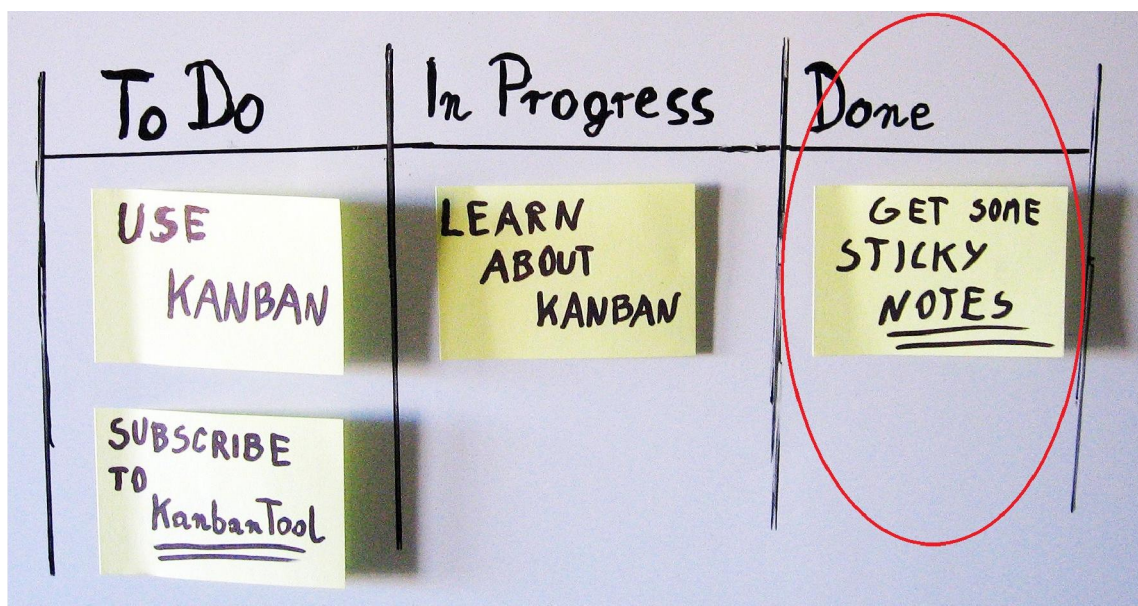
**Figura 54 - Coluna de atividades em andamento (Fonte: NICOLAY, 2013)**

A coluna de atividades em andamento é de suma importância para monitoramento entre a relação prazos e atividades (na Figura 54 - *In Progress*). Esta coluna engloba o andamento do projeto, ou seja, o que está sendo desenvolvido em relação a todas as atividades que devem ser iniciadas. É de grande auxílio ao gestor estar sempre observando as equipes que estão trabalhando e as equipes que, invariavelmente, estão ociosas. Caso seja observada algum excesso de atividades para determinada equipe e/ou equipes desocupadas, é necessário aumentar a lista de atividades em andamento e remanejar a equipe de modo a manter um equilíbrio entre as partes.

É nesta coluna que se define o limite de atividades em progresso, ou seja, quantas atividades poderão estar nesta coluna ao mesmo tempo. Por exemplo, se o limite de atividades em progresso for quatro, apenas quatro atividades poderão estar sendo executadas ao mesmo tempo. Esse limite imposto facilita o monitoramento e equilibra as atividades entre as equipes. Vale ressaltar que esse limite não é fixo, podendo ser aumentado ou diminuído de acordo com o desenvolvimento das atividades.

- **Lista de atividades concluídas**

A coluna que contém as atividades concluídas geralmente é a última coluna em um quadro *Kanban*. Ela contém um resultado parcial do projeto, ou seja, as atividades que já foram desenvolvidas. Esta coluna possibilita o início de mais atividades, caso contenham mais atividades em espera, porque com o limite de atividades em progresso, as atividades não podem ser executadas todas ao mesmo tempo. Assim que algumas delas forem concluídas e passarem pra coluna de atividades concluídas, novas atividades poderão ser iniciadas. A Figura 55 exibe um exemplo desta coluna em um quadro *Kanban*.



**Figura 55 - Coluna de atividades concluídas (Fonte: NICOLAY, 2013)**

Esta coluna possibilita o monitoramento do andamento do projeto como um todo e como as equipes estão desenvolvendo suas partes (na Figura 55 - *Done*). É viável a comparação entre as colunas e os prazos estabelecidos, pois com a junção da metodologia *Lean* com a ferramenta *Kanban* são fornecidas entregas rápidas das partes do projeto. Com isso, deve-se respeitar o acordo pré-estabelecido no início do projeto, com entregas frequentes e *feedback* suficiente para satisfação dos envolvidos.

- **Lista de membros da equipe de desenvolvimento e papéis**

A Lista de membros da equipe de desenvolvimento e papéis contém todos os envolvidos do projeto, ou seja, todos os membros que irão executar alguma atividade no projeto. Além da

coluna que lista todos estes envolvidos, existe a coluna de papéis, que apresentará a função de cada membro. Este artefato é baseado nos processos do RUP e está presente na etapa de Planejamento na metodologia proposta. A Figura 56 apresenta este artefato.

Empresa X		<Data>	
Projeto Y		Projeto<Número><Nome><Versão>	
<b>Lista de Membros da equipe de desenvolvimento e seus Papéis</b>			
<Introdução>			
<Membros>		<Papéis>	
<Local>, <Data>			
<hr/> <Gerente> Gerente do Projeto			

**Figura 56 - Lista de Membros e Papéis**

Conforme apresentado pela Figura 56, o artefato que contém a lista de membros e seus respectivos papéis, tem como objetivo definir responsabilidades dentro das equipes de desenvolvimento. Por exemplo, o membro João será o Chefe da equipe B e assim sucessivamente. Após todos os envolvidos terem sido listados, é importante que o GERENTE assine o documento e repasse para os desenvolvedores.

- **Plano de reuniões**

O Plano de Reuniões é um documento que apresenta quais dias terão reuniões e quem deve participar. Este artefato é baseado nos processos do RUP e é importante para que exista diálogo entre as equipes, excluindo as esperas e os desperdícios, já que serão agendadas as reuniões para quebra de paradigmas e barreiras que porventura surgirem no desenvolvimento

dos projetos. O mesmo está presente na etapa de Planejamento, *Redesign* e *Feedback*. A Figura 57 mostra este documento.

Empresa X Projeto Y		<Data> Projeto<Número><Nome><Versão>	
<b>Plano de Reuniões</b>			
<Introdução>			
Reunião	Data	Horário	Membros
			<Local>, <Data>
<hr/> <Gerente> Gerente do Projeto			

**Figura 57 - Plano de Reunião**

Conforme visualizado na Figura 57, o plano de reunião contém algumas colunas que servirão como um agendamento, ou seja, o nome, a data, o horário e os membros que deverão participar de determinada reunião. Vale ressaltar que o plano de reunião deverá ter o consentimento do GERENTE para assiná-lo e repassá-lo para os interessados.

Nesta seção foram abordados os artefatos contidos na metodologia proposta, ou seja, tudo aquilo que auxilia ou faz parte da execução das atividades. Além disso, foram apresentados os princípios, os componentes essenciais, as etapas e processos, os papéis, os artefatos e os itens necessários para o melhor gerenciamento de *software*. Na próxima seção serão apresentadas as considerações finais sobre o trabalho desenvolvido e uma visão geral dos possíveis trabalhos futuros.

## 5. CONSIDERAÇÕES FINAIS

Neste trabalho se propôs o desenvolvimento de uma metodologia de gerenciamento de *software* para o acompanhamento de projetos utilizando o método *Kanban* e a metodologia *Lean Software Development*. Os principais objetivos do *Kanban* são evitar o retrabalho e identificar falhas no processo durante o seu desenvolvimento, isso é feito através da simples e eficiente visualização dos status do projeto e de cada atividade em um quadro *Kanban*. O *Kanban* não define um processo ou metodologia de desenvolvimento ou gerenciamento em si, ele se adapta ao processo utilizado. O mesmo possui três regras: Visualize o fluxo de trabalho; Limite o trabalho em progresso; e, Acompanhe a execução da atividade.

Estas três regras foram aliadas a novos processos e novas etapas, estas baseadas na metodologia *Lean Software Development*. A metodologia proposta foca no gerenciamento de *software* e todos os processos gerados nela baseiam-se na eliminação de desperdícios, melhor direcionamento de recursos e melhor controle visual das atividades.

Com a análise dos princípios da metodologia *Lean* e sua aplicação no desenvolvimento de *software*, é possível afirmar que esta metodologia serve, entre suas demais características, para auxiliar na identificação de problemas que necessitam ser adequados dentro de processos de gerenciamento de *software*. Estes problemas são resolvidos fazendo a implantação dos princípios, valores e ferramentas, esta última se tem como exemplo o *Kanban*. Todos estes fazem com que o desenvolvimento se torne enxuto e mais coeso, facilitando no gerenciamento dos projetos e entregas satisfatórias.

A metodologia proposta resulta em um produto desenvolvido com base em uma melhoria contínua, onde a visão de negócios encontrada na metodologia *Lean* e o controle visual eficaz encontrado no *Kanban* são aplicados. Isto faz com que o cliente participe de todo o processo de gerenciamento do projeto, de forma muitas vezes indireta, o que acarreta na satisfação garantida do cliente. O processo proposto definiu princípios, papéis, artefatos, etapas, atividades e padrões de gerenciamento de *software*. Tudo isto permitindo que as equipes possam utilizar destes processos para a execução de projetos com foco na agregação de valor para o cliente.

É importante ressaltar que a união do *Lean* e do *Kanban* tem como objetivo aumentar a rentabilidade dos negócios nos quais envolvem o gerenciamento de *software*, a partir dos

princípios de manufatura e controle de produção. Juntas, provavelmente, estas técnicas se tornam ferramentas eficazes para empresas de pequeno, médio e grande porte.

A partir da metodologia demonstrada neste trabalho, onde são adaptados os processos *Kanban*, juntamente com a metodologia *Lean*, chega-se a conclusão de que estas metodologias podem ser aplicadas de uma maneira complementar, ou como alternativa às metodologias tradicionais.

### **5.1. Trabalhos Futuros**

É válido ressaltar que embora a aplicabilidade da metodologia proposta tenha ficado explícita no decorrer deste trabalho, sua eficácia não pôde ser avaliada por motivos institucionais, ou seja, não existem equipes de desenvolvimento para implantação da metodologia. Para que isso seja possível, sugere-se a realização de um estudo mais aprofundado que avalie como a metodologia proposta influencia na prática a identificação e o tratamento de falhas, como atrasos e desperdícios, de maneira a tornar a técnica de gerenciamento de projetos de *software* mais prática e mais viável.

Além disso, aplicar os treinamentos dos princípios e práticas da metodologia proposta para *start-ups*, já que o foco da metodologia é a simplicidade em aplicar seus processos e aumentar a rentabilidade dos negócios. Isto é possível definindo processos de criação de novos produtos e serviços com o mínimo de desperdícios, através do desenvolvimento ágil, muita iteração, medição e validação junto com os consumidores. A metodologia proposta leva consigo o pensamento enxuto, no qual os investimentos são escassos e limitados, se encaixando com a realidade da maioria das *start-ups* brasileiras.



## 6. REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, Paulo. **Fundamentos da Gestão de Projetos**. 2012. Disponível em: <<http://www.slideshare.net/pejota81/gerenciamento-de-projetos-apostila-completa>>. Acesso em 13 de Março de 2013.

ALVES, Rafael; VANALLE, Rosângela. **Ciclo de vida de desenvolvimento de sistemas – visão conceitual dos modelos clássico, espiral e prototipação**. Disponível em: <[http://www.abepro.org.br/biblioteca/ENEGEP2001\\_TR93\\_0290.pdf](http://www.abepro.org.br/biblioteca/ENEGEP2001_TR93_0290.pdf)>. Acesso em 20 de Março de 2013.

AUXILIADORA, Maria. **Gerenciamento de Projetos de Software**. Disponível em: <<http://www.deinf.ufma.br/~maria/arqan/cap5-cronograma.pdf>>. Acesso em 17 de Março de 2013.

BASSI, D. **Experiências com desenvolvimento ágil**. 2008. 170 f. Dissertação (Mestrado) – Departamento de Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2008.

BOENTE, Alfredo; OLIVEIRA, Fabiano; ALVES, João. **RUP como Metodologia de Desenvolvimento de Software para Obtenção da Qualidade de Software**. Disponível em: <<http://www.boente.eti.br/publica/seget2008rup.pdf>>. Acesso em 21 de Março de 2013.

BOSNIC, I. **Desenvolvimento Lean de software: conceitos e aplicações**. Disponível em: <<http://www.ebah.com.br/content/ABAAAfSDMAE/artigo-lean>>. Acesso em 17 de Fevereiro de 2013.

CAMARA, Fabio. **Scrum, uma metodologia ágil**. Disponível em: <<http://imasters.com.br/artigo/10699/gerencia-de-ti/scrum-uma-metodologia-agil/>>. Acesso em 21 de Março de 2013.

CUNHA, Vitor Castro. **Kanban – Gestão de Estoques**. Disponível em: <<http://www.vh2c.com.br/kanban.pdf>>. Acesso em 19 de Março de 2013.

CORREA, Lucas. **Introdução ao Agile: conceitos básicos**. Disponível em: <<http://blog.mlvconsultoria.com.br/2011/08/04/introducao-ao-agile-conceitos-basicos/>>. Acesso em 19 de Abril de 2013.

FORTES, C. **Aplicabilidade de Lean Service na melhoria de serviços de tecnologia da informação**. 168 p. Porto Alegre, 2010.

GHISI, T. **Kanban no desenvolvimento de projetos de software: Entendendo os desafios e a receita para o sucesso**. Engenharia de *Software Magazine*. Edição 45. 11p.

GUSTAVSSON, H. **Lean thinking applied to system architecting**. 2011. 72 f. Tese (Doutorado) – Departamento de School of Innovation. Desing and Engineering, Malardalen University, Vasteras, 2011.

HIBBS, C; JEWETT, S; SULLIVAN, M. **The art of lean software development**. Sebastopol: O'Reilly Media, Inc., 2009. 144 p.

IMAI, M. 1990. **Kaizen: a estratégia para o sucesso competitivo**. Tradução Cecília FagnaniLucca. 3ª Ed. IMAM.

KNIBERG, H. **Lean from the Trenches: Managing Large-Scale Projects with Kanban**. Dallas: The PragmaticBookshelf, 2011. 165 p.

KNIBERG, H.; SKARIN, M. **Kanban e Scrum, obtendo o melhor de ambos**. Edição *online gratuita*. *Enterprise Software Development Series*. 2009. 139 p.

MARIOTTI, F. **Kanban: o ágil adaptativo**. Engenharia de *Software Magazine*. 11 p. Edição 45.

MENDES, Antonio. **Gestão de Projetos de Software**. Disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-2-gestao-de-projetos-de-software/9143>>. Acesso em 13 de Março de 2013.

MOTA, Kleber. **Gerência de Projetos de Software**. Disponível em: <<http://www.fattocs.com.br/livro-apf/citacao/KleberMota.pdf>>. Acesso em 17 de Março de 2013.

OHNO, T. **O Sistema Toyota de Produção** – além da produção em larga escala. 1988. Porto Alegre: Artes Médicas, 1997.

PEDROSA, M. **Kanban no desenvolvimento de Software**. 37 p. Universidade Federal de Pernambuco. Recife, 2012.

PEREIRA, Guilherme. **Metodologia Lean de desenvolvimento de Software**: uma visão geral. Disponível em: <[http://www.ceavi.udesc.br/arquivos/id\\_submenu/387/guilherme\\_metodologia\\_lean\\_de\\_desenvolvimento\\_de\\_software\\_\\_uma\\_visao\\_geral.pdf](http://www.ceavi.udesc.br/arquivos/id_submenu/387/guilherme_metodologia_lean_de_desenvolvimento_de_software__uma_visao_geral.pdf)>. Acesso em 21 de Março de 2013.

POPPENDIECK, M; POPPENDIECK, T. **Implementando o desenvolvimento lean de software**: do conceito ao dinheiro. Porto Alegre: Bookman, 2011. 280 p.

POPPENDIECK, M.; CUSUMANO, M. **Lean Software Development**: A tutorial. 7 p. *IEEE Software*, 2012.

RODRIGUES, Carlos. **O que é um projeto**. 2005. Disponível em: <<http://discursocitado.blogspot.com.br/2005/09/o-que-um-projeto.html>>. Acesso em 13 de Março de 2013.

SOTILLE, Mauro. **Gerenciamento de Projetos na Engenharia de Software**. Disponível em: <[http://www.pmtech.com.br/artigos/Gerenciamento\\_Projetos\\_Software.pdf](http://www.pmtech.com.br/artigos/Gerenciamento_Projetos_Software.pdf)>. Acesso em 20 de Março de 2013.

5CQualiBr. **Desenvolvimento Ágil de *Software***. Disponível em:  
<<http://www.softwarepublico.gov.br/5cqualibr/xowiki/DesenvolvimentoAgil>>. Acesso em 19  
de Março de 2013.