

UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS

CIÊNCIAS ECONÔMICAS E ADMINISTRATIVAS

MBA EM ADMINISTRAÇÃO DE TI

DIEGO SILVA MARTINS

GESTÃO DE EQUIPES *LEAN* –

ESTUDO SOBRE A UTILIZAÇÃO DO *PEOPLE-CMM*

NO CONTEXTO DE DESENVOLVIMENTO DE SOFTWARE *LEAN*

São Leopoldo

2010

DIEGO SILVA MARTINS

GESTÃO DE EQUIPES *LEAN*: ESTUDO SOBRE A UTILIZAÇÃO DO *PEOPLE-CMM*
NO CONTEXTO DE DESENVOLVIMENTO DE SOFTWARE *LEAN*

Projeto apresentado à Universidade do Vale do Rio dos Sinos como requisito parcial para aprovação na atividade acadêmica de capacitação para o trabalho de conclusão do MBA em Administração de Tecnologia da Informação.

Orientador: Prof. Sérgio Crespo

São Leopoldo

2010

SUMÁRIO

1 INTRODUÇÃO	5
1.2 PROBLEMA DE PESQUISA	7
1.3 OBJETIVOS	7
1.3.1 Objetivo Geral	8
1.3.2 Objetivos Específicos	8
2 FUNDAMENTAÇÃO TEÓRICA	8
2.1 ORIGENS DO <i>LEAN THINKING</i>.....	9
2.2 <i>LEAN THINKING</i> NO DESENVOLVIMENTO DE SOFTWARE.....	12
2.2.1 O problema em satisfazer as necessidades de negócio	12
2.2.2 Complexidade envolvida no desenvolvimento de software.....	13
2.2.2.1 Processo de controle empírico	14
2.2.2.2 Dimensões da complexidade	16
2.2.3 Importância do aprendizado através de <i>feedback</i>	17
2.2.4 O desperdício no desenvolvimento de software	19
2.2.5 Eliminando o desperdício através do mapeamento da cadeia de valor	23
2.2.6 A proposta do <i>framework</i> Scrum.....	25
2.3 GESTÃO DE PESSOAS NO CONTEXTO <i>LEAN</i>	29
2.3.1 Rompendo os limites da Administração Científica	29
2.3.2 Diferença entre os modelos <i>Lean</i> e CMM/CMMI	30
2.3.3 Fatores relevantes na gestão de equipes de desenvolvimento.....	33
2.3.4 Equipes <i>Lean</i>	37
2.3.4.1 Fatores motivacionais	37
2.3.4.2 O papel dos líderes de desenvolvimento	39
2.3.4.4 A proposta do Time Scrum.....	41
2.4 <i>People-CMM</i>.....	42
3 METODOLOGIA DE PESQUISA.....	45
3.1 ETAPAS DO MÉTODO.....	46
3.1.1 Pesquisa bibliográfica sobre práticas de desenvolvimento <i>lean</i> e do <i>framework</i> Scrum no contexto de gestão de equipes.....	46
3.1.2 Descrição das áreas de processos do P-CMM relevantes para análise.....	47

3.1.3 Comparação das práticas de gestão de equipes identificadas com os objetivos das áreas de processo	48
4. ANÁLISE E APRESENTAÇÃO DOS RESULTADOS.....	49
4.1 DESCRIÇÃO DAS ÁREAS DE PROCESSOS.....	49
4.1.1 Construindo grupos de trabalho e cultura	50
4.1.1.1 Processo do nível Gerenciado	51
4.1.1.1.1 <i>Comunicação e coordenação</i>	51
4.1.1.2 Processos do nível Definido	52
4.1.1.2.1 <i>Cultura participativa</i>	53
4.1.1.2.2 <i>Desenvolvimento de grupos de trabalho</i>	54
4.1.1.3 Processos do nível Previsível	55
4.1.1.3.1 <i>Integração de competências</i>	55
4.1.1.3.2 <i>Grupos de trabalho empowered</i>	56
4.2 COMPARÇÃO DOS PRINCÍPIOS E PRÁTICAS LEAN COM OS OBJETIVOS DAS ÁREAS DE PROCESSOS.....	57
4.2.1 Verificação do nível Gerenciado	57
4.2.2 Verificação do nível Definido	61
4.2.3 Verificação do nível Previsível	65
5 CONCLUSÃO	68
REFERÊNCIAS	70

1 INTRODUÇÃO

O paradigma econômico atual estabelece que as organizações mantenham um ritmo de inovação contínua, gerando produtos e serviços com maior valor agregado, e que se preocupem com a satisfação do cliente. A evolução dos processos de negócio, impulsionada pelo paradigma tecnológico atual, afeta diretamente o mercado de software que deve responder de maneira rápida e com alta qualidade.

Para lidar com este ritmo incessante de mudanças, empresas de software estão se adaptando para lidar com essa demanda investindo em inovação e tendo cuidados especiais na disseminação da cultura necessária para realização dos objetivos. Neste sentido, conforme Highsmith e Cockburn (2001), diversas abordagens de processos de software surgiram tendo em vista que a mudança é um paradigma que reflete o turbulento cenário de negócio atual. Dentre elas podemos citar *Extreme Programming*, o *framework Scrum* e o desenvolvimento de software *Lean*. Elas fazem parte de um conjunto de práticas que Highsmith e Cockburn (2001) classificam de Métodos Ágeis. Os autores acrescentam ainda que o termo *ágil* possui duas conotações: a primeira é a idéia de que o mundo da tecnologia se tornou turbulento, veloz, incerto e requerendo resposta rápida a mudanças; a segunda, que é uma implicação da primeira, sugere que os processos requeridos necessitam de pessoas e organizações *ágeis*. Portanto, desenvolvimento ágil foca no talento e habilidades individuais adaptando processos específicos para times e pessoas.

O caminho a ser percorrido para consolidar a cultura requerida por métodos ágeis não é simples. Ao desenvolver equipes e pessoas é necessário que se estabeleça uma intenção de longo prazo, (Poppendieck, 2003) mas para isso é necessário oferecer um ambiente motivador e desafiador. Conforme Curtis et al. (2001), na economia do conhecimento as empresas

competem em dois mercados, um com seus produtos e serviços e outro para contratar e desenvolver talentos. Para vencer esse desafio é necessário que organizações estabeleçam procedimentos eficientes no que diz respeito a contratação, desenvolvimento de equipes, otimização do ambiente de trabalho, cultura corporativa dentre outras. São esses justamente alguns dos assuntos abordados pelo *People Capability Maturity Model* (P-CMM), que tem como objetivo melhorar a capacitação das pessoas através de um conjunto de práticas de gestão que desenvolva a força de trabalho para atuar como um centro independente de inteligência sem, contudo, desviar-se da estratégia e dos objetivos organizacionais (Curtis et al., 2001).

O presente projeto de pesquisa visa estudar os conceitos do desenvolvimento de software *lean* e como é tratada a questão de gestão de equipes através dessa abordagem. Pretende-se verificar também como o P-CMM poderia ser utilizado para estabelecer uma estratégia de longo prazo para desenvolvimento de equipes. Busca-se então verificar se existem relações significativas entre as práticas de desenvolvimento de software *lean* e as práticas sugeridas pelo P-CMM.

Assim, a questão a ser respondida por este estudo é a seguinte:

“É possível executar uma estratégia de gestão de equipes baseadas nas práticas do People CMM em uma organização que adota conceitos lean para desenvolvimento de software?”

1.2 PROBLEMA DE PESQUISA

Liker (2008), afirma que o *Lean* baseado no Modelo Toyota, envolve uma transformação cultural muito mais profunda e mais abrangente do que a maioria das empresas podem imaginar. A mudança cultural dentro da organização também deve ser levada em consideração quando aplicada na estrutura organizacional. Estudos sobre a adoção de práticas ágeis no processo de desenvolvimento em grandes empresas tem se demonstrado um grande desafio. Conforme Lindvall et al 2004, este vai além da simples adoção das práticas mas também diz respeito à integração com os processos já existentes. Conforme o autor, grande parte do desafio se concentra em fazer o projeto ágil funcionar bem dentro do ambiente organizacional.

Segundo Talby et al, 2006, apenas empresas que adotaram processos ágeis na sua integralidade – que então determinaram mudanças organizacionais radicais – reportaram resultados substanciais. Conforme mencionado anteriormente, as empresas já estão atentas sobre a necessidade de atentar para a gestão de pessoas e equipes quando novos processos e melhorias são introduzidos.

É neste contexto que o presente projeto de pesquisa visa estudar a adoção dos princípios *Lean* e como o People CMM pode suportar uma estratégia de gerencia de equipes alinhada com as práticas *Lean* para desenvolvimento de software.

1.3 OBJETIVOS

Os objetivos do presente trabalho estão apresentados a seguir.

1.3.1 Objetivo Geral

O objetivo principal deste projeto é estudar a possibilidade de gerenciar times de desenvolvimento *lean* através das práticas sugeridas pelo P-CMM na perspectiva de equipes de trabalho.

1.3.2 Objetivos Específicos

- Estudar o conceito de *Lean* aplicado à processos de desenvolvimento de software;
- Identificar os fatores relevantes para a gerencia de equipes de desenvolvimento *lean*;
- Analisar como práticas ágeis de desenvolvimento e o processo proposto pelo *framework* Scrum se adéquam as áreas de processo do P-CMM, na perspectiva de construção de grupos de trabalho e cultura em seus diferentes níveis de maturidade.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ORIGENS DO *LEAN THINKING*

Durante o período de reconstrução após a Segunda Guerra Mundial, a indústria japonesa, de modo geral, enfrentava grandes problemas em produzir devido à grande expansão do mercado consumidor, falta de recursos e severa restrição de crédito imposta pelas forças de ocupação (Harvey, 2004). Nesta época, imperava a economia de grande escala ocidental encontrada principalmente nas grandes empresas dos Estados Unidos.

A japonesa Toyota, que então desenvolveu seu processo conhecido como Sistema Toyota de Produção (STP), possuía condições empresariais muito diferentes das americanas Ford e GM. Enquanto estas se utilizavam da produção em massa, economias de escala e grandes equipamentos para produzir o máximo possível de peças com o menor custo possível, a Toyota no Japão pós-guerra tinha a mesma exigência de produzir uma variedade de veículos na mesma linha, porém para um mercado substancialmente menor (Liker, 2008). A chave para sua produção era a flexibilidade. Dessa forma a Toyota descobriu que quando se reduz o *lead time*¹ e concentra-se em manter flexíveis as linhas de produção, realmente se obtém uma maior qualidade, melhor resposta dos clientes, melhor produtividade e melhor utilização dos recursos.

Taiichi Ohno, fundador do STP (Liker, 2008), resumidamente definiu seu processo de produção da seguinte forma:

O que estamos fazendo é observar a linha de tempo desde o momento em que o cliente nos faz um pedido até o ponto em que recebemos o pagamento. E estamos reduzindo essa linha de tempo, removendo as perdas que não agregam valor. (Ohno, 1988)

¹ *Lead time* é o período entre o início de uma atividade, produtiva ou não, e o seu término. No contexto de logística, é o tempo entre o momento de entrada do material até a sua saída do inventário. (Wikipedia)

Para Ohno, segundo Liker (2008), para ser uma indústria *Lean* é preciso um modo de pensar que se concentre em fazer o produto fluir através de processos ininterruptos de agregação de valor, um sistema puxado que parta da demanda do cliente, reabastecendo somente o que a operação seguinte for consumir em curtos intervalos, e uma cultura em que todos lutem continuamente para a melhoria.

Conforme Harvey, 2004, Ohno claramente direcionou o foco em obter o melhor com investimentos limitados através das seguintes premissas:

- construir somente o que é necessário: não se deve produzir grandes quantidades de inventário que irão esperar até serem usados;
- eliminar qualquer coisa que não agregue valor: o que significa definir exatamente o que a empresa e seus clientes definem como “valor”;
- parar a produção se algo estiver acontecendo de errado: caso se produza apenas o que é necessário em cada estágio, será possível identificar um defeito muito próximo do ponto em que ele ocorre.

Womack e Jones, que documentaram em 1990 a experiência de produção *Lean* na indústria americana através do livro *The Machine that Changed the World*², identificaram cinco princípios nas organizações que produziam ou entregavam de forma *Lean*:

- Definir o que é valor;

² James P Womack, Daniel T Jones, Daniel Roos, *The Machine that Changed the World*, New York, 1990

- Identificar a cadeia de valor – encadear as atividades que agregam valor e eliminar as que não agregam;
- Criar as condições para o valor fluir através da cadeia;
- Ter o cliente puxando o valor na cadeia;
- Buscar a perfeição trabalhando nas responsabilidades que o sistema tem para atender a demanda de valor do cliente.

Complementando a definição, Poppendieck (2002) acrescenta ainda que empresas que repensam a cadeia e encontram formas de criar o valor esperado pelo cliente, com menos recursos, podem desenvolver um diferencial competitivo que as tornam referência para seus concorrentes, que ainda assim serão incapazes de entregar a mesma proporção de valor.

Na concepção de Bowen e Youngdahl (1998) apud Fortes (2010), que apresentaram o conceito de *lean* aplicados a área de serviços (*lean service*) duas outras importantes características são apresentadas:

- Foco no cliente e treinamento crescente: envolvimento do cliente no desenho dos serviços, treinamento dos empregados no perfil de serviços e treinamentos para clientes sobre como contribuir a qualidade do serviço;
- Autonomia dos empregados: investimento significativo nos empregados (perfil, formação de equipes, participação) e autonomia para alavancar a equação de valor do cliente (benefícios divididos pelo preço e outros custos).

2.2 LEAN THINKING NO DESENVOLVIMENTO DE SOFTWARE

O presente capítulo apresenta como os princípios do *Lean* foram e estão sendo aplicados na empresas de software.

2.2.1 O problema em satisfazer as necessidades de negócio

Mah (2000) aponta estudo realizado pela QSM *Associates* que identificou que em 47% dos projetos analisados não foi possível medir a execução do projeto e o custo do esforço contra o planejamento inicial porque o plano havia mudado muitas vezes. O estudo foi realizado em mais de 200 projetos de 24 empresas de diferentes indústrias incluindo varejo, manufatura, finanças e serviços. Conformidade com o plano não foi o objetivo principal. Ao invés disso, satisfazer o cliente – ao advento da entrega e não da iniciação do projeto – foi mais importante. Em muitos casos, mudanças nos requerimentos, escopo e tecnologia freqüentemente estão fora dos limites de controle do time de desenvolvimento durante o ciclo de vida projeto.

Boehm (1987) afirma que “encontrar e arrumar um defeito no software após a entrega custa cerca de 100 vezes mais do que encontrá-lo e arrumá-lo nas fases iniciais de *design*”. Esta premissa foi uma das principais justificativas para que as abordagens tradicionais focassem nas fases de análise de requisitos e *design* para poder antecipar o conjunto de requerimentos, reduzindo-se assim a possibilidade de mudança. Apesar disso, Highsmith e Cockburn (2001) mencionam que a questão não é se preocupar em estancar as mudanças cedo no projeto, mas sim se preocupar em como lidar com a mudança inevitável durante o ciclo de vida do projeto. No contexto atual, segundo Highsmith e Cockburn (2001), eliminar a mudança

cedo significa ser irresponsável com as condições de negócio, em outras palavras, é uma falha de negócio.

Contudo, a questão não é apenas adequar a mudança, mas sim abraçá-la de forma que a qualidade seja mantida, já que o mercado demanda inovação, software de alta qualidade e que satisfaça as necessidades de negócio de forma rápida.

2.2.2 Complexidade envolvida no desenvolvimento de software

Apesar do *lean thinking* ter sua origem nas linhas de produção seus princípios são amplamente aplicáveis à outras disciplinas (Poppendieck, 2003). Contudo, práticas da produção *lean* não podem ser transcritas diretamente da planta de produção para o desenvolvimento de software. Poppendieck (2003) salienta que os que tentaram fazê-lo simplesmente não obtiveram sucesso porque criar um bom software não é um processo de produção e sim um processo de desenvolvimento. Pensar em desenvolvimento, segundo o autor, é como criar uma receita e produção significa seguir essa receita. Desenvolver uma receita é um processo de aprendizado envolvendo tentativa e erro. Não se pode esperar atingir o melhor na primeira vez. Em realidade, a idéia principal é tentar variações do tema e descobrir qual a melhor opção. A seguinte tabela apresenta as principais diferenças entre produção e desenvolvimento.

Desenvolvimento	Produção
- A qualidade é determinada pelo uso	- Qualidade é conformidade aos requisitos
-Resultado variável pode ser considerado bom	- Resultado variável não é bom

- Iteração gera valor	- Iteração gera desperdício (retrabalho)
-----------------------	--

Tabela 1 - Desenvolvimento versus produção. Adaptado de Poppendieck (2003)

2.2.2.1 Processo de controle empírico

Muitos dos processos empregados para resolver problemas da sociedade, em geral acabam por funcionar devido ao nível de aceitação da imprecisão no resultado. Na indústria automobilística, por exemplo, podemos dizer que o alinhamento dos pneus, balanço dos cilindros e tremor do freio ocorrem em um nível insignificante ao dirigir um automóvel. Carros são produzidos de forma que as peças são ajustadas para um nível de precisão proposto. Conforme Schwaber (2004), repetir o processo onde o resultado sempre satisfaz o nível de precisão aceitável é o que caracteriza *controle de processo definido* ou *processo determinístico*. Quando o controle de processo definido não pode ser utilizado devido à complexidade das atividades intermediárias o autor define que o *controle de processo empírico* deve ser utilizado.

Schwaber (2004) acrescenta ainda que controle de processo definido tende a resultar em produtos que podem ser precificados como *commodity*³. Por outro lado, quando a qualidade resultante está a abaixo da expectativa de mercado e o custo para tornar o produto aceitável é muito alto, o uso e a aceitação do alto custo do controle de processo empírico deve ser levado em consideração. Além disso, o autor salienta que em longo prazo, obter sucesso com controle de processo empírico nas primeiras tentativas torna o custo mais barato que o retrabalho resultante de um produto sem sucesso.

³ *Commodity* é um termo de língua inglesa que, como o seu plural *commodities*, significa mercadoria, é utilizado nas transações comerciais de produtos de origem primária nas bolsas de mercadorias. (Wikipedia)

No quadro abaixo é apresentado os três pilares que sustentam qualquer implantação de controle de processos empíricos conforme Schwaber (2004): transparência, inspeção e adaptação.

Transparência	Garante que aspectos do processo que afetam o resultado devem ser visíveis para aqueles que gerenciam o resultado. Não somente serem vistos como também devem ter significado. Por exemplo: em desenvolvimento de software, classificar certa funcionalidade como “pronta” pode significar que código foi revisado, <i>refactoring</i> ⁴ realizado e funcionalidade testada e aceita.
Inspeção	A inspeção permite que variações inaceitáveis no processo possam ser detectadas. Para determinar a frequência da inspeção é necessário levar em consideração que o próprio ato de inspecionar altera o processo. Outro fator importante é o inspetor que deve possuir domínio sobre aquilo que está sendo inspecionado
Adaptação	Caso um ou mais aspectos do processo sejam identificados como fora dos limites aceitáveis ou que o produto resultante será inaceitável, a adaptação deve ser feita o mais rápido possível para minimizar desvios posteriores.

Tabela 2 - Pilares de uma implantação de controle de processo empírico.

⁴ *Refactoring* é o processo de modificar um sistema de software para melhorar a estrutura interna do código sem alterar seu comportamento externo. (Wikipedia)

2.2.2.2 Dimensões da complexidade

Geralmente a complexidade de um sistema é medida de acordo com o número de entidades e as relações que elas possuem com as outras partes do sistema. Schwaber (2004) limita sua definição de complexidade levando em consideração as dimensões que considera as mais significantes: requisitos, tecnologia e pessoas.

É possível criar software simples em casos onde os requisitos propostos pelo cliente estão bem definidos e se mantêm constantes, sem a necessidade de mudanças, revisões ou modificações de última hora. Levando em consideração o cenário de negócio atual, esta não é a realidade comumente encontrada. Geralmente os *stakeholders* possuem necessidades diferentes. Em muitos casos, segundo Schwaber (2004), clientes somente começam a entender de fato a necessidade após obterem evidências concretas do que já foi estabelecido até então. São requisitos complexos porque, não apenas são ambíguos, mas também sofrem constantes mudanças de negócio.

Na dimensão de tecnologia é colocado o grau de complexidade das relações dos diversos componentes de software.

Na dimensão das pessoas são apresentados os aspectos pessoais que influenciam no desempenho do trabalho. Todos possuem diferentes habilidades, experiências, pontos de vista e atitudes. Quando começam a trabalharem juntas, o grau de complexidade aumenta consideravelmente.

2.2.3 Importância do aprendizado através de *feedback*

Kajko-Mattson et al. (2001) mostram no estudo realizado que cerca de 40% a 90% do custo durante o ciclo de vida de um projeto é gasto na fase de manutenção. Muitas empresas e times de desenvolvimento não vêm com bons olhos mudanças de escopo que acabam como o esforço despendido na fase de planejamento. No entanto, conforme já exposto anteriormente, existirá uma forte tendência de o cliente mudar a prioridade ou o próprio requisito.

Levando-se em conta este ambiente instável, o emprego de uma abordagem determinística (processo definido), tende a não ser a melhor escolha. Pensando em reduzir o custo das adaptações, empresas determinam os detalhes do projeto previamente através de detalhamento de requisitos e *designs* antecipados para as fases de planejamento. Esta abordagem acaba condicionando os clientes a pedirem tudo o que acham necessário, especialmente se acham que receberão o produto todo de uma só vez.

Tal abordagem acaba por gerar grande desperdício na maioria dos casos. Isto é exemplificado pelo famoso estudo chamado *CHAOS Report*, realizado pelo Standish Group⁵, que no ano de 2002 apresentou que 45% das funcionalidades desenvolvida num sistema nunca são usadas e 19% são raramente utilizadas. Através deste dado pode-se inferir que, freqüentemente, clientes não sabem exatamente o que eles querem nas fases iniciais do projeto. Esta influência na dimensão dos requisitos acaba influenciando o risco no projeto. A melhor estratégia, segundo Poppendieck (2003), é evitar generalizações desnecessárias e fazer com que o sistema seja flexível apenas nas áreas mais propícias à mudanças.

⁵ Ibid.

O modelo determinístico é favorecido pelas disciplinas de gerência de projetos que possuem suas origens na administração de contratos, onde as especificações são feitas no início e a gerencia de escopo e de mudanças tendem a minimizar desvios no plano inicial. A abordagem por processo definido acaba por considerar *feedbacks* iterativos uma ameaça, pois muito provavelmente o aprendizado poderá modificar o planejamento feito previamente. Por fim, acrescentar controles determinísticos num ambiente dinâmico tende a alongar o tempo de resposta, o que pode ser tarde quando os problemas vierem a tona. A tabela a seguir apresenta as principais características das duas abordagens para desenvolvimento de software.

Abordagem determinística/processo definido	Abordagem empírica
<ul style="list-style-type: none"> - documentação detalhada dos requisitos; - documentação e ata de reuniões feitas com clientes para oficializar acordos; - rígido controle de mudanças; - rastreamento entre o código desenvolvido e os requerimentos. 	<ul style="list-style-type: none"> - ao invés de coletar mais requisitos dos usuários mostre ao cliente a variedade de potenciais telas do sistema obtendo a sua opinião; - ao invés de adicionar mais documentação ou planejamento detalhado valide idéias escrevendo código; - ao invés de aguardar a validação da entrega de uma etapa ao cliente para iniciar as correções execute testes tão cedo o código seja escrito;

Tabela 3 - Características das abordagens determinística e empírica

2.2.4 O desperdício no desenvolvimento de software

Aprender a identificar o desperdício, segundo Poppendieck (2003), é o primeiro passo para atingir resultados satisfatórios através da implementação do *Lean Thinking*. Taiichi Ohno, criador do Sistema Toyota de Produção (Liker, 2008), claramente define como desperdício qualquer atividade executada durante o processo que não agregue valor. Em outras palavras, é o que resulta algo que seja percebido pelo cliente como algo de valor.

Royce (1970) escreve em seu artigo que os passos fundamentais para desenvolver software são análise e codificação. O autor acrescenta ainda que muitos passos adicionais de desenvolvimento são requeridos, porém nenhum deles contribui tanto para o produto final quanto análise e codificação e, além disso, eles contribuem diretamente para o aumento de custos durante o desenvolvimento. Seguindo a definição de desperdício apresentada a cima, Poppendieck (2003) acrescenta que, conforme os comentários de Royce (1970), claramente pode-se perceber que todos os passos do modelo em cascata⁶, exceto análise e codificação, são desperdício.

Shingeo Shingo apud Liker (2008), durante seus estudos sobre a aplicação do STP na indústria japonesa, identificou sete tipos principais de desperdício durante a manufatura. Na tabela a baixo é apresentado a tradução para o desenvolvimento de software segundo Poppendieck (2003).

⁶ O modelo em cascata é um modelo de desenvolvimento de software seqüencial no qual o desenvolvimento é visto como um fluir constante para frente (como uma cascata) através das fases de análise de requisitos, projeto, implementação, testes (validação), integração, e manutenção de software. A origem do termo cascata é freqüentemente citado como sendo um artigo publicado em 1970 por W. W. Royce; ironicamente, Royce defendia um abordagem iterativa para o desenvolvimento de software e nem mesmo usou o termo cascata. Royce originalmente descreve o que é hoje conhecido como o modelo em cascata como um exemplo de um método que ele argumentava *ser um risco e um convite para falhas*. (Wikipedia)

Desperdícios em manufatura	Desperdícios em desenvolvimento de software
Excesso de estoque	Trabalho parcialmente finalizado
Superprocessamento	Processos extras
Superprodução	Funcionalidades a mais
Transporte	Troca de tarefas
Espera	Espera
Movimento	Movimento
Defeitos	Defeitos

Tabela 4 - Comparação do desperdício na manufatura e no desenvolvimento de software

A seguir é descrito cada um dos desperdícios em desenvolvimento de software.

Trabalho parcialmente finalizado

Trabalho parcialmente finalizado tem a tendência de se tornar obsoleto e pode prejudicar o andamento de tarefas que provavelmente precisam ser feitas. Porém o principal problema é que com software parcialmente finalizado não se pode ter certeza de que realmente irá funcionar. A equipe de desenvolvimento pode até ter documentos detalhados de requisitos e *design*, código já criado e até mesmo testes unitários realizados. Contudo, até que o software esteja integrado com o restante do ambiente não se pode conhecer todos os problemas que podem ocorrer e se realmente irá solucionar o problema de negócio.

Processos extras

Processos de software geralmente requerem que a documentação tenha a aprovação do cliente, ou que forneça rastreabilidade dos requerimentos desenvolvido ou precisam ser criados para aprovar uma mudança. Caso seja necessário produzi-lo é importante criá-lo em alto nível e resumido. Pode-se também criar padrões que reduzem os requerimentos para um formato onde ambos os usuários e desenvolvedores podem rapidamente entender e validar.

O fato de o documento ser um artefato de software não o torna valioso. Se a equipe de desenvolvimento tem que esperar que um documento seja produzido para que daí então inicie o trabalho, a necessidade do mesmo deve ser reavaliada.

Funcionalidades a mais

Pode parecer uma boa idéia entregar ao cliente mais funcionalidades do que o previsto para surpreendê-lo. Mais funcionalidade significa mais código para ser mantido, testado, especificado e documentado.

Troca de tarefas

Envolver uma pessoa em múltiplos projetos tende a ser uma fonte de desperdício. Por ser uma atividade essencialmente intelectual, é necessário, toda vez que ocorrer a troca entre tarefas de projetos diferentes, que o desenvolvedor foque em algo completamente novo.

Pertencer a múltiplos times geralmente causa mais interrupções e então mais tempos para a troca de tarefas.

Espera

Uma das maiores fontes de desperdício ocorre quando se deve esperar para que algo seja produzido. Atrasos distanciam o usuário de receber um produto que satisfaça sua necessidade.

Movimento

Desperdício com movimento geralmente ocorre quando alguma pessoa gasta muito tempo para encontrar uma resposta, quando os responsáveis não estão à disposição para responder uma dúvida técnica ou para esclarecer questões de negócio. Esta é a razão pela qual práticas ágeis geralmente recomendam que o time ocupe uma única sala onde todo mundo tenha acesso aos desenvolvedores, testadores e representantes do cliente. Artefatos como os requerimentos, por exemplo, podem mover-se dos analistas para os arquitetos e então para o desenvolvedor, que por sua vez encaminham para a equipe de teste. O desperdício ocorre pois a documentação não contém toda a informação que a próxima pessoa necessita saber. Grandes porções de conhecimento tácito permanecem com o criador do documento.

Defeitos

A quantidade de desperdício causado por um defeito pode ser determinado pelo seu impacto no produto e por quanto tempo ele permanece desconhecido. Um defeito crítico que é identificado em três minutos não é uma grande fonte desperdício. Um defeito pequeno que não é descoberto em semanas é um desperdício muito maior. A forma de reduzir o impacto dos defeitos é encontrá-los o mais cedo possível eles ocorram através de testes imediatos, integração freqüente e liberação para produção o mais cedo que possível.

2.2.5 Eliminando o desperdício através do mapeamento da cadeia de valor

Poppendieck (2003) indica que fazer o mapeamento da cadeia de valor é uma boa maneira de descobrir o desperdício no processo de desenvolvimento. Através deste trabalho a organização esclarece qual o seu conceito sobre o que é valor agregado para seus clientes. Poppendieck (2003) acrescenta ainda que mapear é um exercício de desenhar no papel o que um gerente pode observar por si próprio na empresa. O importante neste trabalho, segundo o autor, é quem envolvam diretamente as pessoas responsáveis por cada atividade. O objetivo principal é desenhar em alto nível o processo desde a requisição por parte do cliente até a entrega, levando em consideração o tempo investido nas atividades que agregam ou não valor e o tempo de espera entre as etapas.

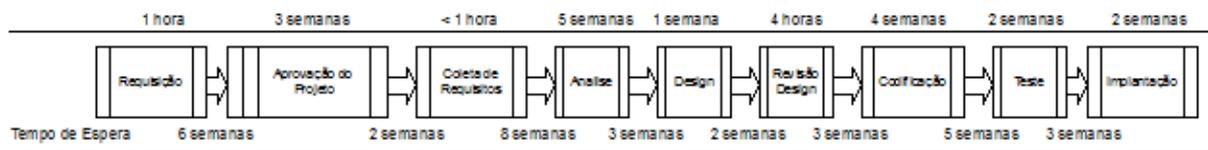


Figura 1 - Mapa da cadeia de valor tradicional. Adaptado de Poppendieck (2003)

A figura acima apresenta a forma tradicional de mapear a cadeia. Supondo que a empresa decida seguir as práticas ágeis, Poppendieck (2003) apresenta a seguinte crítica:

- o modelo *as-is*⁷ indica que o processo de aprovação deve ter seu tempo diminuído. A gerência acorda então que as decisões sobre novas requisições devem ser comprometidas semanalmente. O time estabelece que resposta rápida para as requisições dos clientes é prioridade. Dessa forma, é acordado aprovar somente as requisições que podem ser desenvolvidas com os recursos disponíveis sem a

⁷ Traduz-se como o desenho que representa a situação atual de um processo

necessidade de subcontratar recursos adicionais. Os recursos serão gerenciados de forma que o time de *design* esteja comprometido dentro de uma semana e que o restante da equipe (analistas e desenvolvedores) sejam comprometidos dentro de três semanas.

- o mapa inicial indica que a assinatura do contrato pelo cliente pode se tornar uma fonte tanto de desgaste quanto de demora; mostra também que a revisão do *design* com o cliente deve ser conduzida simultaneamente com o desenvolvimento, já que hoje é uma grande fonte de tempo em espera. Dessa forma, o mapa indica que o planejamento do desenvolvimento deva ocorrer cedo no processo. Uma vez que a equipe decida por um desenvolvimento ágil, os problemas serão resolvidos desenvolvendo em ciclos incrementais, coletando requerimento somente quando necessário, integrando revisões de *design* e código e planejando cedo a colocação do projeto em uso.

Através desta análise Poppendieck (2003) sugere que a cadeia de valor ágil deve seguir o seguinte modelo proposto a seguir:

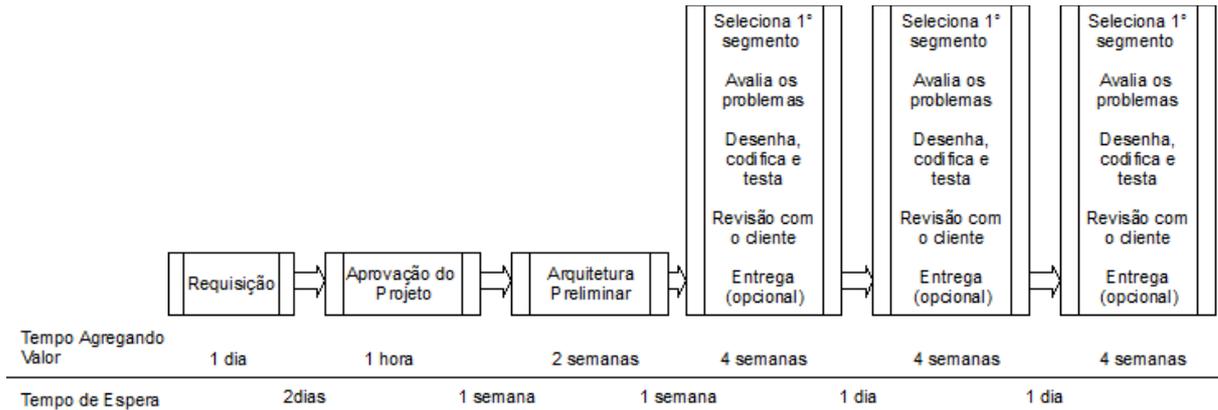


Figura 2 - Mapa da cadeia ágil. Adaptado de Poppendieck (2003)

2.2.6 A proposta do *framework* Scrum

O Scrum vem sendo utilizado para desenvolvimento desde o início dos anos 90. Ele constitui um *framework* dentro do quais diversos outros processos e técnicas podem ser utilizados em conjunto. É fundamentado na teoria de controle de processos empíricos e emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e controlar riscos. O Scrum foi desenvolvido para satisfazer as três dimensões descritas anteriormente para implantação de processos de controle empírico. Levando em conta os processos a seguir podemos dizer que se enquadram perfeitamente nos conceitos de desenvolvimento *lean* descritos até o momento.

O *framework* consiste em um conjunto formado por Times Scrum e seus papéis associados, eventos com duração fixa (*time-boxes*), artefatos e regras.

As **regras** fazem o elo entre os eventos com duração fixa, os papéis e os artefatos e estão incluídas dentro das descrições de cada um dos elementos do *framework*.

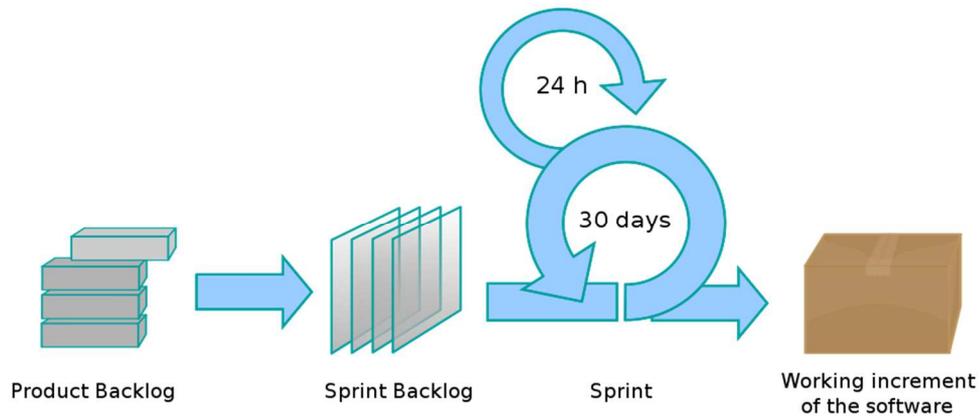


Figura 3 - Processo Scrum

Time Scrum é composto pelo Scrum Master, pelo Product Owner e pelo Time. O Time é responsável por transformar o *Backlog*⁸ do Produto em incrementos de funcionalidades potencialmente entregáveis em cada *Sprint*⁹.

A tabela a seguir apresenta o papel do Scrum Master e do Product Owner.

Scrum Master	<ul style="list-style-type: none"> • Responsável por garantir que o Time Scrum esteja aderindo aos valores do Scrum, às práticas e às regras; • Auxilia na organização e adoção do Scrum; • Ajuda a entender e a usar autogerenciamento e interdisciplinaridade;
--------------	---

⁸ *Backlog*: Lista com requisitos do produto que o time está desenvolvendo. O *Product Owner* é responsável pelo *Backlog* do Produto, seu conteúdo, por sua disponibilidade e por sua priorização. Ele nunca está completo. A seleção inicial somente mostra os requisitos inicialmente conhecidos e melhor entendidos; é dinâmico no sentido de que está sendo constantemente atualizado para identificar o que se necessita para ser apropriado, competitivo e útil. Guia do Scrum, 2009.

⁹ *Sprint*: é uma iteração com eventos de duração fixa. O Scrum Master garante que não será feita nenhuma mudança que possa afetar a meta da Sprint e durante seu andamento tanto a composição do time quanto as metas de qualidade devem permanecer constantes. É composta pela Reunião de Planejamento, o trabalho de desenvolvimento, a Revisão e a Retrospectiva da Sprint. Ocorrem uma após a outra, sem intervalos entre elas.

	<ul style="list-style-type: none"> • Responsável pela remoção de impedimentos que possam aparecer durante o desenvolvimento do incremento de funcionalidade.
Product Owner	<ul style="list-style-type: none"> • Responsável pelo gerenciamento do <i>Backlog</i> do Produto e por garantir o valor do trabalho realizado pelo Time; • Mantém o <i>Backlog</i> do Produto, prioriza os itens e garante que esse esteja visível a todos.

Tabela 5 - Papéis do Scrum Master e do Product Owner

Mais detalhes sobre o Time Scrum são apresentados na seção 2.3.4.4.

Os seguintes itens compõe os **eventos de duração fixa**: Reunião de Planejamento da Release, Sprint, Reunião de Planejamento da Sprint, Revisão da Sprint, Retrospectiva da Sprint e Reunião Diária.

O Propósito da **Reunião de Planejamento da Release** é o de estabelecer um plano de metas que o Time Scrum e o resto da organização possam entender e comunicar. O plano estabelece a meta, as maiores prioridades do Backlog do Produto, os principais riscos, características gerais e funcionalidades que estarão contidas na release. Ele estabelece também uma data de entrega e custo provável se nada mudar. A organização pode então inspecionar o progresso e fazer mudanças nesse plano a cada Sprint. De modo geral, o planejamento requer estimar e priorizar o Backlog do Produto e para isso, existem diversas técnicas que estão fora do alcance do Scrum.

O tempo ideal para uma **Sprint** não deve ultrapassar um mês. O principal argumento para isso é que cada projeto de software possui um horizonte e este é o período de tempo para o qual o plano é válido. Se o horizonte for longo demais, mudanças poderão ocorrer na definição inicial e variáveis não previstas poderão aparecer aumentando assim o risco de insucesso. O Scrum é um *framework* para projetos cujo horizonte não é superior ao período de um mês, onde

já há complexidade suficiente para gerar insucesso. Com isso, o risco de que o projeto saia de controle ou se torne imprevisível é contido pelo menos num período não maior que um mês. Busca-se então dessa forma aumentar a previsibilidade e controlar os riscos do projeto.

A **Reunião de Planejamento da Sprint** é o momento no qual a iteração é planejada. É fixada em oito horas de duração para uma Sprint de um mês. Caso seja mais curta, aloca-se para essa reunião aproximadamente 5% do tamanho total da Sprint. As entradas para essa reunião são o Backlog do Produto, o incremento mais recente ao produto, a capacidade e o histórico de desempenho do Time.

Na **Revisão da Sprint** o time e as partes interessadas colaboram sobre o que acabou de ser feito. O time discute sobre o que ocorreu bem e quais problemas foram enfrentados, além de como esses problemas foram resolvidos. O time demonstra o trabalho que está pronto e responde a questionamento. O ideal é apresentar as funcionalidades desenvolvidas através de protótipos ou versão *demo*.

A finalidade da **Retrospectiva da Sprint** é de inspecionar como ela ocorreu em se tratando de pessoas, das relações entre elas, dos processos e das ferramentas. No final, o time deve identificar medidas de melhoria factíveis que podem se incorporar à próxima Sprint. Tais mudanças se tornam a adaptação resultante da inspeção empírica.

A **Reunião Diária** cada membro do time explica:

- o que realizou desde a última reunião diária;
- o que ele vai fazer antes da próxima reunião diária; e
- quais obstáculos estão em seu caminho.

Tais itens servem para inspecionar o progresso na direção da Meta da Sprint. Elas melhoram a comunicação, eliminam outras reuniões e removem impedimentos para o desenvolvimento. É importante salientar que não é uma reunião de status. Ela é só para as pessoas que estão transformando itens do Backlog em incremento funcional (o time).

2.3 GESTÃO DE PESSOAS NO CONTEXTO *LEAN*

As seções a seguir apresentam conceitos relacionados a motivação de equipes de trabalho, como o modelo *Lean* evolui para adequar-se as necessidades após a era da industrialização e apresenta uma comparação entre o gerenciamento de equipes da abordagem empírica e determinística.

2.3.1 Rompendo os limites da Administração Científica

Conforme já exposto anteriormente, as condições empresariais da indústria japonesa durante a criação dos fundamentos *Lean* de produção eram bem diferentes da indústria americana que, à época, usufruía dos princípios da Administração Científica introduzidos por Frederick Taylor¹⁰. A idéia principal de Taylor aplicada às plantas de produção era dividir todo o conjunto de atividades em pequenas partes e descrever como cada uma das tarefas deveria ser executada. Desta forma, era necessário um grande número de engenheiros para determinar o tempo e descrever os passos de cada atividade.

Taylor descreve que o sistema de trabalho que vigorava até o advento da administração científica era o sistema de iniciativa e incentivo. O operário ficava livre para definir por si próprio os métodos mais econômicos para a realização do trabalho. A crítica de Taylor estava no fato de que a administração não poderia depender da iniciativa operária, o controle do trabalho e de todo processo produtivo deveria permanecer nas mãos do administrador. Dessa forma, a política de incentivo salarial ideal era remunerar o funcionário proporcionalmente ao número de unidades produzidas. O objetivo da boa administração,

¹⁰ *Princípios da Administração Científica* foi originalmente publicado por Taylor em 1911.

segundo Taylor, era pagar altos salários para que produzissem mais, mantendo assim o baixo custo unitário de produção.

Durante a década de 80, tornou-se aparente que as técnicas de manufatura introduzidas pela Toyota, poderiam produzir produtos de alta qualidade mais rápidos e baratos dos que as técnicas da administração científica (Liker, 2008). Através das teorias operacionais e motivacionais colocadas em questão, administradores começaram a explorar além dos limites impostos pela administração científica. Começou assim a surgir uma série de programas como MBO (*Management by Objectives*), TQM (*Total Quality Management*) e ISO9000. Esses por sua vez, não produziam grandes resultados quando não mudavam a realidade de onde trabalho era feito. Frequentemente, aumentavam a intensidade de fatores desmotivacionais (políticas, supervisão e controle) ao invés de contribuírem com os fatores de motivação (realização, reconhecimento e responsabilidade)¹¹. Apesar disso não ser uma falha dos programas, Poppendieck (2003) coloca que eram efeitos comuns que se verificavam após sua implantação.

2.3.2 Diferença entre os modelos *Lean* e CMM/CMMI

O CMM (*Capability Maturity Model*) tem sido usado como um programa de certificação assim como a ISO9000, especialmente por empresas de desenvolvimento de software que buscam negócios em outros países. Ambos CMM e ISO9000 tendem a gerar impactos parecidos criando burocracia e tornando a mudança difícil, ainda que o propósito de ambos não seja esse.

¹¹ Para Herzberg, fatores motivacionais são aqueles que se referem ao conteúdo do cargo, às tarefas e aos deveres relacionados com o cargo em si. O termo motivação envolve sentimentos de realização, de crescimento e de reconhecimento profissional, manifestados por meio do exercício das tarefas e atividades que oferecem um suficiente desafio e significado para o trabalhador.

Uma implantação de CMM focada em documentação e conformidade com determinadas práticas tende a remover a responsabilidade dos desenvolvedores no processo de *design* e de tomada de decisão e direcionam para a administração organizacional. O papel do desenvolvedor é algo que transita entre o administrador e algo parecido com a função do engenheiro da época da administração científica, que conhecia uma melhor prática para desempenhar suas tarefas. O *Lean Thinking*, conforme Poppendieck (2003), investe na inteligência dos trabalhadores da linha de frente e acredita que eles são os que devem determinar e melhorar continuamente a forma como executam seu trabalho. Watts Humphrey, o qual liderou os primeiros esforços do CMM (apud Poppendieck, 2003), acredita que no desenvolvimento de software não se pode obter sucesso sem disciplina e pessoas motivadas. No ambiente *Lean*, acredita-se que o fator crítico de motivação não é recompensar pelo desempenho¹², mas sim transferir a autoridade para o time, movendo as decisões para o menor nível da organização enquanto desenvolve a capacidade dessas pessoas para tomarem decisões sabiamente.

Em 2003, após desenvolver diversos modelos de maturidade, o *Software Engineering Institute* (SEI) combinou todos dentro do CMMI, o que promoveu uma descrição genérica e única para o conceito de *maturidade* para desenvolvimento de software, engenharia de sistemas, desenvolvimento de produtos entre outros. A definição de maturidade do CMMI é baseada em duas suposições:

¹² Rob Autin, autor do livro “*Measuring and Managing Performance in Organizations*, 109-110, discute o porque práticas baseadas na medição de desempenho como o MBO (gerenciar por objetivos) e tentativas de avaliar a capacidade são largamente inapropriadas para trabalhadores do conhecimento..

- Suposição 1: um sistema é melhor gerenciado quando desagregado em produtos de trabalho identificáveis que são transformados de um estado de entrada para um estado de saída para satisfazer metas específicas¹³.
- Suposição 2: uma organização madura é aquela a qual tudo é cuidadosamente planejado e então controlado para executar o plano.

Para Poppendieck (2003), estas suposições soam parecidas com os conceitos da administração científica. Conforme o autor, uma organização *Lean* possui um modelo diferente do que maturidade significa:

- Suposição *Lean* 1: uma organização madura olha para o sistema como um todo; não é o foco otimizar partes desagregadas.
- Suposição *Lean* 2: uma organização madura foca no aprendizado efetivo e dá autoridade de decisão para as pessoas que executam o trabalho.

Uma organização que respeita o desenvolvedor de software como um profissional espera dele que defina sua melhor forma de trabalhar através de treinamento apropriado, *coaching*¹⁴ e suporte. Irá esperar do desenvolvedor que melhore continuamente a forma de trabalho como parte de um processo de aprendizado. Finalmente, irá fornecer o equipamento e o tempo necessário. Em uma organização *Lean*, as pessoas que agregam valor são o centro da organização. Trabalhadores da linha de frente possuem a autoridade do processo de *design* e

¹³ CMMI-SW indica: “O processo apóia e permite a satisfação das metas específicas da área de processo, transformando produtos de trabalho de entrada identificáveis em produtos de trabalho de saída identificáveis” (p. 79).

¹⁴ *Coaching* é um processo, com início, meio e fim, definido em comum acordo entre o *coach* (tutor) e o *coachee* (aprendiz) de acordo com a meta desejada pelo cliente, onde o *coach* apóia o cliente na busca de realizar metas de curto, médio e longo prazo, através da identificação e uso das próprias competências desenvolvidas, como também do reconhecimento e superação de suas fragilidades.

responsabilidade no processo decisório; eles são o foco dos recursos, informações e treinamentos.

2.3.3 Fatores relevantes na gestão de equipes de desenvolvimento

Turner e Boehm (2003) identificaram em seu estudo que apesar de metodologias e técnicas de gerenciamento serem de grande valor, os fatores mais críticos de sucesso residem naqueles que influenciam a dimensão de pessoas. Através da comparação entre equipes de desenvolvimento ágeis e equipes direcionadas a processos determinísticos os autores identificaram cinco áreas que acreditam que as empresas podem atingir grande progresso: *staffing*, cultura, valores, comunicação e gestão expectativas.

A tabela a seguir apresenta as principais diferenças entre as duas abordagens no que diz respeito as cinco áreas identificadas por Turner e Boehm (2003).

	Métodos ágeis	Abordagem determinística ou método <i>plan-driven</i> ¹⁵
<i>Staffing</i>	Foco em ter representante do cliente com dedicação exclusiva; Tamanho da equipe: dez pessoas ou menos.	Foco em gerenciar interação com o cliente baseado em especificações (exemplo, atas de reuniões) e planos contratuais;

¹⁵ A definição de processo determinístico apresentada anteriormente é definida por Turner e Boehm (2003) como método *plan-driven*

		Tamanho da equipe: sem limites de tamanho
Cultura	Ambiente confortável e pessoas responsáveis em determinar melhor forma de trabalho; relação de confiança.	Pessoas se sentem confortáveis quando há procedimentos e políticas claras sobre a definição de suas responsabilidades perante a organização.
Proposição de valor	Foco na entrega de valor; priorização de requerimentos e respostas a mudanças de acordo com proposição de valor dos <i>stakeholders</i> .	Foco na produtividade; prioridade em execução e aderências aos processos estabelecidos.
Comunicação	Comunicação através do conhecimento tácito ¹⁶ e interpessoal; cultiva o entendimento de cada um e a vontade de compartilhá-lo;	Comunicação de fluxo unidirecional. Exemplos: descrição de processos e reporte de progresso; Descrição pelos mais experientes de procedimentos, <i>guidelines</i> e

¹⁶ Conhecimento tácito é aquele que o indivíduo adquiriu ao longo da vida, que está na cabeça das pessoas. Geralmente é difícil de ser formalizado ou explicado a outra pessoa, pois é subjetivo e inerente as habilidades de uma pessoa, como "*know-how*". (Wikipedia)

	<p>Conhecimento geralmente é coletado através de revisões (retrospectiva – ver seção 2.2.6);</p> <p>Interação pessoal e colaborativa. Exemplos: reuniões de pé (<i>stand-up meetings</i>), programação em par e jogo de planejamento (<i>planning game</i>¹⁷);</p> <p>Documentação mínima necessária;</p> <p>Rotatividade pode ser um grande risco.</p>	<p>documentação abrangente. Geralmente prove exemplos de utilização, porém para situações simples;</p> <p>Risco em gerar falsa segurança de conhecimento aos menos experientes por seguir procedimentos estabelecidos como boas práticas; Inflexibilidade pode gerar demanda de documentação desnecessária quando a situação não requer.</p>
--	--	--

Tabela 6 - Comparação de métodos ágeis e abordagem determinística na dimensão de pessoas. Adaptado de Turner e Boehm (2003).

É importante salientar a questão da interação comunicativa onde Turner e Boehm (2003) estabelecem a existência de duas categorias primárias de participantes no ambiente de desenvolvimento de software que são os clientes e os desenvolvedores. Para os autores, a engenharia de software ainda lida com o legado da separação de assuntos que afirma que a

¹⁷ *Planning Game* é uma técnica de estimativa baseada no consenso de toda a equipe, onde é utilizado um conjunto de cartas com valores específicos que podem representar pontos relativos ou até mesmo horas (esses valores podem seguir a seqüência de fibonacci) e é praticado como se fosse um jogo. Uma pessoa apresenta a tarefa ou estória para o time, e, após uma **breve** discussão, cada um escolhe uma carta e coloca virada para baixo sobre uma mesa. Quando todas as cartas estiverem lançadas, elas são viradas e caso não haja consenso nos valores escolhidos, as diferenças são discutidas de forma **breve**, e uma nova rodada acontece até que haja a convergência.

tradução de requisitos em software é complexa e que, portanto, deve ser isolada do contexto das pessoas e dos clientes. As seguintes sentenças são apresentadas para ilustrar a situação:

- a noção de *usuário* não pode ser precisamente definida, portanto não é tratada pela ciência da computação ou pela engenharia de software (Dijkstra, 1979 apud Turner e Boehm, 2003);
- análise e priorização dos requisitos de sistema não é de responsabilidade do grupo de engenharia de software, mas sim é pré-requisito para seu trabalho (Paulk, 1994 apud Turner e Boehm, 2003);
- engenharia de software não é gerenciamento de projetos (Tucker, 2002 apud Turner e Boehm, 2003).

Esta separação de assuntos conforme os autores é completamente prejudicial. O processo de desenvolvimento necessita da proximidade dos clientes. Uma das grandes diferenças do desenvolvimento ágil é que ele enfatiza fortemente que a representação do cliente deve estar diretamente ligada ao projeto. O grande risco, neste caso, é a demanda em manter um representante caro com dedicação integral. Dessa forma, Turner e Boehm (2003) identificaram que o sucesso do projeto depende em ter representantes do cliente que sejam colaborativos, representativos, autorizados, comprometidos e conhecedores do domínio de negócio.

Na dimensão dos desenvolvedores os fatores pessoais influentes nas metodologias ágeis incluem benevolência, habilidade, talento e comunicação (Highsmith e Cockburn, 2001 apud Turner e Boehm, 2003). Métodos ágeis tendem a requerer diversidade de habilidades e entendimentos, o que tende a gerar bons resultados com o trabalho colaborativo em equipes. Por outro lado, métodos baseados em processos definidos funcionam bem com pessoas

qualificadas, mas tendem a organizar as tarefas de desenvolvimento de forma a envolver menos recursos para diminuir o risco.

2.3.4 Equipes *Lean*

Nas seções abaixo são apresentados itens relevantes no que diz respeito a montagem e gerenciamento de equipes no contexto *Lean*.

2.3.4.1 Fatores motivacionais

Fatores motivacionais, para Herzberg, são aqueles que se referem ao conteúdo do cargo, às tarefas e aos deveres relacionados com o cargo em si. O termo motivação envolve sentimentos de realização, de crescimento e de reconhecimento profissional, manifestados por meio do exercício das tarefas e atividades que oferecem um suficiente desafio e significado para o trabalhador. Tais sentimentos são os que impulsionam a motivação intrínseca.

Para Poppendieck (2003), empresas devem promover a motivação intrínseca entre seus funcionários, que é especialmente poderosa quando o time se compromete junto para satisfazer um propósito em comum. Abaixo alguma das ações que o autor indica como formas de incentivar a motivação intrínseca:

- estabelecer propósitos factíveis;
- incentivar o time a interagir com os clientes; é a melhor forma para o time entender o propósito do que ele produz;

- deixar o time estabelecer seu próprio comprometimento; ninguém outro deve presumir quanto trabalho deve ser feito em determinado tempo;
- papel do gerente deve servir de interface para o time; líder deve aconselhar ao invés de dizer como deve ser feito. Ele pode não satisfazer todas as requisições mas o importante é manter a motivação zelando pela representatividade do time.

Como visto, a motivação intrínseca é motivada pelo senso de um propósito. Porém, outros fatores devem ser levados em consideração. Pesquisas têm demonstrado que sentimentos de segurança, competência e desenvolvimento são necessários para motivar a equipe¹⁸.

Na tabela abaixo são apresentados os fatores que compõem a motivação intrínseca segundo Thomas (2000) apud Poppendieck (2003).

<p>Escolha</p> <ul style="list-style-type: none"> • Delegar autoridade • Confiar nos funcionários • Segurança (sem punição para erros ocasionais) • Propósito claro • Informação 	<p>Competência</p> <ul style="list-style-type: none"> • Conhecimento • <i>Feedback</i> positivo • Reconhecimento das habilidades • Desafio • Sem padrões de comparação
<p>Significado</p>	<p>Progresso</p>

¹⁸ Thomas apud Poppendieck (2003). *Intrinsic Motivation at Work*. List the building blocks of intrinsic motivation: choice (self determination), meaningfulness (purpose), competence and progress (p. 49)

<ul style="list-style-type: none"> • Clima não cínico • Identificação passional • Tarefas de propósito relevantes 	<ul style="list-style-type: none"> • Clima colaborativo • <i>Milestones</i> • Celebrações • Acesso aos clientes • Avaliação do desenvolvimento profissional
--	--

Tabela 7 *Building Blocks* da motivação intrínseca. Adaptado de Thomas (2000) apud Poppendieck (2003).

Levando em consideração os fatores mencionados acima o que se pode inferir é que o ambiente necessário para proporcionar motivação é o de recompensar o esforço do grupo sem distinção de individualidades. Criar um ambiente de ganhadores e perdedores tende a ser um fator de grande risco para a motivação. Com relação ao acompanhamento do progresso do time é importante que o resultado do trabalho seja posto a prova dos clientes o quanto antes. O que se espera é que o cliente reaja bem vendo que o resultado do projeto possa ser usado, mesmo que em versão *demo*, o que vai alimentar o sentimento de significado reanimando conseqüentemente o time.

2.3.4.2 O papel dos líderes de desenvolvimento

Em desenvolvimento de sistemas geralmente os líderes técnicos recebem o papel de engenheiro de sistemas, programador chefe e arquiteto. Este papel é de extrema relevância. Em alguns casos, os líderes de desenvolvimento não são identificados no início de cada projeto. Conforme estudo de Curtis et al. (1988) apud Poppendieck (2003), a liderança é percebida pela

demonstração de conhecimento e não porque foi formalmente designado como líder. Líderes de conhecimento possivelmente serão respeitados e servirão como parâmetro para o desenvolvimento da equipe. Além disso, o autor salienta que apesar de possuírem extensa experiência de domínio tecnológico, também devem entender das questões de negócio. Devem entender das restrições, interações, requisitos ainda não descritos e potenciais mudanças. Ter a capacidade de olhar par o sistema num nível de abstração alto e também descer ao nível de detalhe e complexidade que ambos desenvolvedores e clientes compreendem também é uma habilidade requerida.

Nas empresas de software, muitas vezes a pessoa que desempenha o papel mencionado acima é designada para trabalhar como aconselhador na definição da arquitetura estabelecida e de verificador da implantação das melhores práticas. Poppendieck (2003) salienta porém, que no ambiente *lean*, o líder de desenvolvimento deve fazer parte do time e ter envolvimento direto com o detalhe do trabalho. Eles serão responsáveis por fornecer a liderança necessária para que o time tome as decisões, faça rápido progresso e desenvolva software de qualidade.

2.3.4.3 O papel do gerente de projetos

Conforme exposto nas seções anteriores, o desenvolvimento de software em um ambiente *lean* é baseado em interações curtas onde os próprios membros da equipe fazem seus comprometerimentos e monitoram o próprio progresso para atingir o objetivo. Através da estruturação do trabalho por um sistema puxado pela demanda do cliente é possível sinalizar aos desenvolvedores o que fazer. Dessa forma, num ambiente *lean* ideal, um gerente de projetos não aloca os recursos as tarefas e não monitora seu andamento.

Na lista a baixo são apresentados algumas das principais responsabilidades do líder de projetos identificadas por Poppendieck (2003):

- identificar o desperdício e esboçar o mapa da cadeia de valor do processo de desenvolvimento atual;
- identificar os gargalos no processo;
- coordenar o planejamento das iterações;
- fornecer os recursos necessários para que o time atinja seu objetivo;

2.3.4.4 A proposta do Time Scrum

Times Scrum são projetados para otimizar flexibilidade e produtividade. Para esse fim, eles são auto-organizáveis, interdisciplinares e trabalham em iterações. O Time Scrum é composto pelo Scrum Master, pelo Product Owner e pelo Time. Para defini-lo, Schwaber e Sutherland (2009) apresentam a seguinte estória:

Uma galinha e um porco estão juntos quando a galinha diz: “Vamos abrir um restaurante!”. O porco reflete e então diz: “Como seria o nome desse restaurante?”. A galinha diz: “Presunto com Ovos!”. O porco responde: “Não obrigado, eu estaria comprometido, mas você estaria apenas envolvida!”

Os membros do Time Scrum são chamados “porcos” enquanto qualquer outra pessoa é chamada de “galinha”. “Galinhas” não podem dizer aos “porcos” como eles devem fazer seu trabalho.

O Time deve ser interdisciplinar, onde membros devem possuir todo o conhecimento necessário para criar um incremento no trabalho. Os membros freqüentemente possuem conhecimentos específicos, como programação, controle de qualidade, análise de negócios, arquitetura, projeto de interface de usuário e projeto de banco de dados. Todos devem contribuir mesmo que isso exija aprender novas habilidades. Não devem existir títulos dentro do Time nem sub-times dedicados a áreas particulares como testes ou análise de negócio. Além disso, o

Time deve ser auto-organizável e ninguém deve dizer como transformar um item do *Backlog* em incremento de funcionalidade. Cada membro aplica sua especialidade a todos os problemas. A sinergia resultante melhora a eficiência e eficácia geral como um todo.

2.4 People-CMM

Conforme já visto anteriormente a dimensão de pessoas possui peso significativo para que uma organização atinja sucesso. Uma forma de gerenciar esta dimensão é através da implementação do P-CMM. O objetivo principal do P-CMM é melhorar a capacitação das pessoas através de um conjunto de práticas de gestão e de desenvolvimento de sua força de trabalho, visando que essa atue como um centro independente de inteligência sem, contudo, desviar-se da estratégia e dos objetivos organizacionais (Curtis et al., 2001). Embora o P-CMM tenha sido originalmente aplicado para empresas que necessitam intensidade de conhecimentos suas práticas podem ser aplicadas para a maioria das organizações fazendo os ajustes apropriados para cada necessidade.

O modelo proposto apresenta uma estrutura em cinco níveis de maturidade onde a organização transforma sua cultura a cada passagem de nível, propiciando o estabelecimento e a evolução das práticas de atração, aprimoramento, motivação e retenção de pessoas.

Conforme Curtis et al. (2001), o P-CMM - como variante do CMM – fornece um guia de alto nível para o desenvolvimento do processo organizacional, não entrando em detalhes de como suas práticas devem ser implementadas. A organização que o adota deve

atentar para fatores como sua dimensão, região geográfica estabelecida, objetivos de negócio, ao ambiente em que está inserida e sua cultura.

Cada nível de maturidade é composto por várias *Áreas de Processos* (PA's). As PA's são agrupadas dentro de diferentes perspectivas e cada uma delas contém um conjunto de metas. Para satisfazer determinado nível de maturidade é necessário que a empresa alcance todos os objetivos de todas as PA's que compõe determinado nível.

O presente projeto de pesquisa irá pesquisar somente as PA's que compõem a perspectiva Construindo Grupos de Trabalho e Cultura (ver passo da metodologia de pesquisa, seção 3.1.2). As PA's são apresentadas no quadro abaixo.

<i>Níveis de Maturidade</i>	<i>Perspectivas</i>			
	Desenvolvendo a Capacidade Individual	Construindo Grupos de Trabalho e Cultura	Motivando e Gerenciando o Desempenho	Ajustando a Força de Trabalho
5 – Otimizado	Desenvolvimento Contínuo da Capacidade		Alinhamento do Desempenho Organizacional	Inovação Contínua da Força de trabalho
4 – Previsível	Ativo baseado em Competência <i>Mentoring</i>	Integração de Competência Grupos de Trabalho <i>Empowered</i>	Gestão Quantitativa do Desempenho	Gestão da Capacidade Organizacional
3 – Definido	Desenvolvimento de Competência Análise de Competência	Desenvolvimento de Grupos de Trabalho Cultura Participativa	Desenvolvimento de Carreiras Práticas Baseadas em Competências	Planejamento da Força de Trabalho
2 – Gerenciado	Treinamento e Desenvolvimento	Comunicação e Coordenação	Ambiente de Trabalho Gestão do Desempenho Compensação	Contratação
1 - Inicial	Não há			

Tabela 8 - Áreas de Processo por Nível de Maturidade e Perspectiva do P-CMM. Adaptado de Curtis et al, 2009.

A seguir, uma breve descrição de cada nível de maturidade.

Nível 1 – Inicial: Caracteriza-se por desempenhar suas práticas de gestão de pessoas de maneira incongruente ou ritualística e apresentam, geralmente, gestores despreparados para desempenharem atividades relacionadas às pessoas. Organizações neste nível enfrentam dificuldades para reter talentos.

Nível 2 – Gerenciado: A organização neste nível apresenta um conjunto básico de boas práticas de gestão de pessoas disciplinadamente. Desenvolve a capacidade de gerir as habilidades e o desempenho de cada nível da unidade organizacional assegurando, dessa maneira, que as pessoas sejam capazes de desempenhar os trabalhos acordados. Preparação de gestores de pessoas é uma prioridade para empresas neste nível.

Nível 3 – Definido: A organização executa práticas básicas de gestão de pessoas porém ainda apresentam inconsistências quanto tais práticas são executadas em nível das unidades e possui pouca sinergia em nível de toda a organização. Através disso, trata a força de trabalho como ativo estratégico capaz de sustentar a busca dos objetivos de negócio.

Nível 4 – Previsível: Possui infra-estrutura estabelecida para desenvolver a força de trabalho. Tal estrutura é sustentada por atividades de *mentoring*. É possível prever a demanda de força de trabalho pois gerencia o processo de competências alinhado com as necessidades de negócio.

Nível 5 – Otimizado: Toda a organização esta focada em melhoria continua. As melhorias são úteis para indivíduos e grupos de trabalho, para o processo de competências e práticas de trabalho. Os resultados de atividades administrativas definidas no nível 4 guiam a implantação de melhorias no nível 5. Mudança de gerencia é tratada como processo ordinário que pode ocorrer com regularidade.

3 METODOLGIA DE PESQUISA

Para atingir o objetivo proposto pelo presente projeto de pesquisa, foi realizada uma pesquisa qualitativa. De acordo com Godoy (1995), a pesquisa qualitativa envolve a obtenção de dados descritivos sobre situações, procurando compreender os fenômenos segundo a perspectiva dos sujeitos, ou seja, dos participantes da situação em estudo. Contrapondo esta definição com os objetivos a serem alcançados notamos que esta se trata da forma de pesquisa que melhor se adéqua ao projeto de pesquisa. O estudo possui caráter exploratório realizado através de pesquisa bibliográfica. As fontes de pesquisa foram livros sobre o tema, material publicado na internet e de domínio público e publicações disponibilizadas pelas bases de dados da instituição.

Com base no tipo de investigação apresentado acima foi estabelecido o método de estudo apresentado no esquema da figura abaixo.

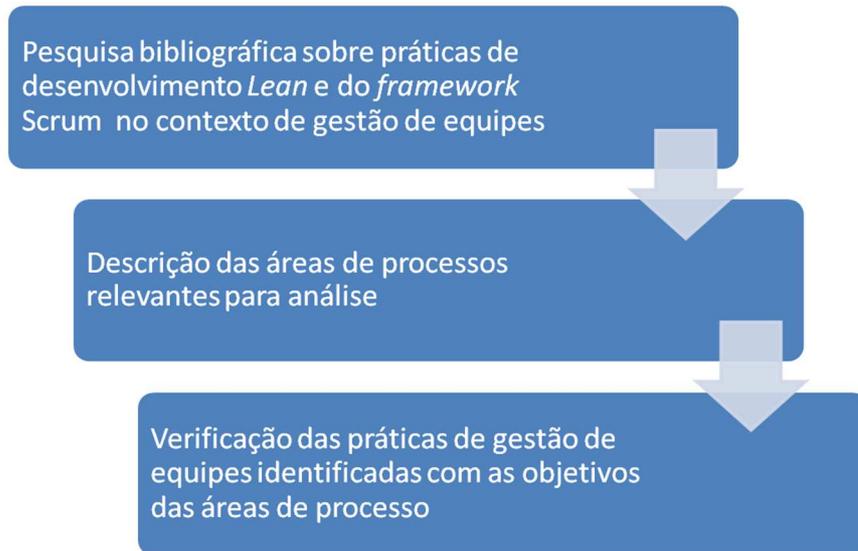


Figura 4 - Passos do método de pesquisa

3.1 ETAPAS DO MÉTODO

Com base no esquema apresentado acima, as seções seguintes descrevem cada uma das etapas realizadas.

3.1.1 Pesquisa bibliográfica sobre práticas de desenvolvimento *lean* e do *framework* Scrum no contexto de gestão de equipes

A primeira etapa pode ser considerada a base para todo trabalho desenvolvido. Ela foi responsável por dar suporte a todas as etapas subsequentes. A bibliografia consultada contemplou livros, periódicos relacionados ao tema abordado, artigos publicados na internet e de domínio público e artigos publicados e disponibilizados nas bases de dados da instituição. Boa parte das fontes de consulta está no idioma inglês e a tradução para o português é de inteira

responsabilidade do autor deste projeto. Alguns termos permaneceram em inglês pois não foram encontradas traduções que mantivessem o significado transmitido pelo original.

A pesquisa bibliográfica buscou identificar o surgimento dos princípios *Lean* nascidos na indústria automobilística japonesa e como eles evoluíram para serem aplicados ao ambiente de desenvolvimento de software. Faz-se uma análise dos métodos de engenharia de software tradicionais baseados em procedimentos que evoluíram da indústria para as empresas de software. É mostrado também como o cenário de negócio atual e peculiaridades dos projetos podem estimular uma abordagem de desenvolvimento empírica. Além disso, mostra os fatores relevantes no que diz respeito a gestão de equipes no contexto *lean*. Por fim, é apresentado um resumo sucinto do *People Capability Maturity Model* (P-CMM).

3.1.2 Descrição das áreas de processos do P-CMM relevantes para análise

As áreas de processos do P-CMM são classificadas em cinco diferentes perspectivas que se enquadram nos diferentes níveis de maturidade. A pesquisa restringiu-se a analisar somente a perspectiva Construindo Grupos de Trabalho e Cultura, pois o objetivo do trabalho é analisar o aspecto de gestão de equipes. Conforme Cockburn (2001) o desenvolvimento ágil é caracterizado por intensa colaboração auto-organização. Salienta ainda que time ágil requer membros tenham foco em comum, respeito e confiança mútuos; um colaborativo e rápido processo de decisão; e habilidade de lidar com ambigüidades. A auto-organização confere aos grupos adaptação necessárias para satisfazer os desafios propostos. Dessa forma serão analisadas as áreas de processos que colaboram para esse fim.

3.1.3 Comparação das práticas de gestão de equipes identificadas com os objetivos das áreas de processo

O estudo bibliográfico permitiu que fossem feitas análises descritivas sobre a relação dos objetivos das áreas de processos do P-CMM com o intuito de práticas de desenvolvimento de software *Lean* focando principalmente nas práticas estabelecidas pelo *framework* Scrum. Pretende-se assim justificar o uso do P-CMM para auxiliar na gestão de equipes no contexto *lean*.

4. ANÁLISE E APRESENTAÇÃO DOS RESULTADOS

Este capítulo destina-se a apresentar e analisar os resultados obtidos no estudo, conforme ultima etapa descrita na metodologia de pesquisa.

4.1 DESCRIÇÃO DAS ÁREAS DE PROCESSOS

Conforme já mencionado anteriormente, o P-CMM descreve um sistema de práticas que auxiliam organizações a desenvolverem uma visão de como integrar um sistema de práticas para alcançar os objetivos traçados para a força de trabalho e provê um *framework* que guia a implementação dessas práticas.

Cada nível de maturidade, com exceção do nível inicial, consiste de três a sete áreas de processos (PA's). Cada PA identifica um grupo de práticas relacionadas que, quando executadas coletivamente, podem alcançar objetivos considerados importantes no desenvolvimento da capacidade das pessoas. As PA's também organizam uma série de práticas inter-relacionadas dentro de uma determinada perspectiva. A perspectiva abrange os diferentes

níveis e cria um sistema de processos relacionados que pode ser sistematicamente implementados visando transformar e melhorar a capacidade gerenciar a força de trabalho. Dessa forma, apesar de PA's residirem apenas em um nível de maturidade, são relacionadas entre si nos diferentes níveis através de sua perspectiva em comum.

O relacionamento entre as PA's estabelece que práticas de determinado nível são transformadas e uma ou mais práticas do nível superior. Por exemplo, práticas da PA Treinamento e Desenvolvimento que foram estabelecidas no nível de maturidade Gerenciado são transformados em praticas da PA Desenvolvimento de Competências do nível Definido. A estrutura conceitual do P-CMM com a relação das áreas de processos em seus diferentes níveis pode ser vista na tabela 6.

A seguir é apresentada uma descrição breve e os objetivos das PA's que compõem as perspectivas *Construindo Grupos de Trabalho e Cultura* e *Desenvolvendo a Capacidade Individual*.

4.1.1 Construindo grupos de trabalho e cultura

Esforços para melhorar coordenação e interação entre pessoas começam no nível Gerenciado através de investimentos nas habilidades de comunicação interpessoal. O foco desse nível é a coordenação entre indivíduos e unidades estabelecendo localmente a capacidade de gerenciar o trabalho comprometido.

Nas práticas do nível Definido, a organização estabelece capacidade de gerenciar dependências resultantes da coordenação das habilidades individuais. Busca-se reduzir o esforço de coordenação através de definição de processos específicos para cada área de

competência. Processos baseados em competências fornecem as bases para desenvolver grupos de trabalho e ajuda a organização a desenvolver cultura participativa.

No nível Previsível, a organização se beneficia dos processos baseados em competências. Busca-se integrá-los e relacioná-los a diferentes áreas de conhecimento para criar um ambiente multidisciplinar que aumente a eficiência na gerencia de dependências entre grupos de trabalhos. A organização investe também na autonomia dos grupos.

No nível Otimizado, grupos de trabalho busca melhoram continuamente a integração dos esforços individuais de cada membro do grupo. Lições aprendidas durante o desenvolvimento dos grupos são revisadas para determinar se serão uteis para outros grupos de processos e áreas de competência.

4.1.1.1 Processo do nível Gerenciado

A seguir consta a descrição das áreas de processos das perspectivas relevantes no nível Gerenciado.

4.1.1.1.1 Comunicação e coordenação

Tem como propósito estabelecer comunicação apropriada através da organização e garante que a força de trabalho possui as habilidades de compartilhar informação e coordenar eficientemente as atividades.

Esse processo estabelece uma cultura de compartilhamento aberta entre os diferentes níveis organizacionais, aumenta o fluxo informacional e é um atributo crítico para

que os indivíduos sintam-se confiantes em levantar questionamentos para a gerência sem receio de retribuições. Estabelecer comunicação efetiva começa pela divulgação dos valores, políticas, práticas e outras informações relevantes. Adicionalmente a esta comunicação *top-down* (de cima para baixo) comunicação *bottom-up* (de baixo para cima) é estimulada para buscar opinião dos indivíduos sobre as condições de trabalho. Procedimentos formais são estabelecidos para eliminar o medo de repressões como forma de respeito aos indivíduos.

Objetivos

- Informação é compartilhada entre toda a organização;
- Indivíduos e grupos são incentivados a questionar e reportar a questão ao nível gerencial;
- Indivíduos e grupos coordenam suas atividades para realizarem as metas propostas;
- Práticas de comunicação e coordenação são institucionalizadas para garantir que sejam realizadas como processos gerenciáveis.

4.1.1.2 Processos do nível Definido

A seguir consta a descrição das áreas de processos das perspectivas relevantes no nível Definido.

4.1.1.2.1 Cultura participativa

O propósito é habilitar a capacidade integral da força de trabalho em tomar decisões que afetam o desempenho das atividades de negócio.

A comunicação aberta estabelecida com o processo de comunicação e coordenação do nível Gerenciado fundamenta a cultura participativa. Ambiente participativo estimula o fluxo informacional e incorpora o conhecimento de indivíduos ao processo decisório. A cultura inicia provendo indivíduos e grupos com informações sobre seu desempenho e como eles influenciam a organização. A estrutura do processo decisório é analisada e papéis apropriados para tal são definidos. Processos definidos são utilizados para tomar decisão e resolver conflitos e disputas.

Objetivos

- Informações sobre atividades de negócio e resultados são comunicados através da organização;
- Decisões são delegadas a um nível apropriado da organização;
- Indivíduos e grupos participam de um processo estruturado de tomada de decisão;
- Práticas da cultura participativa são institucionalizadas para garantir que sejam realizadas como processos definidos pela organização.

4.1.1.2.2 Desenvolvimento de grupos de trabalho

Tem como propósito organizar e alinhar o trabalho com os processos baseados em competências.

O P-CMM define grupo de trabalho como um conjunto de pessoas que trabalham juntas em tarefas que são altamente interdependentes com objetivos compartilhados. A organização define métodos e procedimentos comuns aos grupos. Os grupos atribuem membros aos papéis definidos dentro atividades de negócios. O grupo reporta para um responsável que analisa o trabalho e estabelece relações entre tarefas. Além disso, o responsável gerencia o desempenho rastreando o status do trabalho desenvolvido. Quando atividades de negócio são concluídas pelo grupo, processos ordenados são executados para preservar o ativo gerado, finalizar atividades e direcionar os membros para outras atividades e ou grupos.

Objetivos

- Grupos são estabelecidos para aperfeiçoarem e realizarem trabalhos interdependentes;
- Grupos de trabalhos adéquam processos definidos e papéis ao planejamento e execução do trabalho;
- Atividades de *staffing* focam em alocar, desenvolver e consolidar as competências da força de trabalho;
- Desempenho do grupo é gerenciado através de objetivos documentados e associados ao trabalho comprometido;
- Práticas de desenvolvimento do grupo são institucionalizadas para garantir que sejam executadas como processos definidos pela organização.

4.1.1.3 Processos do nível Previsível

A seguir consta a descrição das áreas de processos das perspectivas relevantes no nível Previsível.

4.1.1.3.1 Integração de competências

Tem como propósito melhorar a eficiência e agilidade de trabalhos interdependentes integrando habilidades de diferentes áreas de conhecimento.

Um processo baseado em competências integradas é aquele que provê conhecimentos e habilidades necessárias para suportar múltiplos papéis em um processo de negócio integrado. Dessa forma, indivíduos com diferentes competências trabalham juntos em um processo multidisciplinar integrado e único ao invés de trabalharem separadamente apenas em sua área de conhecimento. O foco é direcionado ao gerenciamento do fluxo de atividades no contexto do negócio, produção ou serviço ao invés de gerenciar apenas tarefas dentro de determinada área de competência. Geralmente refere-se a um processo multifuncional, uma vez que integra atividades de áreas de competências antes isoladas dentro funções departamentais e trata o negócio como um único *workflow* de processos de negócio.

Objetivos

- Processos baseados em competências empregados por diferentes áreas de conhecimento são integrados para melhorar a eficiência de trabalho interdependente;
- Processos baseados em competências integrados são usados para realizar trabalho que envolve dependências entre várias áreas de conhecimento;
- Práticas da força de trabalho são designadas para suportar trabalho multidisciplinar;
- Práticas de integração de competências são institucionalizadas para garantir que sejam realizadas como processos definidos pela organização.

4.1.1.3.2 Grupos de trabalho *empowered*

O propósito é investir para que grupos de trabalhos tenham responsabilidade e autoridade em determinar como conduzir suas atividades de negócio de forma mais efetiva.

Um grupo de trabalho *empowered* é aquele que recebe concessão de autonomia considerável e autoridade para gerenciar e realizar seu trabalho. Para potencializar o grupo, membros são preparados para atuarem como entidades independentes. Envolve treinar os membros em habilidades necessárias e gerenciar o grupo como entidade única ao invés de gerenciar os indivíduos.

Objetivos

- Grupos de trabalho *empowered* possuem responsabilidade e autonomia em seus processos de trabalho;
- As práticas e atividades da força de trabalho da organização estimulam e suportam o desenvolvimento e desempenho dos grupos;

- Grupos de trabalhos *empowered* realizam práticas de trabalhos selecionadas internamente;
- Práticas de grupos de trabalhos *empowered* são institucionalizadas para garantir que sejam executadas como processos definidos pela organização.

4.2 COMPARÇÃO DOS PRINCÍPIOS E PRÁTICAS *LEAN* COM OS OBJETIVOS DAS ÁREAS DE PROCESSOS

A presente seção visa buscar relações entre os princípios e práticas *lean* - identificados na literatura e presente neste projeto - com os objetivos das áreas de processos do P-CMM dentro das perspectivas consideradas relevantes. Procura-se verificar a possibilidade de executar uma estratégia de gestão de pessoas baseadas nas práticas do P-CMM em uma organização que adota conceitos *lean* para desenvolvimento de software. A análise é apresentada agrupando as áreas de processos por nível de maturidade.

4.2.1 Verificação do nível Gerenciado

A questão da comunicação no nível Gerenciado foca em difundir e construir as bases de uma cultura de comunicação aberta. Dentre um dos comprometimentos para esse nível está o estabelecimento e manutenção formal das atividades de comunicação e coordenação. É importante que a organização mantenha um ambiente aberto que suporte o fluxo de comunicação em todas as direções. Essas definições do P-CMM vão de encontro com a

definição de Turner e Boehm (2003) sobre a área da comunicação em métodos ágeis de desenvolvimento. Conforme os autores, a comunicação aberta é incentivada pela troca de conhecimento tácito e interpessoal, cultivando o entendimento de cada um e a vontade de compartilhar informações.

Dentre as formas relevantes que a PA identifica como exemplos de disseminar a cultura de comunicação aberta estão incentivos para comunicação sem receio de repressão, ênfase na colaboração e trabalho em equipe, sensibilidade e respeito pela diversidade cultural dentro da força de trabalho, ênfase na qualidade de vida no trabalho (*work-life balance*) entre outros.

Levando em consideração as características citadas, Turner e Boehm (2003) mencionam que a Reunião Diária do *framework* Scrum colabora diretamente para isso. A Reunião Diária, conforme Schwaber e Sutherland (2009), melhora a comunicação, elimina outras reuniões, identifica e remove impedimentos para o desenvolvimento, ressalta e promove a tomada rápida de decisões e melhora o nível de conhecimento de todos acerca do projeto. Outro importante aspecto da Reunião Diária é o fato de interessados no projeto (definição de “galinhas” do time Scrum, ver seção 2.3.4.4) não possuem autorização para falar nem interferir. Dessa forma a equipe, que é responsável por transformar itens do *backlog* em incremento do produto, são incentivados a falar e reportar problemas enfrentados.

Ainda sobre ambiente de comunicação aberto, uma das regras de gerência do XP menciona que caminhos vitais de comunicação podem ser traçados simplesmente removendo as barreiras que dividem as pessoas (Wells, 2009). Colocando computadores numa área central sem donos proporciona que a programação em pares seja encorajada. O autor ainda cita a utilização de mesas de conferências que tornem a discussão espontânea durante o dia e espaços abertos que auxiliem na manutenção das reuniões de pé (*stand up meeting*).

Com relação a questão de coordenação das atividades o *framework* Scrum estabelece formas dentro do controle de processo empírico onde membros do time possam inspecionar o andamento de trabalho comprometido. A Reunião Diária, como mencionada acima, é uma importante ferramenta de coordenação das atividades do time. O uso do gráfico *burndown* é um artefato do Scrum que fornece informações nesse sentido. Adicionalmente, o uso de *kanban* de software (Poppendieck, 2003) também é utilizado. Conforme Hiranabe (2008), *kanban* de software é uma prática para visualizar e compartilhar status de projeto colocando cartões numa parede do ambiente de trabalho. Cada cartão pode ser identificado como uma tarefa e geralmente é uma unidade desmembrada de histórias de usuários (*user stories*¹⁹) que contem descrição e estimativa. O status é indicado pela área em que o cartão se encontra. Cada área possui um significado que descreve passos do desenvolvimento (exemplos: a fazer, em progresso, pronto, testado, aceito, bloqueado, etc). O quadro *kanban* ajuda na visualização das tarefas sendo executadas no momento. Poppendieck (2003) acrescenta que controles visuais e gerência por sinalização são conceitos do TPS que foram adaptados ao desenvolvimento para permitir que todo o time veja o que esta sendo feito, o que ainda precisa ser feito, quais problemas existem e que progresso foi realizado. Acrescenta ainda que lista de problemas, melhoramento em idéias, impactos no negócio, glossário de linguagem ambígua entre outras, são itens candidatos a entrarem num quadro grande e visível.

Na questão de compartilhamento das informações entre toda a organização não foram encontradas nem descritas práticas que auxiliem especificamente neste sentido. Os conceitos *lean*, como apresentado acima, enfatizam na comunicação aberta dentro do time, porém não descreve mecanismos para compartilhar os resultados obtidos com toda a organização. Também não descreve papéis específicos a serem desempenhados por membros

¹⁹ *User story* é um requisito de sistema de software formulado como uma ou mais sentenças do dia-a-dia ou da linguagem de negócio do usuário. *User stories* devem ser escritas por clientes e constituem o principal instrumento de influencia no desenvolvimento de software que o cliente possui.

de cada unidade de trabalho, como mencionado nas práticas da PA comunicação e coordenação. A integração de informações e divulgação entre equipes interdependentes são questões que podem ser endereçadas pelas práticas do P-CMM e adaptadas pela organização ao modelo *lean*.

A visão de Poppendieck (2003) sobre a avaliação do resultado produzido sempre deve se levar em consideração o esforço do grupo, sem distinção de individualidades. No caso do Time Scrum, Schwaber e Sutherland (2009) menciona que as equipes devem ser multidisciplinares onde membros devem possuir todo o conhecimento necessário para criar um incremento no produto. Portanto, ao se constituir um time é necessário assegurar que as habilidades fundamentais requeridas estão de acordo com a capacidade de esforço da equipe. Variações com pessoas mais e menos experientes fornecem oportunidades de desenvolvimento e estabelecem formas de a organização criar um plano de desenvolvimento de carreiras. A PA treinamento e desenvolvimento, da perspectiva de desenvolvimento individual, auxilia na composição dos times mantendo através de suas práticas formas de identificar as habilidades fundamentais - aquelas que podem comprometer a execução de uma atividade – e direcionar necessidade de treinamentos imediatos.

No quadro abaixo é apresentado um resumo as áreas de processos desse nível e comentários sobre a relação com os conceitos de desenvolvimento *lean*.

Área de Processo	Relação
Comunicação e coordenação	Métodos ágeis, de forma geral, fornecem excelentes práticas para satisfazer os objetivos desse processo. Desenvolvimento <i>lean</i> foca fortemente na comunicação aberta e transparente. O processo de controle empírico requer que inspeções sejam

	<p>conduzidas freqüentemente e para isso é necessário manter comunicação assídua. Artefatos do Scrum e outros controles visuais baseados no <i>kanban</i> do TPS servem de ferramentas para coordenar as atividades ao longo da duração de todas as iterações. A multifuncionalidade resultante do Time Scrum pode fornecer bases para a criação de um plano de carreiras na organização, que são questões abordadas por processos da perspectiva individual.</p>
--	---

Tabela 9- Resumo da verificação no nível Gerenciado

4.2.2 Verificação do nível Definido

Dentro da perspectiva de construção de cultura o P-CMM define que a organização deve se comprometer em manter políticas que incentivem a participação. Com isso é importante que a estratégia de negócio da empresa esteja esclarecida, pois o resultado da participação deve ser direcionado a resolução de problemas ligados ao negócio. O *lean* sugere que as pessoas foquem naquelas atividades que agregam maior valor ao cliente (Liker, 2008). Portanto a proposição de valor dos clientes e *stakeholders* devem estar alinhadas para que indivíduos contribuam agregando valor e removendo desperdício. O comprometido em manter um responsável que faça com que indivíduos contribuam de forma participativa pode ser atribuído ao papel do Scrum Master. Membros do time relatam a ele impedimentos no desenvolvimento do incremento de produto. Conseqüentemente, cabe ao Scrum Master a responsabilidade de levar os problemas ao nível gerencial para resolvê-los quando necessário. Apesar disso, o Scrum não estabelece nem requer formalidade como descrito nos objetivos da PA. Uma

alternativa seria transformar impedimentos não resolvidos de imediato em itens de um *backlog* do time, separados do *backlog* do produto. Esses itens seriam gerenciados pelo Scrum Master. Esta poderia ser uma forma de manter histórico e acompanhamento dos itens reportados para o nível gerencial minimizando o estabelecimento de processo burocrático e inflexível. É importante salientar também que no TPS, é responsabilidade de todos parar a produção se algo estiver acontecendo de errado (Harvey, 2004). Isso salienta a importância de disseminar a proatividade entre a força de trabalho para que quando seja detectado algum risco, por qualquer membro, este seja reportado o quanto antes.

O P-CMM também estabelece para a PA de cultura participativa a definição de processo decisório e papéis. Métodos de tomada de decisão fornecem procedimentos específicos que podem ser incluídos no trabalho individual, de grupos e de unidades. É necessário também analisar a estrutura de decisão para que se delegue ao nível organizacional correto. Neste aspecto, Poppendieck (2003) salienta que investir na inteligência dos trabalhadores da linha de frente é uma forma de fomentar o processo decisório. Conforme Bowen e Yougdahl (1998) dar autonomia para os empregados é alavancar a equação de valor do cliente.

No desenvolvimento de grupos de trabalhos é importante salientar que geralmente equipes são agrupadas por uma única área de competência (time de desenvolvimento, time de vendas, finança, auditoria) e indivíduos preenchem papéis estruturados nas habilidades definidas dentro de cada área de competência. No nível Definido, ocasionalmente se requer coordenação de atividades executadas por indivíduos de diferentes áreas de conhecimento. Dessa forma, pessoas de diferentes funções são associadas a papéis dentro de um grupo de trabalho sem necessariamente tornarem-se membros integrantes do grupo. Se levarmos em consideração a idéia do Time Scrum em prática, fica a impressão de que ele não leva em consideração esse ambiente. O P-CMM descreve que empresas nesse nível possuem um

ambiente onde indivíduos usam interfaces definidas entre suas diferentes áreas de competências para gerenciar a dependência mútua. Esse cenário é semelhante ao que Rummler e Brach (1992) descrevem como uma organização de visão “tradicional” onde as estruturas são organizadas na forma em que os autores definiram como “silos” funcionais. Portanto, se existir forte relacionamento hierárquico nessa estrutura, as interfaces de relacionamento entre assuntos interdepartamentais serão inflexíveis forçando que gerentes tenham que resolver problemas, o que acaba elevando o custo e o tempo de execução das tarefas. Por outro lado, o *framework* Scrum permite, através do Time Scrum, que membros componham todas as habilidades críticas para produzir valor. Royce (1970) menciona que as atividades que mais agregam valor definitivamente são análise de requisitos e codificação. Todas as outras atividades básicas que contribuem diretamente para a criação de valor devem estar presentes nas habilidades do grupo. Dentre habilidades se poderiam incluir, além de desenvolvimento: arquitetura de sistemas, análise de requisitos, análise de qualidade (testes), especialistas de interface, documentadores e outros.

Com relação a avaliação de desempenho do trabalho do grupo, Poppendieck (2003) salienta que no ambiente *lean* as pessoas se sentem motivadas quando existe um propósito compartilhado. O Scrum também salienta que não se deve existir distinções específicas para membro do grupo (Schwaber e Sutherland, 2009). Dessa forma a avaliação e compensação devem levar em conta o esforço do grupo sem distinção de individualidades. Métodos de definição de métricas para objetivos como o *Balanced Score Card*²⁰ (Kaplan e Norton, 1997) seriam mais adequados para satisfazer os objetivos da PA na gestão de desempenho do grupo.

²⁰ Ferramenta que procura fornecer a visão de conjunto dos fatores críticos de sucessos. Essa técnica focaliza quatro dimensões - chamadas perspectiva do cliente; perspectiva interna; perspectiva da inovação e aprendizagem organizacional; perspectiva financeira - importantes do desempenho da empresa, cada uma das quais se desdobra em medidas específicas que podem dividir-se em indicadores.

No quadro abaixo é apresentado um resumo das áreas de processos desse nível e comentários sobre a relação com os conceitos de desenvolvimento *lean*.

Área de Processo	Relação
Cultura participativa	<p>Práticas mencionadas no nível Gerenciado fornecem as bases para a organização disseminar a cultura participativa. Nesse nível é necessário que a organização mantenha formas de divulgação da estratégia e esclareça a proposição de valor de clientes externos e internos para que indivíduos possam contribuir e participar na realização dos objetivos.</p> <p>Estabelecimento de procedimentos específicos para toma de decisão não são endereçados pelas práticas <i>lean</i>, porém seus princípios salientam a transferência adequada de responsabilidade e priorizam investimentos nas pessoas de linha de frente para que estejam aptas a tomarem decisões.</p>
Desenvolvimento de grupos de trabalhos	<p>Práticas e regras recomendadas pelo <i>framework</i> Scrum na perspectiva de equipes parecem não considerar o ambiente descrito no nível Definido. Métodos específicos para determinação de indicadores de desempenho poderiam ser utilizados para gerenciar baseado em objetivos.</p>

Tabela 10 - Resumo da verificação no nível Definido

4.2.3 Verificação do nível Previsível

A integração de competência nesse nível preocupa-se em prover a organização de conhecimentos e habilidades necessárias para suportar múltiplos papéis em um processo de negócio integrado. Com isso é necessário que grupos trabalhem juntos, de forma multidisciplinar ao invés de isolados em suas áreas de conhecimento. Conforme Schwaber e Sutherland (2009), o conceito do Time Scrum vai de encontro as necessidades colocadas por esse processo. Time Scrum deve ser interdisciplinar e auto-organizável. Como já mencionado na análise anterior, a equipe deve possuir todas as habilidades críticas e papéis dentro do time devem ser definidos para contemplar as demais tarefas que agregam valor.

Entretanto, o *framework* Scrum não estabelece formas para gerenciar e sincronizar o esforço de vários times quando necessário. Quanto mais interfaces existirem para gerenciar maior será a complexidade do processo. Rummler e Brache (1992) mencionam que empresas que adotam uma visão “funcional”, descrita no nível Definido, exigem a manutenção de diversas interfaces entre as funções causando perdas de tempo e aumento do custo na execução dos processos que se cruzam. Em contra partida, Rummler e Brache (1992) mostram que o conceito de gestão direcionado a processos alinha os esforços individuais de cada área. A idéia principal dessa abordagem é a de que negócios cruciais da empresa mereçam atenção prioritária dos esforços gerenciais com as diferentes funções cumprindo um papel de apoio para a execução dos mesmos.

Aproximando esse conceito ao desenvolvimento e software, o *Feature-Driven Development* (FDD) (Palmer e Felsing, 2002) é uma prática que auxilia no gerenciamento de diferentes times que trabalham no desenvolvimento de um mesmo sistema. O FDD estabelece responsabilidade individual de módulos e classes do sistema. Uma vez que funcionalidades

requerem manutenção de diversas classes, FDD auxilia na criação de times separados por funcionalidade. Práticas descritas no P-CMM para esse nível poderiam completar a lacuna no que diz respeito a coordenação e sincronização do esforço de trabalho, o que significa que é necessário estruturar dependências entre diferentes áreas de competências.

Poppendieck (2003) menciona ainda a abordagem de sincronização por matriz quando há necessidade de interação de times distintos de desenvolvimento que trabalham na criação do mesmo software. Essa abordagem, segundo o autor, é particularmente apropriada quando diferentes times não estão localizados no mesmo espaço podendo estar até mesmo em fuso horário diferente. O desenvolvimento então se inicia com a criação das interfaces para depois se criar os subsistemas. Todos os pontos de intersecção entre os times devem ser apresentados no início. Grupos de trabalhos são designados para cada um desses pontos de intersecção. Esse trabalho permite que componentes de software que se cruzam possam ser demonstrados no final de cada iteração.

Conforme o exposto se percebe que equipes *lean* tendem a visualizar o processo na perspectiva *end-to-end*²¹ acompanhado e entendendo o fluxo da cadeia de valor.

No processo de desenvolvimento de grupos *empowered* se busca conceder autonomia considerável para o grupo tomar decisões e se potencializa as habilidades de membros do grupo com intuito de capacitá-los para tais responsabilidades. *Empowerment* no ambiente *lean* significa, para Staats (2009), criar um ambiente onde as pessoas sintam que elas possuem influencia real nos padrões de qualidade, serviços e negócios dentro de suas áreas de

²¹ A expressão “*end-to-end*” está relacionada com o termo utilizado em ambientes de redes que significa um ponto de comunicação entre os nodos origem e destino, ou seja, a comunicação entre dois ou mais “*end points*”. Na visão empresarial, pode significar a visualização de um processo de negócio a partir do momento em que ele entra no controle de determinado departamento até que seja completado e entregue a outro departamento ou outra organização externa. Numa outra perspectiva, significa uma abordagem de negócios direcionada a processos, ao invés da abordagem funcional tradicional, com o intuito de avaliar os impactos de decisões estratégicas nos processos empresariais críticos para a organização.

responsabilidade. Para o autor, poucos níveis administrativos e cultura baseada em processo são características encontradas em empresas que propiciam tal ambiente. Poppendieck (2003) também salienta que transferir responsabilidades para o time é uma forma de estimular a motivação intrínseca nos funcionários. O poder do grupo é delegado através de estabelecimento de propósitos factíveis, incentivo a interação com clientes, permissão para que o time faça seus próprios comprometimentos e que o papel do gerente seja de representar a equipe perante a organização. Essas características vão de encontro com as regras do Time Scrum que estabelece que apenas pessoas comprometidas em desenvolver incremento de produto são permitidas a determinar como as tarefas identificadas para atingir os objetivos devam ser executadas.

No quadro abaixo é apresentado um resumo das áreas de processos desse nível e comentários sobre a relação com os conceitos de desenvolvimento *lean*.

Integração de competências	Regras do <i>framework</i> Scrum compõem excelentes práticas na satisfação de alguns dos objetivos dessa área de processo. Com o estabelecimento de um time multidisciplinar, munido com as habilidades críticas e capacitação para a tomada de decisão permitem ao grupo adequar papéis e processos definidos pela organização. A questão de sincronização e integração de grupos interdependentes é abordada por diferentes práticas dos métodos ágeis que podem trabalhar em conjunto com as práticas do Scrum.
Grupos de trabalho <i>empowered</i>	<i>Empowerment</i> é um dos princípios de desenvolvimento <i>lean</i> mais importante e confere as pessoas capacidade de manter uma cultura de melhoria contínua. Este

	também é um dos importantes fatores que motivam a equipe a trabalhar por um senso comum.
--	--

Tabela 11 - Resumo da verificação no nível Previsível

5 CONCLUSÃO

O presente projeto buscou através de uma pesquisa qualitativa de caráter exploratório buscar maiores conhecimentos sobre as práticas de desenvolvimento *lean* no âmbito de gerenciamento de equipes. Atualmente, muitas empresas vêm aos poucos adaptando práticas do *framework* Scrum dentro de seus processos sem de fato adotarem a estrutura organizacional necessária para a manutenção dessas práticas. Conforme mencionado no trabalho, a adoção de *lean* vai além da simples adoção das práticas, mas também diz respeito à integração com os processos existentes, sendo que o grande desafio se concentra em fazer o projeto ágil funcionar dentro do ambiente organizacional.

De qualquer maneira, é necessário que a organização estabeleça certo nível de formalidade em mecanismos que propiciem o desenvolvimento da equipe alinhado aos objetivos organizacionais. Através disso, é possível criar uma infra-estrutura que dê o suporte para um desenvolvimento sustentável através de boas práticas de mercado. É com este intuito que o P-CMM ajuda no melhoramento do processo de gestão e no estabelecimento de um programa de desenvolvimento contínuo da força de trabalho.

Através do referencial teórico foi possível mostrar como os conceitos do *lean*, nascidos na indústria japonesa pós-guerra, foram adaptados ao contexto de desenvolvimento de software. Verificou-se que grande parte do insucesso dos projetos reside no fato de que o ambiente de negócio atual requer que empresas mantenham ritmo de inovação constante, o que acaba impactando em mudanças frequentes nos processos que suportam o negócio. Para esse contexto, pesquisadores de práticas *lean* e métodos ágeis sugerem uma abordagem empírica de desenvolvimento ao invés dos métodos tradicionais que tratam a engenharia de software como um processo determinístico.

Foi com base nesses conceitos que surgiu o *framework* Scrum. Através da flexibilidade de seus processos é possível que organizações façam adaptações para adequarem à sua realidade. Tanto o Scrum quanto práticas *lean* focam fortemente na questão de pessoas. Pesquisadores consideram que das três dimensões (requisitos, tecnologia e pessoas) que envolvem a complexidade de projetos de software, a que fornece maior risco é a dimensão das pessoas. Por isso, buscou-se verificar quais são os fatores determinantes dentro dessa perspectiva e como elas se relacionam dentro do trabalho em equipes. Concluiu-se então dessa forma que a adoção de Times Scrum fornece a base para processos *lean* satisfazendo de certa forma alguns dos objetivos dos processos do P-CMM para a perspectiva de desenvolvimento de grupos de trabalho.

A análise da relação entre os conceitos de *lean* com os processos do P-CMM também mostrou que a cultura de comunicação é aberta é muito importante para se atingir sucesso. Ela permite que a empresa e equipes ataquem um dos principais problemas (que é a comunicação) e, conforme o P-CMM, fornece as bases para o desenvolvimento das equipes. Dessa forma, verificou-se que o ambiente *lean* necessário está fortemente relacionado com práticas do nível Gerenciado. Concluiu-se portanto que, empresas que adotem formalmente processo *lean* aliado com o *framework* Scrum se habilitam consideravelmente a obtenção do

nível Gerenciado descrito no P-CMM, dentro da perspectiva de gerenciamento de equipes. Para o nível Definido, verificou-se sua caracterização não descreve o ambiente que se busca com Times Scrum, portanto, uma adoção de *lean* não iria contribuir muito para satisfazer objetivos de processos para esse nível. Já no nível Previsível, verificou-se que o ambiente que se busca através de uma implementação *lean* é o descrito por esse nível de maturidade. Mas para isso é necessário que se tenha disciplina e um grande trabalho para prepara a infra-estrutura necessária.

Conforme o exposto é possível verificar que através da execução dos passos metodológicos estabelecidos para o projeto de pesquisa cobriram todos os objetivos especificados para o mesmo.

REFERÊNCIAS

Boehm, B. W. “*Industrial software metrics top 10 list*”. IEEE Software, 4(5):84–85, Set 1987.

Bowen, D. E.; Youngdahl, W. E. “*Lean Service: In Defense of a Production-Line Approach*”. International Journal of Service Industry Management; v. 9, Issue 3, 1998
Research paper.

CMMI para Desenvolvimento – Versão 1.2. Agosto de 2006.

Cockburn, A., “*Agile Software Development*”. Addison-Wesley Professional. October, 2001. P 84-88.

Curtis, B.; Hefley, W. E.; Miller, S. A. “*Experiences applying the people capability maturity model.*” Cross Talk – The journal of defense software engineering, v. 16, n. 4, p. 24-27, abr. 2003.

Curtis, B.; Hefley, W. E.; Miller, S. A. “*People Capability Maturity Model*”. Pittsburg: Software Engineering Institute, 2001. 735 p. Disponível em: <<http://www.sei.cmu.edu/pub/documents/01.reports/pdf/01mm001.pdf>>.

Curtis, B.; Herb, K.; Iscoe, N.. “*A field Study of the Software Design Process for Large Systems*”. Communications of the ACM 31(11): November 1988, 1268-1287.

Dijkstra, E. “*Panel Discussion. Fourth International Conference on Software Engineering*”, 1979.

Fortes, C. S., “*Aplicabilidade de Lean Service na melhoria de serviços de tecnologia da informação (TI)*”. Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal do Rio Grande do Sul. Porto Alegre, 2010.

Highsmith, J., Cockburn, A. R. “*Agile Software development: the business of innovation*” Humans and Technology, Sept. 2001. Disponível em: <http://alistair.cockburn.us/Agile+software+development%3a+the+business+of+innovation>.

Acessado em 10 de junho de 2010.

Harvey, D. “*Lean, Agile*”. Paper for Workshop “The Software Value Stream” OT2004. Revision: 43, 2004.

Highsmith, J., Cockburn, A. R. “*Agile Software development: the people factor*” Humans and Technology, Sept. 2001. Disponível em: <http://alistair.cockburn.us/Agile+software+development%3a+the+people+factor>. Acessado em 10 de junho de 2010.

Hiranabe, K. “*Kanban Applied to Software Development: from Agile to Lean*”. Janeiro, 2008. Disponível em: <http://www.infoq.com/articles/hiranabe-lean-agile-kanban>. Acessado em 15 de Agosto de 2010.

Mah, M. “*IT Organization, Benchmark Thyself, part 2*”. IT Metrics Strategies. April, 2000. Disponível em http://www.qsma.com/pdfs/ITMSVol_VI4.pdf. Acessado em 10 de junho de 2010.

Lindvall, M.; Muthig, D.; Dagnino, A.; Wallin, C.; Stupperich, M.; Kiefer, D.; May, J. e Kahkonen, T. 2004. “*Agile Software Development in Large Organizations*”. Computer 37, 12 (Dec. 2004), 26-34. DOI= <http://dx.doi.org/10.1109/MC.2004.231>.

Ohno, Taiichi. “*Toyota Production System: Beyond Large-Scale Production*”. Portland, OR: Productivity Press, 1988.

Paulk, M., et al. “*The Capability Maturity Model for Software: Guidelines for Improving the Software Process*”. Reading, MA: Addison-Wesley, 1994.

Poppendieck, M. “*Principles of Lean Thinking*”. Novembro, 2002. Disponível em <http://www.poppendieck.com/publications.htm>. Acessado em: 2 de Junho de 2010.

Talby, D.; Keren, A.; Hazzan, O.; Dubinsky, Y.; , "Agile software testing in a large-scale project" *Software, IEEE*, vol.23, no.4, pp.30-37, July-Aug. 2006

doi:

10.1109/MS.2006.93

URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1657936&isnumber=34708>.

Rummler, G.A; Brache, A. P. “*Melhores desempenhos das empresas: Ferramentas para melhoria da qualidade e da competitividade*”. ed 1, São Paulo: Makron, 1992.

Schwaber, K. “*Agile Project Management with Scrum*”. Microsoft Press, 2004.

Schwaber, K., Sutherland, J. “*Scrum Guide*”. February, 2009. Disponível em <http://www.scrum.org/scrumguideenglish/>.

Staats, B. “*Employee Empowerment is Key to Success With Lean*”. Agust, 2009. Diponível em:

<http://ezinearticles.com/?Employee-Empowerment-is-Key-to-Success-With-Lean&id=2725391>. Acessado em 10 de Agosto de 2010.

Palmer, S. R., Felsing, J. M. “*A Practical Guide to Feature Driven Development*”. Prentice Hall, 2002. 12.

Tucker, A. “On the Balance Between Theory and Practice.” IEEE Software Sept.-Oct. 2002.

Wells, D. “*Extreme Programming: A gentle introduction*”, 2009. Disponível em:

<http://www.extremeprogramming.org/>. Acessado em 12 de Agosto 2010.

Royce, W. “*Managing the development of large software systems*”. In Proceedings of IEEE Wescon, pages 1–9, Aug 1970