

CENTRO PAULA SOUZA

GOVERNO DO ESTADO DE
SÃO PAULO

**Faculdade de Tecnologia de Americana
Curso de Bacharelado em Análise de Sistemas e Tecnologia da
Informação**

SCRUMBAN PARA *SERIOUS GAMES* NA ÁREA DA SAÚDE

João Henrique de Oliveira Zanini

Americana, SP

2012

**Faculdade de Tecnologia de Americana
Curso de Bacharelado em Análise de Sistemas e Tecnologia da
Informação**

SCRUMBAN PARA *SERIOUS GAMES* NA ÁREA DA SAÚDE

João Henrique de Oliveira Zanini

joaozanini@outlook.com

Projeto desenvolvido em cumprimento curricular da disciplina Projeto de Graduação do Curso de Bacharelado em Análise de Sistemas e Tecnologia da Informação da FATEC – Americana, sob orientação da Prof^o. Me. Kléber de Oliveira Andrade.

Área: Métodos Ágeis, Serious Games, Saúde.

Americana, SP

2012

João Henrique de Oliveira Zanini 0912817

SCRUMBAN PARA SERIOUS GAMES NA ÁREA DA SAÚDE

Monografia aprovada como requisito parcial para obtenção do título de Bacharel no curso de Análise de Sistemas e Tecnologia da Informação da Faculdade de Tecnologia de Americana.

Banca Examinadora

Orientador: _____
Kléber Andrade de Oliveira, Mestre, FATEC - Americana

Professor da Disciplina: _____
Clerivaldo José Roccia, Mestre, FATEC - Americana

Professor Convidado: _____
Cleberon Eugenio Forte, Mestre, FATEC - Americana

Americana 13/12/2012

Dedico este trabalho aos meus pais pelo suporte e amor incondicional, assim como toda minha família, amizades e todas as corajosas almas que optaram por fazer da complexa arte de criar jogos parte de suas vidas.

AGRADECIMENTOS

Primeiramente a Luciana Fávero pelo enorme apoio prestado no desenvolvimento deste trabalho e pela amizade infinita.

A Larissa Ariadne de Sousa Viana pela ajuda prestada neste trabalho, pelo companheirismo e sua grandiosa amizade.

Aos meus pais, Rosa Maria de Oliveira Zanini e João Bega Zanini, sem eles nada seria possível.

Ao meu orientador, Kléber de Oliveira Andrade, pela dedicação inspirando-me e viabilizando a realização deste trabalho.

Ao professor Cleberson Eugenio Forte pelos seus esforços em proporcionar a melhor experiência de aprendizado possível aos seus alunos.

A todos os meus amigos e amigas que me acompanharam nessa grande jornada.

“C'est le temps que tu as perdu pour ta rose qui fait ta rose si importante” (Antoine de Saint-Exupéry).

RESUMO

A corrida incessante do homem contra o tempo e a globalização do mercado mundial nos trouxe neste momento onde métodos ágeis são aplicados no desenvolvimento de softwares, dentre tais metodologias destaca-se o Scrumban, uma iniciativa recente e promissora. Em paralelo outro tópico que tem demonstrado pioneirismo e inovação é o uso de *serious games* na área da saúde. A fim de compreender suas origens e estruturas, este trabalho tem como objetivo estudar a aplicação do Scrumban no desenvolvimento de *serious games* na área da saúde. Para o desenvolvimento do trabalho, realizou-se uma pesquisa bibliográfica que envolveu os temas: *serious games*, metodologias ágeis e a aplicação de soluções tecnológicas na área da saúde. A partir dos dados levantados, fez-se uma análise sobre os processos envolvidos na aplicação do Scrumban no desenvolvimento de *serious games* voltados para a área da saúde destacando as adaptações necessárias para que o método possa servir de estrutura para este tipo específico de projeto de software. Concluiu-se que é possível conduzir um projeto de desenvolvimento de um *serious game* voltado para a saúde usando como ferramenta o Scrumban, tal estudo abre um leque de possibilidades para projetos futuros de caráter pioneiro e inovador.

Palavras-Chave: Scrumban, *Serious Games*, Saúde.

ABSTRACT

The man's incessant race against time and the globalization of the world market has brought us to this point where agile methods are applied in software development, among these methods Scrumban is highlighted, which is a recent and promising initiative. In parallel another topic that has demonstrated pioneering and innovation is the use of serious games for health. In order to understand its origins and structures, this work aims to study the application of Scrumban in the development serious games for health. In order to develop this work a literature search has been carried out involving these topics: serious games, agile methodologies and the application of technological solutions in healthcare. From the data collected, an analysis was conducted about the processes involving the application of Scrumban in the development of serious games focused on healthcare highlighting the adjustments needed in order to the method properly serve as a framework for this particular type of software project. It was concluded that it is possible to lead a project to develop a serious game for health using Scrumban, this study opens up a range of possibilities for pioneering and innovative future projects.

Keywords: Scrumban, Serious Games, Health.

LISTA DE ABREVIATURAS E SIGLAS

Daily – Daily Meeting

IA – Inteligência Artificial

ILO – Intended Learning Outcomes

JIT – Just in Time

Nº - Número

OMG - Object Management Group

PERT - Program Evaluation and Review Technique

PO - Product Owner

Retro - Sprint Retrospective

RVA – Realidade Virtual Aumentada

SG – Serious Game

SGA – Serious Games Association

SM - Scrum Master

STP - Sistema Toyota de Produção

TEPT - Transtorno de Estresse Pós-traumático

TPS - Toyota Production System

UML - Unified Modeling Language

WIP - Work In Progress

XP - eXtreme Programming

LISTA DE FIGURAS

Figura 1 - Resumo do desenvolvimento ágil e seus ciclos	6
Figura 2 - Visão resumida do Scrum	7
Figura 3 – Exemplo de Diagrama de Caso de Uso	9
Figura 4 – Exemplo de estrutura de nota de Estória por Usuário.....	11
Figura 5 – <i>Story Board</i> aplicado na prática	12
Figura 6 – Exemplo de um <i>Product Backlog</i>	14
Figura 7 – Visualização dos processos de uma <i>Sprint</i>	20
Figura 8 – Exemplo visual entre incremento e iteração.....	20
Figura 9 – Ordenação da documentação com o decorrer do projeto	24
Figura 10 – Exemplo de Gráfico de <i>Burndown</i> com uma linha ideal de progresso ...	27
Figura 11 – Exemplo de um time com um desempenho abaixo do esperado	28
Figura 12 – Exemplo de um time com um desempenho acima do esperado	29
Figura 13 – Exemplo visual do <i>One Piece Flow</i> na manufatura.....	37
Figura 14 – Amostra dos processos envolvidos na métrica de <i>Lead Time</i>	38
Figura 15 – Ciclo ideal de uma <i>Sprint</i> segundo o Scrum	46
Figura 16 – Fluxo ideal de trabalho segundo o Kanban	47
Figura 17 – Ciclo ideal de iteração segundo o Scrumban	48
Figura 18 – Paciente durante tratamento de fobia relacionada à altura	57
Figura 19 – Visual dentro do jogo da paciente sob tratamento da fobia de altura.....	57
Figura 20 – Imagem do jogo Fear of Spiders Treatment HD disponível para iPad ...	58
Figura 21 – Full Spectrum Warrior, utilizado na diagnose e tratamento do TEPT em veteranos de guerra do exército norte americano	59
Figura 22 – The Joyboard primeiro controlador voltado para <i>Exergames</i>	60
Figura 23 – Imagem do jogo Snacktown Snackdown da série The Incredible Adventures of the Amazing Food Detective	62
Figura 24 – Imagem do jogo Free Dive utilizado na terapia da distração.....	63
Figura 25 – Pet Pals: Animal doctor simulador de veterinária	64
Figura 26 – RVA no treinamento e simulação de intervenções cirúrgicas	65
Figura 27 – Cena do <i>serious game</i> voltado à educação ambiental, Pappi World	67
Figura 28 – Exemplificação visual da presença dos estágios durante um projeto	70
Figura 29 – Conjunto de lançamentos num projeto de múltiplos anos de duração ...	70

LISTA DE TABELAS

- 1 – Diferenças entre o Scrum e o Kanban.
- 2 – Diferenças entre jogos para entretenimento e serious games.

SUMÁRIO

1. INTRODUÇÃO	1
1.1 OBJETIVO.....	1
1.2 JUSTIFICATIVA.....	2
1.3 ESTRUTURA DO TRABALHO.....	4
2. MÉTODOS ÁGEIS	5
2.1 SCRUM.....	7
2.1.1 Levantamento de Requisitos	8
2.1.1.1 Casos de Uso	8
2.1.1.2 <i>User Story</i>	10
2.1.1.3 <i>Story Board</i>	11
2.1.2 Product Backlog	13
2.1.3 Planejamento e Estimativas	14
2.1.3.1 Histórico de Informações.....	15
2.1.3.2 Especialistas	15
2.1.3.3 PERT	16
2.1.4 Story Points	17
2.1.4.1 <i>Planning Poker</i>	17
2.1.5 Release Backlog	18
2.1.6 Sprint	19
2.1.6.1 <i>Sprint Planning</i>	21
2.1.6.2 <i>Sprint Velocity</i>	21
2.1.6.3 <i>Sprint Backlog</i>	22
2.1.6.4 Documentação	23
2.1.6.5 Desenvolvimento	24
2.1.6.5.1 <i>Daily Meetings</i>	25
2.1.6.6 Verificação.....	25
2.1.6.7 Gráfico de <i>Burndown</i>	26
2.1.6.8 <i>Sprint Retrospective</i>	29
2.1.6.9 <i>Sprint Review</i>	30

2.1.7	Gráfico de <i>Burndown</i> e as Mudanças de Escopo	30
2.1.8	Recursos Humanos.....	31
2.1.8.1	Papéis	31
2.1.8.2	<i>Product Owner</i>	32
2.1.8.3	<i>Scrum Master</i>	32
2.1.8.4	Time	33
2.1.8.5	Usuários	33
2.1.8.6	<i>Stakeholders</i>	33
2.1.8.7	Gerentes	33
2.2	KANBAN.....	35
2.2.1	Características básicas e ferramentas.....	36
2.2.2	Hierarquia	39
2.2.3	Flexibilidade	40
2.3	SCRUMBAN	41
2.3.1	Scrum X Kanban.....	41
2.3.2	Scrum e Kanban	43
2.3.3	Implantação	44
2.3.3.1	Ciclo Produtivo	46
3.	SERIOUS GAMES	52
3.1	<i>SERIOUS GAMES</i> PARA SAÚDE.....	54
3.1.1	Auxiliares de terapia	56
3.1.2	Promoção da saúde e condicionamento físico	59
3.1.3	Monitoramento	62
3.1.4	Treinamento.....	63
3.1.5	Métodos ágeis e <i>serious games</i>	65
4.	SCRUMBAN APLICADO A JOGOS	68
4.1	ESTÁGIOS NO PROJETO DE UM JOGO.....	69
4.1.1	Conceito.....	71
4.1.2	Pré-Produção.....	71
4.1.3	Produção.....	72
4.1.4	Pós-Produção.....	72
4.2	SCRUMBAN PARA <i>SERIOUS GAMES</i> NA ÁREA DA SAÚDE.....	73

5.	CONCLUSÃO.....	77
5.1	TRABALHOS FUTUROS	78
6.	REFERÊNCIAS BIBLIOGRÁFICAS	79

1. INTRODUÇÃO

Partindo da premissa de que todo objetivo traçado tem por trás um motivo inicial, justo é então apresentar primeiramente o motivo pelo qual o trabalho que será exposto foi desenvolvido, o que pode ser resumido em uma pergunta, o quanto ideal é a aplicação de métodos ágeis no desenvolvimento de *serious games* na área da saúde?

Atualmente, com a evolução constante das tecnologias e dos métodos que embora o desenvolvimento de aplicações computacionais, a agilidade e a inovação são buscadas incessantemente. Da análise deste cenário, três hipóteses foram levantadas:

- A aplicação de métodos ágeis no desenvolvimento de *serious games* provê resultados positivos e efetivos.
- O Scrumban é um equilíbrio entre os dois métodos ágeis de grande ascensão na área de desenvolvimento de software, Kanban e Scrum.
- O emprego de métodos ágeis é compatível com os processos de desenvolvimento de um *serious games* na área da saúde.

Assim, inserido neste contexto, foi possível dar início a definição do objetivo pelo qual este trabalho foi desenvolvido e os termos que justificam a execução do mesmo, pontos que são levantados logo a seguir.

1.1 OBJETIVO

Entender os processos envolvidos no desenvolvimento de um *serious game* para a área da saúde quando aplicado um método ágil, especificamente o Scrumban, é o objetivo geral dessa monografia. De forma mais específica, procura-se estudar os conceitos, composição geral e concepção de um *serious game*, averiguando a tecnologia e os processos envolvidos por trás deste tipo específico de cenário de desenvolvimento.

Definindo o mesmo de forma mais específica, procura-se estudar os conceitos, composição geral e concepção de um *serious game*, averiguando a

tecnologia e os processos envolvidos por trás deste tipo específico de cenário de desenvolvimento.

Realizando o mesmo através de pesquisas bibliográficas para compreender o funcionamento dos métodos ágeis e como decorre seu ciclo de vida sobre o projeto a ser executado, investigando também a existência de iniciativas atuais de desenvolvimento de *serious games* que utilizem de métodos ágeis como ferramenta organizacional.

Para que o escopo deste projeto seja realizado com sucesso, as seguintes metodologias de pesquisa serão utilizadas tendo como base as premissas estabelecidas por Andrade (2009):

- Pesquisa Básica: Têm como objetivo a geração de novos conhecimentos úteis para o desenvolvimento científico sem aplicações práticas previstas. Envolve verdades e interesses gerais.
- Pesquisa Bibliográfica: Baseia-se na elaboração do trabalho a partir de materiais já publicados, constituído principalmente de livros, artigos periódicos e atualmente com material disponibilizado na internet.
- Pesquisa Exploratória: Busca proporcionar familiaridade com o problema, com o intuito de torná-lo explícito ou a construir hipóteses. Envolve levantamento bibliográfico, entrevistas com pessoas que possuam experiências práticas com o assunto/problema pesquisado, análise de exemplos que estimulem a compreensão.

1.2 JUSTIFICATIVA

O desenvolvimento de *serious games* para saúde toma partido como uma iniciativa inovadora tanto na área de desenvolvimento de jogos digitais quanto na área da saúde. Apesar de recente, uma considerável quantidade de informações relevantes ao tópico encontra-se disponível ao redor mundo através de publicações e artigos científicos, mas até que ponto essa tecnologia é relevante na vida das pessoas? Ao pensar nessa questão caímos em um velho paradigma, jogos digitais não são meras ferramentas de entretenimento?

“Jogos digitais não podem ser somente utilizados para diversão e entretenimento. O termo "*serious games*" (SG) denota jogos digitais servindo a propósitos com seriedade como educação, treinamento, propaganda, pesquisa e saúde. Recentemente, uma nova geração de jogos têm-se aproveitado dos movimentos corporais de forma completa.” (WIEMEYER e KLIEM, 2011, p. 41. Tradução nossa).

De acordo com Wiemeyer e Kliem (2011) torna-se evidente que jogos não são meros instrumentos de entretenimento e que os mesmos podem ser utilizados para servir a outros propósitos. É neste contexto que se introduz o conceito de *serious games* aplicando-o na área da saúde.

Os inúmeros benefícios sociais que este tipo de aplicação tem o potencial de prover justificam os esforços na imersão e estudo deste tema.

A área específica envolvida no trabalho é o desenvolvimento de *serious games* para saúde, especificamente focando na utilização do método ágil Scrumban, com o intuito de estudar e analisar os processos envolvidos e prover resultados quanto sua eficiência e eficácia no desenvolvimento do projeto.

A abordagem deste tema é de extrema relevância, pois trata da união de duas iniciativas muito recentes e com alto caráter de inovação que são a utilização de jogos digitais na área da saúde juntamente com o Scrumban, que é um método ágil ainda em processo de desenvolvimento e adaptação no mercado.

Além de servir de base de estudo e comparação entre os métodos tradicionais de desenvolvimento, também servirá como base de informação e guia para quem tiver interesse em desenvolver projetos de jogos voltados a área da saúde e tiver como objetivo utilizar-se de métodos ágeis atuais.

O projeto é viável pela facilidade na obtenção de recursos em geral para execução do mesmo. Para desenvolver aplicações como jogos digitais não são necessários recursos computacionais de última geração, logo, não demandando investimento financeiro além de um computador de uso pessoal. Existem no mercado ferramentas relacionadas a tais práticas de livre uso para estudo acadêmico, assim, não se faz necessário o investimento em licenças.

Métodos ágeis estão em alta no mundo acadêmico e no mundo dos negócios. A realização de seu levantamento bibliográfico é menos escassa que a dos outros tópicos a serem abordados, facilitando assim a imersão inicial no trabalho.

1.3 ESTRUTURA DO TRABALHO

Este primeiro capítulo foi reservado para a introdução ao trabalho.

O segundo capítulo destina-se aos métodos ágeis abordando dois de seus prodigiosos filhos, o Scrum e o Kanban no intuito de dar embasamento teórico para melhor compreensão do método derivado de sua união que é parte do tema deste trabalho o Scrumban.

No terceiro capítulo são abordados os *serious games* e posteriormente suas variadas aplicações na saúde e suas respectivas categorias.

No quarto capítulo é demonstrado como o Scrumban é aplicado no desenvolvimento de um jogo digital e então como o mesmo adapta-se para servir de ferramenta de apoio para um projeto de *serious game* voltado para a área da saúde.

O quinto capítulo é destinado à conclusão.

E por fim o sexto capítulo as referências bibliográficas.

2. MÉTODOS ÁGEIS

Segundo relatos de Keith (2010), na década de oitenta, a reação contra metodologias de cascata estava em ritmo crescente.

Grandes projetos dos setores militar e de TI foram falhando com frequência cada vez maior. Isso levou a elaboração de numerosos livros e artigos que almejavam encontrar melhores práticas para o desenvolvimento de projetos relacionados à área de tecnologia.

Algumas dessas metodologias, tais como a entrega evolutiva, promoveram o desenvolvimento incremental de produtos usando iterações, cada iteração continha uma fatia de todas as fases de desenvolvimento, em vez de o desenvolvimento ser espalhado por todo um ciclo da cascata.

As iterações podem ser tão curtas quanto uma semana, mas sempre incluindo os processos de análise, planejamento, codificação, integração e testes dentro desse prazo, ao invés de espalhar cada um deles ao longo dos anos, como eles provavelmente estariam em um projeto em cascata.

Muitos processos iterativos emergentes e metodologias incrementais foram referidas como métodos leves até que em 2001 um grupo de especialistas e intelectuais sobre o assunto reuniram-se na cidade de Snowbird, em Utah, nos Estados Unidos da América, e decidiram por chamá-los de metodologias ágeis.

O resultado deste encontro foi a criação do "manifesto ágil", que resume os principais valores e princípios desses métodos leves, posteriormente o grupo formou a Agile Alliance responsável pela disseminação da cultura ágil denominada Agile.

O Manifesto Ágil é composto por doze princípios que expressam a linha de pensamento partilhada em comum por todos os idealizadores dos métodos ágeis, estes mandamentos podem ser resumidos nas seguintes 4 afirmações também contidas nas descrições do próprio manifesto:

“Indivíduos e entre eles mais que processos e ferramentas.

Software em funcionamento mais que documentação abrangente.

Colaboração com o cliente mais que negociação de contratos.

Responder a mudanças mais que seguir um plano.

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.” (MANIFESTO ÁGIL, 2001).

Estes valores permitiram que várias metodologias ágeis, como Scrum, Crystal, e XP pudessem compartilhar de uma filosofia em comum, com esta união sua presença torna-se sólida e firme, aumentando sua influência e dominância no mercado mundial que é cada dia mais globalizado e imediatista.

O conceito de método ágil pode ser melhor compreendido visualmente na figura 1, que mostra seus ciclos de desenvolvimento, tais ciclos e seus conceitos entenderemos melhor no próximo capítulo ao abordar o método ágil Scrum.

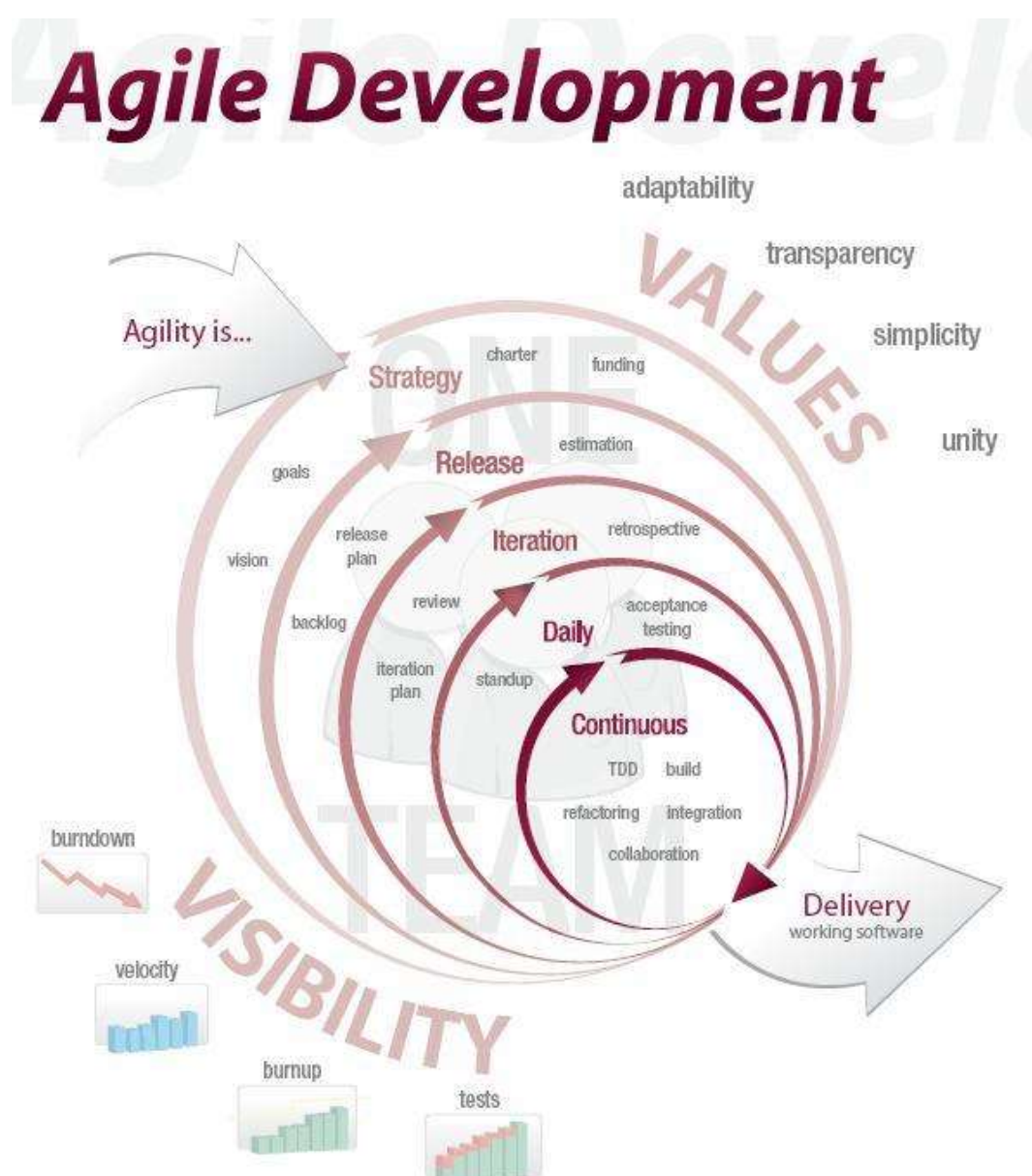


Figura 1 - Resumo do desenvolvimento ágil e seus ciclos (STRICKLER, 2008).

2.1 SCRUM

Segundo Keith (2010), a primeira aparição do Scrum como modelo para desenvolvimento de software foi na obra *Wicked Problems, Righteous Solutions* (DEGRACE E STAHL, 1990). Tal modelo foi aplicado pela primeira vez na “Easel Corporation.” por Jeff Sutherland e Ken Schwaber iniciando as primeiras obras práticas com a metodologia.

Observado o Scrum de uma forma geral (Figura 2), denota-se a busca pela geração de valor e a preocupação com entrega de uma parcela de produto útil dentro de curtos espaços de tempos de forma interativa e incremental, procurando utilizar-se das mudanças de escopo do projeto a favor da melhoria e evolução do mesmo.

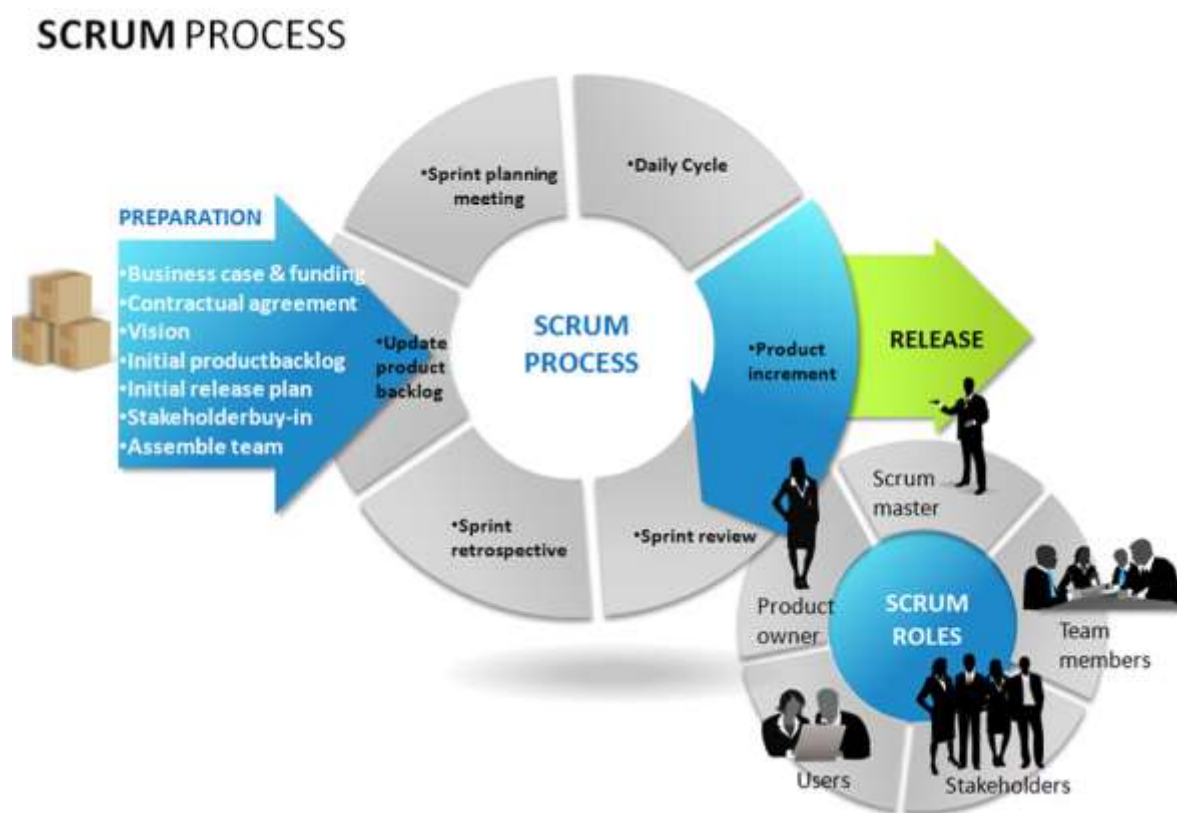


Figura 2 - Visão resumida do Scrum (THOMPSON, 2012).

2.1.1 Levantamento de Requisitos

Pries e Quigley (2011) explanam que inicialmente existem diversas maneiras e métodos de realizar-se o levantamento de informações quanto às necessidades do cliente, passo essencial para que a execução e conclusão sejam bem sucedidas, tais métodos não são exclusivos do Scrum, mas tratando deste tipo de tópico relacionado ao planejamento e gerência no âmbito de desenvolvimento de software sua inclusão é válida. Dentre tais métodos podemos citar os seguintes: análise de orientação/escopo do produto, modelagem de papéis, entrevista com o consumidor, questionários e observação de consumidores potenciais (etnografia).

No desenvolvimento de qualquer projeto de produto, incluindo projetos de Scrum, as ideias quanto ao conteúdo do produto vem costumeiramente do próprio consumidor, entretanto o desenvolvedor também pode adicionar algumas inovações (PRIES e QUIGLEY, 2011).

Tais adições precisam ser racionalmente analisadas antes de implementadas, pois focar demais na exuberância visual do produto, por exemplo, pode gastar um tempo que poderia ser melhor aplicado realizando outras adições de significância e valor muito superior ao cliente. Qualquer adição deve ser parcimoniosa, afinal de nada valerão os esforços se o fruto dos mesmos prejudicar a qualidade, segurança ou qualquer outro ponto atrelado às necessidades e desejos do consumidor.

Seguindo os dogmas do Manifesto Ágil (2001), o Scrum prega o contato frequente com o usuário, podendo este ser de mensal até diário, visando reduzir ou até mesmo eliminar o risco proveniente das mudanças.

2.1.1.1 Casos de Uso

Existem diversas ferramentas para coletar as informações necessárias ao desenvolvimento do produto com o cliente, uma que é fortemente recomendada ao Scrum é a utilização de Casos de Uso.

“A captação de requerimentos faz parte do planejamento para o desenvolvimento de qualquer tipo de projeto, portanto encontra-se no Scrum e desenvolvimento ágil. Uma abordagem padronizada para tal é o uso de Casos de Uso, uma técnica definida pela Object Management Group (OMG) em algumas Linguagens Unificadas de Modelagem (UML) especificações dadas pela última década e meia. Sua simplificada abordagem gráfica e textual dinamiza a captura de requerimentos funcionais melhorando a interação com os consumidores com o desenvolvimento do produto.” (PRIES ; QUIGLEY, 2011, p. 16. Tradução nossa).

Tal ferramenta não é prescrita pelo Scrum ou quaisquer outras metodologias ágeis, mas sua simplicidade e usabilidade são suficientes para sanar as necessidades comuns atreladas à documentação quanto a métodos ágeis como o Scrum.

Um exemplo genérico de diagrama de caso de uso pode ser visualizado na figura 3.

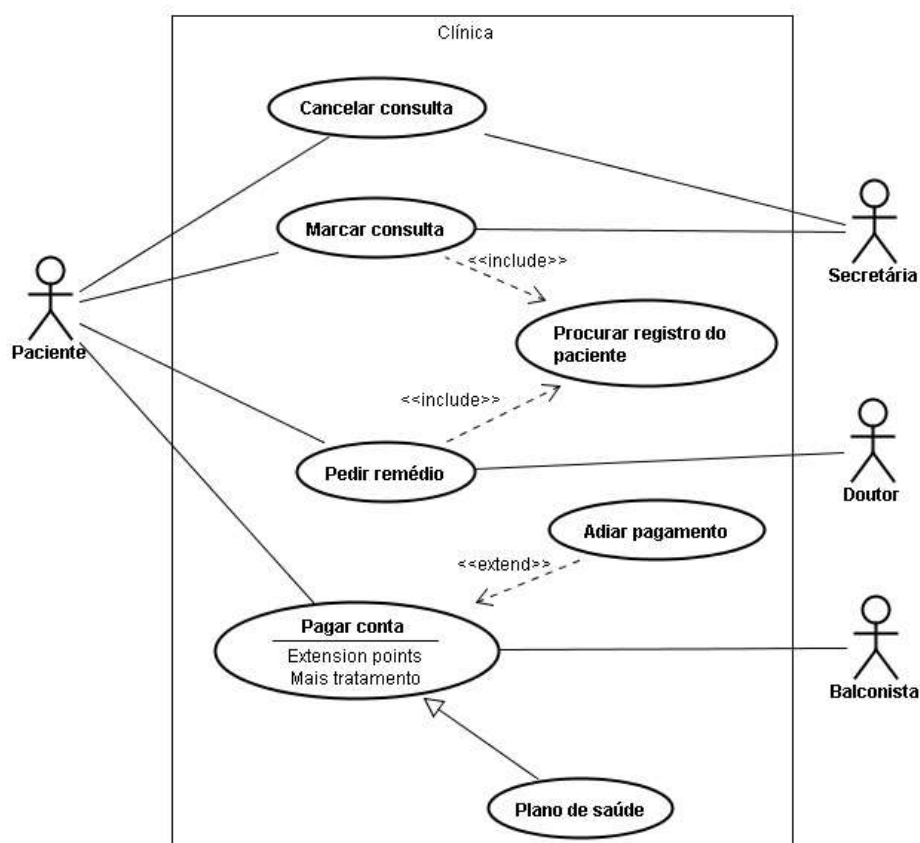


Figura 3 – Exemplo de Diagrama de Caso de Uso (SAMPAIO, 2007).

2.1.1.2 *User Story*

A *User Story* ou Estória por usuário é um tipo de abordagem fortemente empregada no Scrum quando se trata de levantamento de requisitos, como tal busca o mínimo de desperdício ao simplificar os processos tornando-os claros e concisos, diante das inúmeras mudanças que podem ocorrer durante o processo se utilizada uma abordagem tradicional de documentação detalhada e complexa, muito tempo seria desperdiçado com revisão e reestruturação.

O objetivo desta técnica é definir as funcionalidades a serem desenvolvidas, baseando-se nas ações que o usuário final tomará durante o uso do sistema para satisfazer suas necessidades, sendo assim o resultado é muito mais satisfatório quando o próprio cliente ou usuário está presente para auxiliar a equipe de desenvolvimento na criação das estórias aumentando assim a probabilidade de ao final de cada estória o cliente já ter em mãos uma parte de produto comercializável, tradicionalmente as estórias são transcritas em espécies de blocos de notas adesivos denominados *Post-it* e são expostos visualmente no *Story Board* outra ferramenta que será vista no capítulo seguinte.

Segundo Pries e Quigley (2011), antes de definir uma estória precisamos nos perguntar: (i) Quantos tipos diferentes de clientes vão utilizar este produto? (ii) Quais são os requerimentos deste produto para atender as necessidades dos clientes? (iii) Como diversos tipos de clientes utilizarão o produto?

Apesar de não ser prescrito, existe um modelo adotado pelo senso comum na transcrição de Estórias por Usuário apresentado abaixo:

Como um “Tipo de usuário” **eu desejo** “Ação a ser executada” **para que seja possível** “Motivo para tal ação”.

Por exemplo:

Como um **Administrador de conteúdo** eu desejo **visualizar, filtrar e classificar os artigos enviados pelos escritores** para que seja possível **decidir quais artigos serão publicados e em qual categoria no site**.

Têm-se um exemplo visual de tal estrutura na figura 4.



Figura 4 – Exemplo de estrutura de nota de Estória por Usuário (PRIMO, 2011).

Para o Scrum é mais interessante tomar conhecimento quanto às necessidades de negócio atreladas ao software ou infraestrutura do que ter uma documentação técnica muito profunda e exageradamente detalhada do produto. Quando possível é muito mais proveitoso quando o próprio cliente escreve as estórias de usuário para então posteriormente a equipe revisar e se necessário reestrutura-las de forma a englobarem e definirem funcionalidades a que virão a ser desenvolvidas, afinal a essência do escopo desta ferramenta está na busca pelo entendimento das necessidades do cliente na sua forma mais básica e pura, é importante ressaltar que o Scrum busca a simplificação e não a eliminação da documentação assumindo que tal é importante para a organização e controle do projeto, mas a mesma precisa ser enxuta e concisa para não causar empecilhos caso mudanças venham a ocorrer.

2.1.1.3 Story Board

Segundo Pries e Quigley (2011), o *Story Board* ou Quadro de Estórias (Figura 5), é utilizado para manter-se um controle visual da trajetória das estórias durante os estágios no desenvolvimento do software.

Tipicamente composto fisicamente por um quadro branco onde os *Post-Its* com suas Estórias e respectivas tarefas atreladas a tais são colados, o quadro é dividido de forma ordenada em seções que representam cada etapa do processo de desenvolvimento do software, tais etapas não são prescritas ou definidas partindo do processo que para cada tipo de projeto haverá uma diferente abordagem logo diferentes etapas envolvidas no seu desenvolvimento, mas é essencial que tais seções sejam criadas e que as estórias e suas respectivas tarefas sejam movidas no quadro conforme seu progresso durante o projeto para assim representar de forma visual e prática para todos os envolvidos no projeto como está seu andamento.



Figura 5 – *Story Board* aplicado na prática (GLOGER, 2008).

As Estórias por Usuário tem seu valor agregado nesta parte do processo facilitando o alinhamento e disposição das funcionalidades no quadro branco, os

diagramas de UML apesar de essenciais para o levantamento de requisitos não possuem um processo próprio para decantação das funcionalidades e tarefas envolvidas dificultando sua transcrição no quadro que é obrigatória para atender o princípio da transparência pregado pelo Manifesto Ágil.

2.1.2 *Product Backlog*

O *Product Backlog* (Figura 6) é a peça inicial dos processos envolvidos no Scrum, mas o que exatamente isso é podemos ver no trecho a seguir:

“Os requerimentos pelo sistema ou produto sendo desenvolvido pelo projeto(s) são listados no *Product Backlog*. O *Product Owner* é responsável pelo conteúdo, priorização e disponibilidade do *Product Backlog*. O *Product Backlog* nunca está completo e o *Product Backlog* utilizado no planejamento do projeto é uma mera estimativa inicial dos requerimentos. O *Product Backlog* é dinâmico; seu gerenciamento muda-o constantemente para identificar as necessidades ao produto para que seja apropriado, competitivo e útil. Enquanto o produto existir, o *Product Backlog* também existirá.” (SCHWABER, 2004, p. 19. Tradução nossa).

Pode-se então concluir que o *Product Backlog* é uma listagem resumida das funcionalidades e suas tarefas as quais pretendemos executar dentro de um determinado período de tempo. Associando com as Estórias por Usuário podemos então ter uma noção de como é feita a segregação das informações na composição do *Product Backlog*. Dentre os processos envolvidos na sua composição a priorização é de suma importância tanto para definir quais funcionalidades são mais necessárias para cliente, logo de maior valor, para que o time possa definir a prioridade e ordem com a qual serão desenvolvidas e entregues.

As a	I want to	So that (I can)	Business Value	Estimate
HR Manager	Publish new vacancies	Find candidates	80	20
Job Hunter	Apply for a job	Quickly apply for a job	80	40
HR Manager	Triage applicants	Politely eliminate unpromising candidates	50	8
Googlebot	effectively find and index all postings	Ensure that internet searchers can find job postings on this site	50	13
System Admin	quickly recognize and analyze system	ensure rapid resolution of technical problems	30	20

Figura 6 – Exemplo de um *Product Backlog* (CARLOS, 2012).

Pries e Quigley (2011) afirmam que ambos o *Product Backlog* e o *Sprint Backlog* provêm um cenário oportuno para implementação do Kanban, outro método ágil que veremos futuramente.

2.1.3 Planejamento e Estimativas

No Scrum como em qualquer outro tipo de metodologia relacionada ao desenvolvimento de um projeto, nenhuma decisão surge do nada e a existência de tal artefato se justifica pela necessidade de tomar atitudes decididas de forma racional, logo se faz presente o processo de planejamento e estimativas, mas o que seriam as estimativas no conceito do Scrum?

“Estimativas são parte das atividades de planejamento de um projeto, são sempre palpites, que poderiam variar selvagemmente em sua qualidade, às vezes de forma impensada sobre quais ações serão necessárias para produzir o entregável. Está limitação pode ser superada com a premeditação e dissecção das tarefas constituintes do entregável.” (PRIES e QUIGLEY, 2011, p. 22. Tradução nossa).

Através da colocação anteriormente feita é notável que o processo de realizar uma estimativa quanto a uma tarefa envolve a intuição do desenvolvedor que se propõe a executá-la – e envolve, também, o uso do raciocínio e um pré-planejamento, mais do que algum sexto sentido ou predição mística. Existe uma série de ferramentas que os desenvolvedores podem utilizar para realizar a estimativa de tarefas com certo grau de precisão e certeza, ferramentas que apresentaremos em seguida.

2.1.3.1 Histórico de Informações

Através dos relatos de Pries e Quigley (2011) uma forma simples e muito eficaz de reduzir o nível de incerteza quando realizando uma estimativa é a utilização de qualquer método de Histórico de Informações, comumente empregado em organizações e negócios de grande porte, a numerosa presença indivíduos e seus respectivos times e projetos passados é uma excelente fonte de informação em meio ao nada, quando se possui arquivos e documentos quanto seu desempenho passado (isto assumindo que o desempenho passado é uma predição do desempenho futuro).

Apesar deste tipo de informação ter suas limitações quando se trata de abordar uma nova tecnologia ou fornecedor, ainda pode ser utilizada para mitigar possíveis riscos comparando o novo cenário com o aspecto histórico.

2.1.3.2 Especialistas

Quando uma organização possui profissionais altamente conceituados na execução de certo tipo específico de atividade comumente estes profissionais são eleitos especialistas em determinados campos e áreas de conhecimento sendo então requisitados quando alguma estimativa relacionada a uma de suas áreas de dominância é relacionada, funcionando como o Histórico de Informações, mas de forma muito mais pessoal, pois neste caso a informação que vai clarificar o cenário

para realização da estimativa vem diretamente um profissional experiente na área, é extremamente preferível acatar o conselho ou parâmetros passados por um especialista do que desperdiçar tempo discutindo as variáveis relacionadas a uma tarefa ou problema em particular onde paira a incerteza.

2.1.3.3 PERT

Assim como em outros métodos convencionais de desenvolvimento de projetos o uso do PERT (*Program Evaluation and Review Technique* termo em inglês para “Programa de Avaliação e Revisão Técnica”) pode auxiliar na estimativa do tempo de duração na execução de uma atividade. Através de cálculos matemáticos baseados em fórmulas estatísticas. O PERT procura mensurar um campo de possíveis datas para tarefas de risco elevado. No Scrum utilizamos esta ferramenta para ter uma média da possível duração das tarefas, sendo neste caso levemente mais próxima do resultado real mais provável.

Segundo Pries e Quigley (2011) o Departamento de Defesa dos Estados Unidos da América classifica o sistema de estimativa como pessimista, otimista e mais provável através das seguintes fórmulas:

$$\text{Duração} = [(\text{Pessimista} + 4 \times \text{Mais provável} + \text{Otimista}) / 6]$$

Valores os quais são dados pelo futuro executor da tarefa, dando assim ao gestor do projeto um indicador médio de certeza, então o gestor utilizara-se destes valores para calcular a variância da atividade, para ver o qual incerta está a predição através da seguinte fórmula.

$$\text{Variância da atividade} = [(\text{Pessimista} - \text{Otimista}) / 6]^2$$

Variações no PERT implicam em incerteza absoluta. Entretanto se o gestor assumir que as estimativas seguem uma distribuição normal, então ele pode refinar ou ampliar as estimativas.

É possível expandir o PERT utilizando-se do conceito Monte Carlo de simulação, se houver um Histórico de informações possíveis suficientes para que as fórmulas estatísticas sejam aplicadas.

2.1.4 *Story Points*

O conceito de *Story Points* ou Pontuação por Estória é uma maneira de estimar a quantidade de esforço a ser realizado no desenvolvimento de uma funcionalidade, as quais foram previamente denominadas como Estórias.

Essencialmente a Pontuação por Estória representa uma unidade de medida definida pelo time, não é definida nenhuma regra sobre como deve ser a ordem e progressão dos valores, somente que eles devem ser um senso comum entre o time auxiliando no processo de estimativa. Um ponto de Estória pode incorporar múltiplos dias de *design*, codificação e teste dependendo da abordagem que o time assim preferir. Alternativamente o time pode decidir entre usar dias ou horas.

O time determina quantos Pontos de Estória podem ser realizados. Isso ajuda a determinar quais estórias serão desenvolvidas em quais iterações.

2.1.4.1 *Planning Poker*

Esta é uma ferramenta muito utilizada por times no Scrum para estimar a complexidade do trabalho que terão de realizar, durante tal atividade todo time é envolvido, às vezes o *Product Owner* pode participar, mas somente como auditor ou observador.

Segundo as descrições feitas por Pries e Quigley (2011), o método funciona da seguinte maneira, cada integrante do time recebe um baralho de cartas com valores de estimativas válidas. Um dos valores mais utilizados é a sequência de Fibonacci (1, 2, 3, 5, 13, 21, 34, 55 podendo prosseguir mais adiante ainda dependendo do caso).

O *Scrum Master* pode preparar ele mesmo o conjunto de baralhos antes da reunião, os quais devem ser escrito em número grande o suficiente para serem visíveis numa pequena sala de reunião, assumindo o pressuposto que dificilmente um time que trabalha com Scrum terá mais de seis integrantes na equipe.

Como o Scrum apoia a mentalidade enxuta inicialmente pregada pelo Lean e posteriormente adotada por muitas metodologias ágeis, é de bom grado manter o baralho para sessões futuras.

A sessão inicia-se com o *Scrum Master* apresentando e descrevendo oralmente os casos de uso ou estórias de usuário a serem estimadas. Após todas as dúvidas forem sanadas, cada membro escolherá confidencialmente um carta que acredita melhor representar o nível de complexidade daquela tarefa, então aquele que tiver escolhido o menor valor debaterá com o que escolheu o maior valor apresentando as razões pela qual decidiu por tal estimativa, a ideia é incentivar a discussão em grupo sem levar ao pensamento em grupo e dispersar o debate em pontos que não serão de todos úteis na definição final da estimativa daquela estória, lembrando que tal processo não é uma solução mágica para mitigar os possíveis erros que podem ocorrer durante a estimativa e o grau de incerteza das mesmas.

2.1.5 Release Backlog

Derivado do *Product Backlog* o *Release Backlog* é uma lista com a funcionalidades que pretende-se entregar ao cliente a cada entrega do produto. É uma sub porção do *Product Backlog* geralmente populada pelas estórias definidas como de maior prioridade para o cliente ou baseado em dependências técnicas. Lembrando que questões técnicas vêm primeiro que a vontade do cliente, mesmo tendo-se o foco em satisfazer o cliente em primeiro lugar vem à geração de valor, não é válido entregar primeiramente uma funcionalidade que venha a agradar o cliente se a mesma vai sofrer de complicações técnicas comprometendo o valor que a mesma vai agregar no final.

O *Release Backlog* é distribuído por entre uma quantidade específica *Sprints*, quantidade que é determinada pela velocidade de trabalho no time e sua eficiência na obtenção de resultados.

O *Release Backlog* pode ser então futuramente quebrado em várias *Sprint Backlogs* com um foco específico para cada *Sprint* subsequente, assim cada *Backlog* vai incorporando cada vez mais detalhes sobre o trabalho a ser realizado

para alcançar o objetivo final, produzir uma parcela de produto funcional a ser entregue para o cliente para obtenção de retorno.

2.1.6 *Sprint*

Inúmeros autores e idealizadores sobre métodos ágeis têm de sua própria forma clara e sólida definição do termo *Sprint*, mas dificilmente têm-se um consenso absoluto sobre seus termos, cruzando as ideias expostas por Schwaber (2004) e Pries e Quigley (2011), podemos afirmar que *Sprint* é uma iteração no ciclo de desenvolvimento de software do Scrum de cronograma fechado que se decorre dentro de um período que pode ter uma extensão máxima de média de 30 dias e mínima de média de 12 dias tendo como escopo ao chegar o fim da interação entregar ao cliente uma parcela de produto funcional ou o produto completo em si dependendo da magnitude e complexidade do projeto, as funcionalidades a serem desenvolvidas durante este período de iteração são definidas pelo *Sprint Backlog* conceito a ser explanado futuramente.

Toda funcionalidade que não é desenvolvida a tempo de acabar o *Sprint* retorna ao *Product Backlog*, uma das propriedades chave da *Sprint* é a flexibilidade do desenvolvimento que está ligada um dos princípios do Manifesto Ágil, apesar de não ser recomendado por existir a possibilidade surtir efeitos negativos no desempenho da produção o *Sprint Backlog* pode ser alterado no decorrer de uma *Sprint*, porem quando o time começa a trabalhar numa funcionalidade específica que pode estar atrelada a uma ou mais Estórias por Usuário não se pode substitui-la, caso o *Product Owner* em algum momento julgue que tal funcionalidade não será mais útil para o usuário final sua produção pode ser cancelada, mas substituir uma funcionalidade já em desenvolvimento não é permitido pelo fato de que isso destruiria o alinhamento feito entre todos os envolvidos durante o planejamento da *Sprint* fazendo com que a mesma quebre e tenha que recomeçar do principio.

Como o Scrum prega a resposta rápida à mudança uma opção permitida é a substituição de Estórias entre o *Sprint Backlog* e o *Product Backlog* desde que as mesmas ainda não estejam sendo desenvolvidas.

Podemos melhor visualizar o conceito de *Sprint* na figura 7.

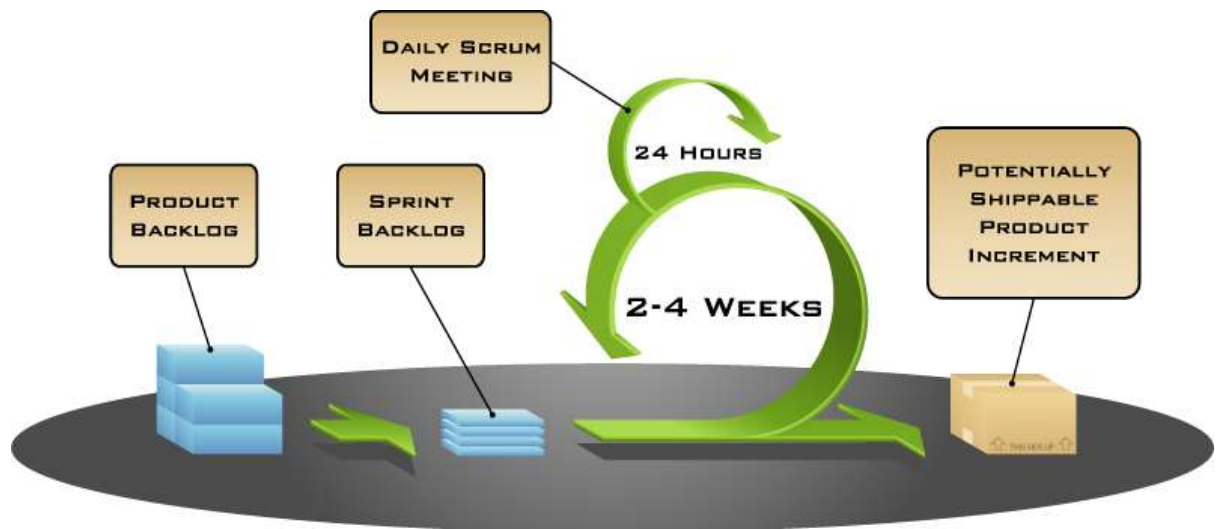


Figura 7 – Visualização dos processos de uma *Sprint* (COHN, 2005).

A *Sprint* é um processo de desenvolvimento iterativo e exceto quando o projeto é simples ao ponto de ser finalizado em uma única *Sprint* também é incremental, pois a cada ciclo de desenvolvimento (iteração) uma parte do software final é feita e a cada *Sprint* vão sendo incrementadas mais parcelas desse software até então acabarem-se as *Sprints* e ter o software em sua versão final em mãos.

A seguir na figura 8 temos um exemplo visual dos conceitos de incremento e iteração.

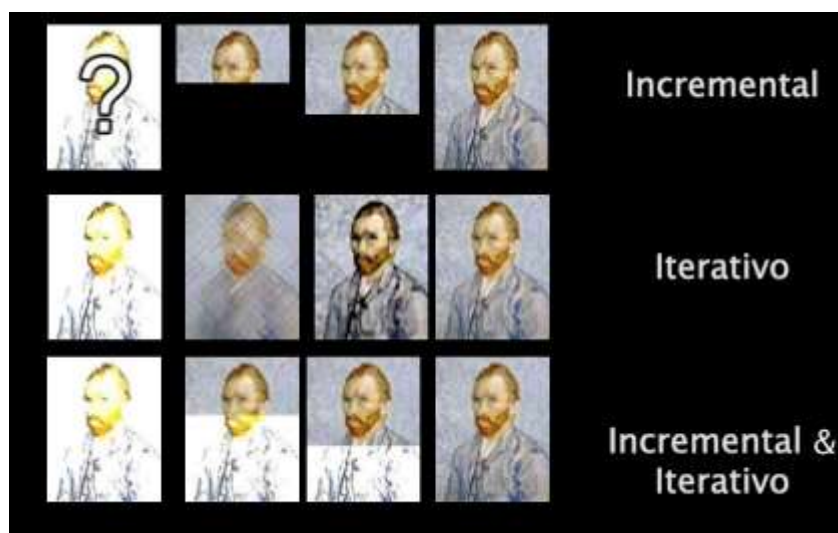


Figura 8 – Exemplo visual entre incremento e iteração (MORDEZKI, 2011).

De acordo com Simoes (2011) para garantir que cada *Sprint* desenrole-se sem sucumbir em completa anarquia existe um controle empírico sobre os processos baseado em três conceitos:

- **Transparência**, o primeiro e mais essencial dos conceitos, decorre-se na boa comunicação do time e seu alinhamento quanto o processo de desenvolvimento.
- **Inspeção**, processo que visam periodicamente averiguar se o projeto está devidamente direcionado segundo o escopo definido, uma ferramenta métrica que ajuda na execução deste processo é o Gráfico de *Burndown* que será devidamente apresentado futuramente.
- **Adaptação**, a capacidade de estar respondendo as mudanças e alterações de escopo no andamento do projeto tanto quanto a substituição de Estórias como visto anteriormente ou outras mudanças mais abrangentes a serem abordadas no capítulo 2.1.7.

2.1.6.1 *Sprint Planning*

Schwaber (2004) explica que o momento o qual o planejamento é realizado no Scrum é denominado *Sprint Planning* onde após o alinhamento entre o *Scrum Master* e o *Product Owner* o time reúne-se por um determinado período de tempo (não devendo ultrapassar a dois dias de reunião) para discutir e estimar as atividades a serem desenvolvidas na *Sprint*, momento onde então através do *Product Backlog* surgirá o *Sprint Backlog*, conceito a ser melhor explicado futuramente.

2.1.6.2 *Sprint Velocity*

De acordo com Pries e Quigley (2011), a *Sprint Velocity* ou velocidade da *Sprint* é determinada pela quantidade de horas disponíveis durante o definido período de trabalho.

Este termo é comumente referenciado a velocidade de trabalho do time que pode ser calculada através da seguinte fórmula matemática:

Velocidade da *Sprint* = N^o de Pessoas x Horas disponíveis x Dias de trabalho

Utilizam-se então os valores obtidos para realizar comparações com estimativas das tarefas e os resultados de cada *Sprint*. As horas disponíveis devem ser de fato a quantidade de horas que os integrantes do time de desenvolvimento estarão desenvolvendo e não sua carga horária de trabalho semanal, então não é permitido incluir o tempo gasto durante refeições, pausas e outras atividades.

A velocidade da *Sprint* tende a mudar com o progresso do projeto, é factível que todo projeto tenha uma eficiência reduzida em seu início devido a uma série de fatores dos quais podemos destacar a falta de familiaridade com a tecnologia utilizada, falta de familiaridade com o escopo visual e falta de entrosamento entre os desenvolvedores do time.

O *Scrum Master* deve esperar que a velocidade do time aumente de acordo com o decorrer do projeto, isto assumindo que o time não sofra bloqueios e atrasos devido a empecilhos tecnológicos.

Adicionar novos integrantes ao time de desenvolvimento não significa um aumento imediato na velocidade de produção, pois todo novo integrante levará um tempo para alinhar-se quanto ao projeto, efeito denominado de curva de aprendizado, ocasionalmente ele também pode consumir uma parcela de tempo produtivo de outros integrantes do time que necessitarão parar suas atividades atuais para lhe dar suporte e instrução.

2.1.6.3 *Sprint Backlog*

Pries e Quigley (2011) explana que o *Sprint Backlog* é um conjunto de estórias selecionadas a partir do *Product Backlog* que serão executadas dentro de uma determinada *Sprint*. O time identifica as tarefas de maior prioridade no *Product Backlog* e as funcionalidades as quais cada uma delas está atrelada e os prováveis meios necessários para concluí-las com sucesso.

Ocasionalmente uma determinada funcionalidade do *Product Backlog* pode depender de uma série de múltiplas histórias, precisando ser assim fragmentada em sub funcionalidades para que possa assim ser distribuída entre mais de uma *Sprint*.

É importante que tais atividades a serem desenvolvidas sejam eleitas pelo próprio time de desenvolvimento para que assim possa haver um comprometimento maior do time em relação aos objetivos.

2.1.6.4 Documentação

De acordo com Pries e Quigley (2011) algumas pessoas acreditam que a aplicação do Scrum em sua essência se resume a extinção da documentação o que é certamente um pensamento errôneo, o que o Scrum de fato prega é a minimização do impacto negativo causado no desempenho do projeto pela geração de documentação extremamente detalhada e tradicionalista.

A primeira ou as primeiras *Sprints* são frequentemente utilizadas para documentar os requisitos e a cada *Sprint* uma pequena parcela de documentação é adicionada até que todo o produto esteja documentada seguindo a linha iterativa e incremental do Scrum tanto no desenvolvimento quanto na documentação.

Sendo assim, a documentação é produzida dentro de um modelo *Just in Time*. Ao invés de passar longos meses documentando tudo que foi desenvolvido, o que certamente faria com que alguns conceitos pudessem ficar obsoletos ou serem deprecados no decorrer do projeto, a cada *Sprint* deve-se fazer uma revisão do que foi documentado na *Sprint* passada, atualizando as mudanças e incrementando as adições conforme o exemplo dado na figura 9.

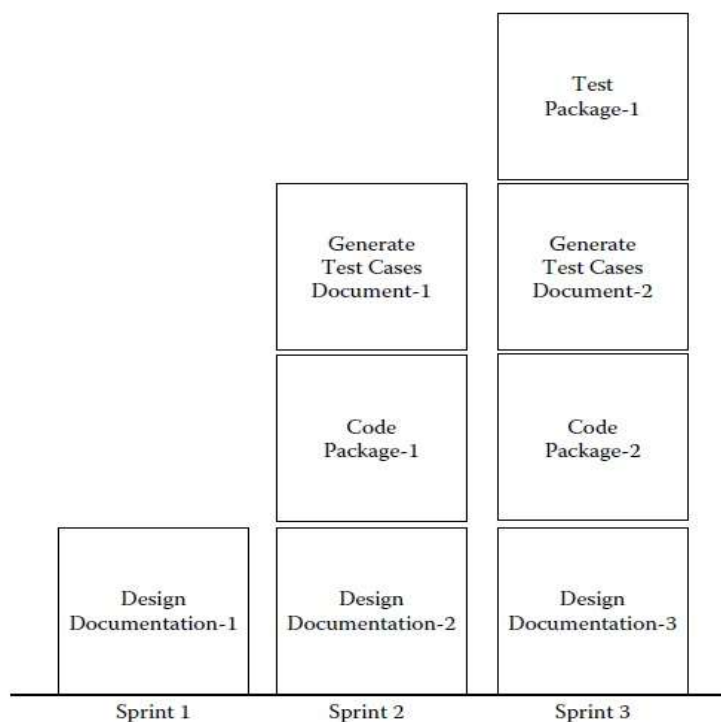


Figura 9 – Ordenação da documentação com o decorrer do projeto (PRIES e QUIGLEY, 2011)

2.1.6.5 Desenvolvimento

Uma vez que nós completamos a documentação do *design* durante a primeira *Sprint*, estamos prontos para iniciar o trabalho de desenvolvimento. O software é escrito ou a infraestrutura é montada utilizando as direções e detalhes provenientes do *Product Backlog* que já tem as prioridades definidas e a documentação de *design* feita (PRIES e QUIGLEY, 2011).

Conforme o trecho acima fica claro que a presença da documentação no Scrum além de real faz se necessária para que o desenvolvimento possa ocorrer de forma apropriada, um ponto interesse a se levantar é que problemas decorrentes durante o desenvolvimento de uma *Sprint* não são escondidos e deixados às traças, pelo contrário, são fortes candidatos a serem incorporados no *Sprint Backlog*, lembrando que o objetivo é entregar um produto utilizável então qualquer procedimento não planejado ou estimado que no decorrer do desenvolvimento demonstre-se por necessário para que tal objetivo seja alcançado, deverá ser

adicionado e simultaneamente o *Scrum Master* deverá reajustar o planejamento e estimativas feitas e se necessário alinhar com o *Product Owner* o cancelamento de alguma estória para que a *Sprint* seja de fato concluível.

Durante o processo de desenvolvimento para que o time mantenha-se alinhado e o projeto não fique desbalanceado e fora de sua trajetória é necessário um canal de comunicação entre todos, neste momento entram as *Daily Meetings*.

2.1.6.5.1 Daily Meetings

Diariamente o time deve se reunir para alinhar seu progresso e realizar comunicados pertinentes a todos os envolvidos no projeto, tais reuniões são denominadas *Daily Meetings* sendo comumente referenciadas como somente “*Daily*”.

Como o Scrum prega a agilidade, a reunião não ocorre de forma anárquica, segundo Schwaber (2004) a reunião desenrola-se com cada um dos integrantes respondendo três questões: (i) O que você fez neste projeto desde a última *Daily*? (ii) O que você pretende fazer neste projeto do presente momento até a próxima *Daily*? (iii) Existe algum impedimento não lhe permitindo cumprir o que pretende realizar para esta *Sprint* e projeto?

Terminada a reunião, o *Scrum Master* deverá tomar medidas para resolver os impedimentos de cada integrante do time caso algum tenha sido revelado durante a *Daily*, tais reuniões são essenciais para que o time esteja ciente de como está à trajetória do desenvolvimento do projeto e tenha uma noção real e precisa do que está acontecendo ao seu redor, mantendo-se assim alinhado.

2.1.6.6 Verificação

Através dos relatos de Pries e Quigley (2011) os membros da equipe de teste os quais são os responsáveis pelo processo de verificação daquilo que já foi desenvolvido geralmente são alocados juntamente com a equipe de *design*. A

documentação de *design* é utilizada na interação entre o time de desenvolvimento e o de teste para gerar as documentações de teste, que geralmente ocorrem em paralelo com o processo de desenvolvimento procurando suprir todas as especificações da *Sprint* anterior. Os casos de teste são derivados das documentações de *design*.

Esta colaboração mútua ajuda na clarificação de futuros *Sprint Backlogs* a serem entregues ao time de teste. Conforme defeitos são encontrados e falhas são expostas é feita uma comunicação imediata com a porção do time responsável pelo desenvolvimento e os erros comumente denominados como “*bug*” são adicionados ao *Sprint Backlog*, geralmente os erros e defeitos são monitorados num *backlog* a parte para erros e falhas, muitas vezes para que a equipe de teste consiga acompanhar o ritmo do projeto e garantir que as falhas serão passadas aos desenvolvedores a tempo de serem corrigidas antes do final da *Sprint* faz se necessário o uso de testes automatizados e customizados.

2.1.6.7 Gráfico de *Burndown*

Analisando os conceitos apresentados por Schwaber (2004) e Pries e Quigley (2011) pode-se dizer que o Gráfico de *Burndown* é uma ferramenta utilizada para visualizar a quantidade de trabalho já realizada pelo time e a quantidade de trabalho restante para que a *Sprint* esteja de fato completa, existem vários modelos de Gráficos de *Burndown*, possuem valores adicionais como a quantidade atual de tarefas e estórias para indicar adições feitas ao *Sprint Backlog* durante a *Sprint* ou a quantidade de horas planejadas e a quantidade atualmente necessária de horas para a conclusão da *Sprint* a fim de indicar se membros extras ou esforços extras dos membros atuais serão necessários, mas nos ateremos ao modelo tradicional do qual todos os outros derivam como pode ser observado na figura 10 abaixo.

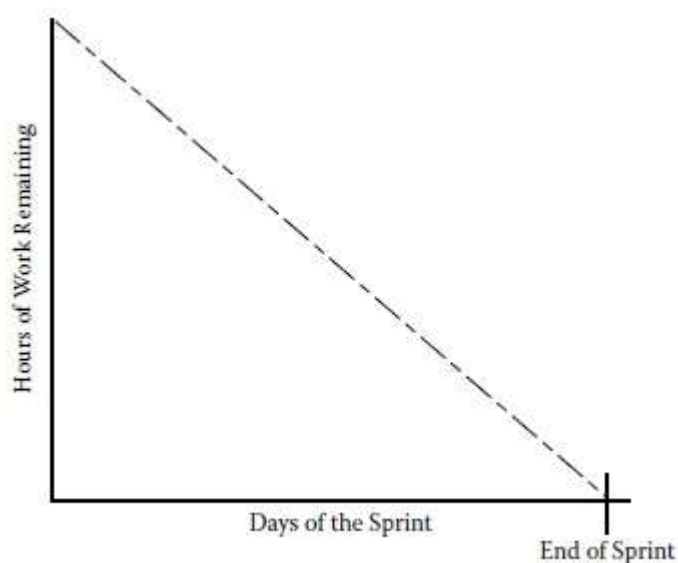


Figura 10 – Exemplo de Gráfico de *Burndown* com uma linha ideal de progresso (PRIES e QUIGLEY, 2011).

No eixo vertical do gráfico é representada a quantidade de horas restantes para a realização de todo o trabalho planejado para a *Sprint* e no eixo horizontal a quantidade de dias da *Sprint* tendo o último dia marcado por uma linha fazendo uma intersecção no final do eixo.

O desempenho do time é impresso no gráfico através de uma linha que toma início no ponto superior do eixo vertical indicando que no momento inicial da *Sprint* nenhum trabalho foi ainda realizado, a linha traçada indica o desempenho ideal o qual o time deve ter, certamente não é comum que um time siga a linha perfeitamente durante todo o decorrer do projeto, mas é uma atribuição do *Scrum Master* tomar medidas quando a divergência toma proporções notáveis.



Figura 11 – Exemplo de um time com um desempenho abaixo do esperado (PRIES e QUIGLEY, 2011).

Na figura 11 acima podemos ver um exemplo de um time que está com um desempenho abaixo do esperado, vários motivos podem levar a esse tipo de situação, imprecisão na estimativa das atividades ou até mesmo falha na análise das tarefas a serem desenvolvidas para satisfazer as estórias levando a criação de novas tarefas não planejadas anteriormente, quando isto ocorre o *Scrum Master* precisa analisar a situação do time e ver se o quadro é reversível, caso contrário ele precisa comunicar o *Product Owner* que alterações precisarão ser feitas no *Sprint Backlog* e talvez alguma estória deverá ser cancelada.

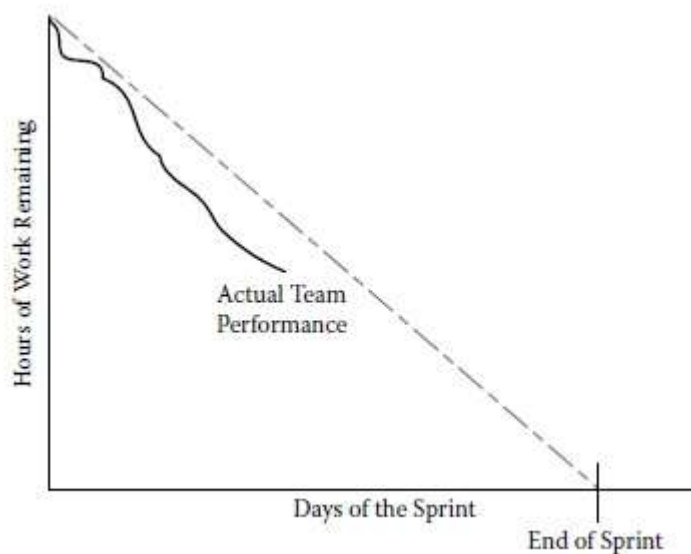


Figura 12 – Exemplo de um time com um desempenho acima do esperado (PRIES e QUIGLEY, 2011).

Na figura 12 podemos analisar um exemplo contrário ao anterior onde o time está com um desempenho acima do planejado, apesar da impressão positiva este tipo de cenário indica uma superestimação inicial das tarefas e falha no planejamento do *Sprint Backlog*, neste caso o *Scrum Master*, deve averiguar as causas relacionadas à super estimativa e se necessário incrementar itens do *Product Backlog* ao *Sprint Backlog*.

2.1.6.8 *Sprint Retrospective*

Pries e Quigley (2011) explanam que ao final de cada *Sprint* ocorre a *Sprint Retrospective* popularmente denominada como “Retro”, uma sessão de autocrítica do time em relação ao seu desempenho nessa *Sprint* passada onde cada integrante deve apontar os pontos positivos e negativos observados durante o decorrer da *Sprint*, para definir o que deve ser mantido e o que deve ser trabalhado para que melhore, para Schwaber (2004) a Retro é um reflexo da natureza empírica do Scrum que busca a melhoria contínua através da análise das atitudes tomadas no passado.

Como todo o tipo de reunião dentro do Scrum tem um padrão comportamental visando transparência e agilidade limitando as discussões aos pontos pertinentes ao bom andamento do projeto, inicialmente é eleito um líder da reunião que será responsável por ser o facilitador da reunião, coletando e registrando as informações passadas pelos membros do time, quase sempre este papel é do *Scrum Master*.

2.1.6.9 *Sprint Review*

Ao final de cada *Sprint* o objetivo é ter um produto entregável que seja comercializável (PRIES e QUIGLEY, 2011).

Ao final de cada *Sprint* ocorre a *Sprint Review*, onde as partes interessadas pelo produto final seja somente o *Product Owner* ou outras partes como os Stakeholders, usuários finais ou gerentes se encontram presentes para que lhes seja apresentado pelo time o que foi desenvolvido no determinado período de tempo.

Obviamente que o produto final não será apresentado na primeira *Sprint Review*, não faria de certa forma sentido, pois o Scrum é projetado para ser incremental e iterativo e tal fato não o tornaria tão diferente de outra metodologia escalonada como cascata ou RUP exceto que pelos seus ritos e processos internos, mas quando se trata de projetos absurdamente pequenos existe essa possibilidade.

Segundo os relatos de Pries e Quigley (2011) a intenção é demonstrar aos interessados o que foi desenvolvido e o que têm-se em mãos no momento, o ideal é que sempre seja entregue uma parcela funcional do produto a ponto que caso seja de interesse já possa ser testado ou até mesmo utilizado pelo cliente.

2.1.7 Gráfico de *Burndown* e as Mudanças de Escopo

De acordo com Pries e Quigley (2011) a abordagem Scrum é feita para ajudar a gerenciar as mudanças de escopo. Em geral mudanças e adaptações ocorrerão tanto por parte do time quanto por parte do cliente, assim como time pode passar por impedimentos técnicos que lhe obrigue a realizar mudanças ou descubra

alternativas para melhor satisfazer específicas necessidades do cliente atreladas ao produto, o próprio cliente ao longo do projeto pode descobrir que uma específica funcionalidade não lhe serve o propósito ou até mesmo descobrir novas necessidades as quais desconhecia previamente.

Schwaber (2004) afirma que os eventos do Scrum que são a *Sprint Planning*, *Daily Meeting*, *Sprint Retrospective* e *Sprint Review* tornam aplicável tal flexibilidade que o Scrum possui, tais eventos dependem diretamente dos artefatos do Scrum que são o *Product Backlog* e o *Sprint Backlog*, a união de todas essas ferramentas torna o Scrum o *framework* que é.

2.1.8 Recursos Humanos

Dentre os muitos recursos envolvidos, o mais decisivo e valioso para o sucesso de um projeto orientado pelo Scrum, são os recursos humanos, pois Schwaber (2004) diz que no Scrum os times são auto-gerenciáveis. Segundo Mordezki (2011) um time não deve ultrapassar a quantidade de nove integrantes destacando a importância de cada indivíduo no desenrolar do projeto.

2.1.8.1 Papéis

No Scrum há diversos papéis a serem exercidos para que o método seja executado com sucesso, Pries e Quigley (2011) explica que para melhor compreender os papéis envolvidos no Scrum primeiramente os mesmo são divididos em duas categorias, os porcos e as galinhas.

O motivo pelo qual são dados esses nomes é devido a um velho ditado, onde quando se trata de cozinhar, o porco está comprometido, mas a galinha está envolvida.

Dissecando este velho ditado, porcos são aqueles que estão diretamente ligados projeto e devem atender aos ritos e regras estabelecidas pelo Scrum sendo estes o *Scrum Master*, o *Product Owner* e o Time. Enquanto as galinhas são aqueles

que estão ligados ao projeto de forma indireta sem serem afetados por seus ritos e regras sendo estes os *Stakeholders*, Gerentes e Usuários.

2.1.8.2 *Product Owner*

O *Product Owner* é responsável por representar a voz do cliente.

Assim sua atribuição é inspecionar o trabalho do time sob uma visão de negócio, dando suporte na geração de estórias e criação do *Product Backlog*, acompanhamento na criação do *Sprint Backlog* e o decorrer de sua respectiva *Sprint* e por fim avaliar e realizar o processo de aprovação ao que foi desenvolvido na *Sprint Review*. Descrição baseada nos conceitos apresentados por Pries e Quigley (2011).

2.1.8.3 *Scrum Master*

O *Scrum Master* é responsável por conduzir as atividades relacionadas ao Scrum principalmente no início do projeto. Sua principal atribuição é agir como facilitador resolvendo todos os bloqueios e removendo quaisquer obstáculos que possam impedir a realização de uma tarefa, além manter o time livre de quaisquer interferências externas. Apesar de ser um cargo que requer um bom senso de liderança, tal liderança só é fortemente presente no início do projeto, posteriormente a figura do *Scrum Master* é mais aproximada a de um facilitador conduzindo as atividades, ajudando o time a cumprir com as regras e procedimentos prescritos pelo Scrum. Descrição baseada nos conceitos apresentados por Pries e Quigley (2011).

2.1.8.4 Time

O Time tem a responsabilidade de caracterizar, produzir, processar e entregar o produto ou serviço. Comumente composto de cinco a nove indivíduos multifuncionais com habilidades apropriadas para o desenvolvimento do projeto. Descrição baseada nos conceitos apresentados por Pries e Quigley (2011).

2.1.8.5 Usuários

Os usuários são para quem os projetos são desenvolvidos, são aqueles que por fim irão usufruir do produto ou serviço desenvolvido no decorrer do projeto. Descrição baseada nos conceitos apresentados por Pries e Quigley (2011).

2.1.8.6 Stakeholders

Os *Stakeholders* são as pessoas para qual o projeto surtirá algum tipo de impacto. Estes podem ser membros da equipe, clientes, acionistas, fornecedores, gerentes, clients internos, fornecedores internos, a comunidade local, enfim qualquer pessoa para quem o projeto tenha relevância. Os *Stakeholders* podem participar das *Sprint Reviews*, mas geralmente não estão envolvidos nas atividades diárias. Descrição baseada nos conceitos apresentados por Pries e Quigley (2011).

2.1.8.7 Gerentes

Os gerentes têm responsabilidades funcionais diferenciadas quanto ao desenvolvimento do projeto, eles geralmente são os proprietários dos processos assim como proprietários do ambiente no qual o time atua. Em alguns casos, o

gerente pode ser o próprio *Scrum Master*, entretanto tal abordagem parece ferir a constituição do time. Descrição baseada nos conceitos apresentados por Pries e Quigley (2011).

2.2 KANBAN

Por não ser um termo comum no ocidente, antes de tudo é necessário entender o significado de “Kanban”.

Kanban literalmente significa “Cartão Visual”, “Tabuleta”, “Quadro de Avisos” (PATTON, 2009, p. 3. Tradução nossa).

Segundo o autor acima o termo está associado a cartões visuais, algo que se pode relacionar aos *Post-its* e também seus significados secundários como Tabuletas e Quadros os quais remetem ao *Story Board*, pontos que ligeiramente apontam à comunicação e organização, coincidentemente campo onde as metodologias para desenvolvimento de software se encontram, ainda mais as ágeis por terem um grande foco nos indivíduos como diz o próprio Manifesto Ágil (2001).

Compreendendo o significado do termo é mais fácil entender o contexto onde o mesmo está inserido, porém ainda não clarifica sua origem, tópico abordado em seguida.

“Após a Segunda Guerra Mundial, companhias de manufatura japonesas procuravam por novas maneiras para aumentar a produtividade. No final dos anos 40, Taiichi Ohno estava trabalhando na Toyota e desenvolveu o “Kanban” ou “Quadro de avisos” sistema utilizado para implementar o processo “Just-in-Time” (JIT).

O processo foi desenvolvido após o pessoal da Toyota ter observado a forma como os supermercados empregavam as técnicas para reabastecimento das prateleiras, onde as prateleiras eram estocadas pouco antes o consumidor necessitar de tal item.

O sistema Kanban na Toyota utilizou a taxa de demanda para controlar a taxa de produção. O sistema Kanban foi implementado por toda a Toyota, mas não se tornou popular no ocidente até os anos 70.” (MURRAY, 2012, Tradução nossa).

No trecho acima Murray apresenta uma boa visão histórica da origem do Kanban baseando-se nos ensinamentos passados pelo formulador do Kanban para desenvolvimento para Software, David J. Anderson em suas obras *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results* (Anderson, David. Setembro 2003, Prentice Hall. ISBN 0-13-

142460-2.) e Kanban (Anderson, David. Abril 2010, Blue Hole Press. ISBN 0-9845214-0-2), mas o mesmo esqueceu-se de citar que o Kanban foi criado e desenvolvido não somente por Taiichi Ohno, mas em conjunto com Sakichi Toyoda como pode ser visto no histórico da Toyota (2012).

Fica claro então que o Kanban foi inicialmente criado como parte do Sistema Toyota de Produção (STP) para auxiliar no emprego do *Just-in-Time* (JIT) e posteriormente devido sua eficácia e eficiência foi adaptado por David J. Anderson para o campo de desenvolvimento de software.

Kniberg (2011) explana que no ocidente o Sistema Toyota de Produção é denominado como Lean, que entre seus muitos significados na língua inglesa lhe são atribuídos os adjetivos de magro e esguio segundo o Dicionário Oxford para Aprendizes Avançados (OXFORD ADVANCED LEARNER'S DICTIONARY, 2011), mas a intenção deste termo é se referir à palavra “enxuto”, referência a mentalidade enxuta, filosofia pregada pelo STP onde todos os colaboradores devem se comprometer e trabalhar em conjunto para minimizar todo tipo de perda ou desperdício nos processos envolvidos na produção.

2.2.1 Características básicas e ferramentas.

De acordo com Toombs (2012), todas as descrições do Kanban como uma ferramenta de gerenciamento e melhoria do fluxo de trabalho contém três elementos básicos:

- **Visualizar o fluxo de trabalho:** Uma representação visual do processo lhe permite ver exatamente o estado atual de uma determinada atividade (por exemplo: pronto, em progresso, terminado). O “Quadro de Kanban” é utilizado para tal, possuindo colunas que refletem os estágios de cada etapa do fluxo de produção. Através desta ferramenta o trabalho e seu fluxo tornam-se visíveis e suas atividades e empecilhos óbvios, conceito similar ao *Story Board*.

- **Trabalho em Progresso:** O Kanban limita o “Trabalho em Progresso” (WIP, do inglês *Work in Progress*) através de uma explícita declaração política pelo time pela promoção da qualidade, foco e entrega (por exemplo: o time não aceitará que mais de uma ou duas atividades sejam atribuídas a um membro por vez.). Existe um limite em relação à quantidade de atividades que você consegue exercer simultaneamente e obter um bom resultado. De acordo com relatos dados por Mike Cottmeyer e Dennis Stevens (2012), um processo comumente aplicado para controlar o fluxo de trabalho é o “*One Piece Flow*” (Figura 13) que assim como o Kanban é derivado do STP e criado como uma ferramenta para auxiliar na implementação do JIT, uma filosofia que prega que cada indivíduo só pode executar uma tarefa por vez, mas seu uso não é prescrito pelo método em si.

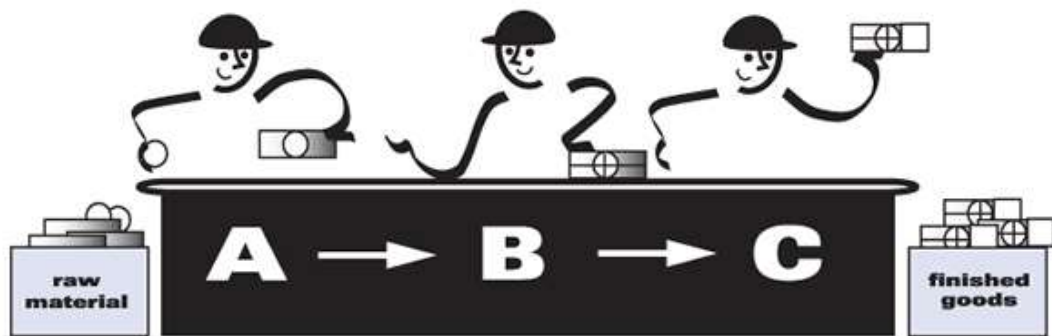


Figura 13 – Exemplo visual do *One Piece Flow* na manufatura (MARCHWINSKI e SHOOK, 2008).

- **Métricas e Melhoria do Fluxo:** O Kanban promove o uso de métricas e acompanhamento do fluxo de trabalho como um alerta em prol da melhoria contínua procurando gerar valor de forma previsível e contínua. Outro relato de Mike Cottmeyer e Dennis Stevens (2012), diz que um tipo de métrica muito utilizado pelos times que adotam o Kanban é o *Lead Time*, métrica utilizada para medir o desempenho do time a partir de cálculos realizados com a quantidade de tempo que uma determinada atividade levou para ser terminada desde o momento que foi adicionada ao projeto, uma amostra das métricas envolvidas pode ser visualizada na figura 14.

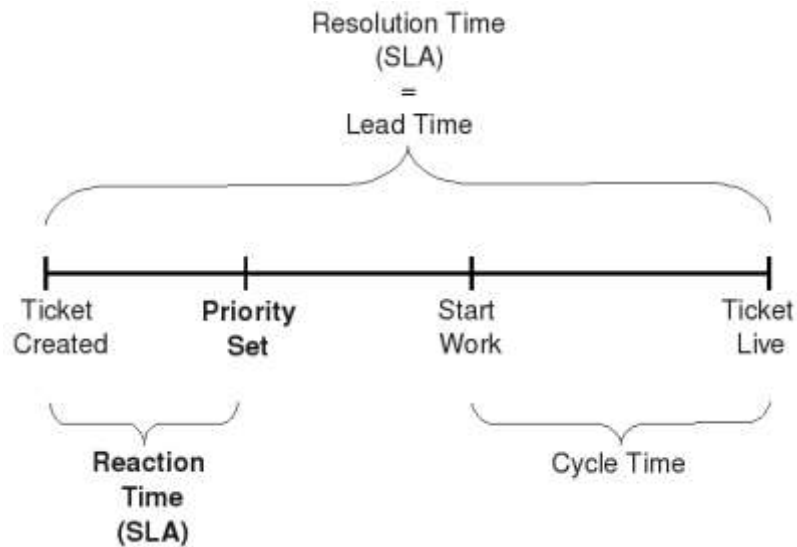


Figura 14 – Amostra dos processos envolvidos na métrica de *Lead Time* (ROOCK, 2010).

Segundo Toombs (2012), tais elementos aparentam serem simples de implementar, mas veteranos práticos do assunto como Mike Cottmeyer e Dennis Stevens (2012) relatam que sua aplicação pode ter efeitos e consequências tremendas mesmo na mais simples das formas:

- Os sistemas de controle visual são valiosos na mudança do comportamento, pois eles expõem a situação num formato de fácil visualização. Isso garante que todos compartilhem do mesmo entendimento quanto à situação atual do trabalho e as restrições no processo. Transparência é a chave para alcançar a mudança organizacional.
- A limitação do trabalho em progresso encoraja todos a trabalharem como um time, prevenindo qualquer indivíduo de ir longe demais que os demais.
- O Kanban aumenta o nível de alerta quanto a impedimentos, forçando o time a interromper as atividades atuais ou executar qualquer trabalho adicional até o impedimento ser resolvido.
- Trabalho adicional é puxado para dentro do sistema quando há capacidade para lidar com tal carga, ao invés de ser empurrado dentro do sistema conforme a demanda.
- Com o Kanban, o foco do gerenciamento e do time torna-se o fluxo do trabalho e não a utilização do time.

- Enquanto o Scrum possui as *Sprint Restrospective* ao final de cada iteração para tratar os itens anteriormente citados, o Kanban explicitamente aponta impedimentos em tempo real e encoraja o time a resolvê-los assim que aparecem.

Segundo Toombs (2012) existem inúmeras formas de aplicar o limite no fluxo de trabalho, exemplos que podem ser destacados são:

- Inserção de filas adicionais entre as colunas iniciais para melhor articular os estados espera associados aos gargalos de produção no projeto e oportunidades de melhoria.
- Uso do conceito de “próxima prioridade” na fila de execução para trazer realidade ao campo de desenvolvimento atentando para itens “indispensáveis” de trabalho.

Algumas situações podem requerer uma prévia discussão e acordo entre os membros do time sobre como tratar defeitos, impedimentos e o limite do trabalho em progresso. Por exemplo, suponha que seja encontrado um defeito em um “site no ar” significando que tal já está acessível por todos os usuários, através de um acordo algumas regras adicionais podem ser adicionadas como: (i) Este tipo de ocorrência tem prioridade máxima. (ii) Este tipo de ocorrência tem permissão para quebrar o limite de WIP. (iii) Este tipo de ocorrência pode pular parte de algum processo. (iv) Pode ser liberado sem aprovação de um supervisor. (v) Tem um limite de entrega automaticamente imediato.

Tais tipos de acordo e planejamento prévio de ação requerem experiência do time tanto com o funcionamento da metodologia quanto dos próprios processos envolvidos no desenvolvimento e suas tecnologias atreladas.

2.2.2 Hierarquia

Ao ser questionado sobre inúmeras literaturas apontando a falta de hierarquia e papéis no Kanban, Anderson (2010) explana que o Kanban possui somente um papel específico prescrito que é informalmente denominado “Treinador” ou “Instrutor” no qual o indivíduo é imbuído a preparar os envolvidos no projeto para executar o Kanban.

De resto de fato não há mais quaisquer papéis ou hierarquia descrita a ser seguida pela metodologia, Anderson (2010) afirma que tal característica é um dos pontos fortes da metodologia, pois como Peter Senge escreveu “Pessoas não resistem a mudança, elas resistem serem mudadas.”, ao implantar um novo método numa organização, a obrigatoriedade de implantar uma nova hierarquia pode causar conflitos e resistência, característica não inerente ao Kanban.

2.2.3 Flexibilidade

Um ponto forte do Kanban é sua flexibilidade e liberdade lhe dando um grau de customização superior aos outros métodos ágeis disponíveis, mas considerando o ponto que os métodos definem regras a serem seguidas, qual a vantagem de um método aberto?

“Um atributo particularmente notável associado ao Kanban é o potencial de aplicar-se a um ciclo de desenvolvimento de software ou metodologia de gerenciamento de projeto já existente. Kanban não pede para que seja feita uma revolução na forma como as pessoas trabalham; Pelo contrário, ele encoraja a mudança gradual. Mudança que é compreendida e aceita através do consenso dos trabalhadores e seus colaboradores.” (TOOMBS, 2012, Tradução nossa).

Toombs (2012) complementa ainda melhor sua colocação dizendo que o Kanban basicamente tem seu foco no gerenciamento do fluxo de trabalho e não tem o foco em mudar de forma alguma a estrutura da organização, tendo seu foco somente na forma como o trabalho se decorre, é a opção ideal quando você deseja melhorar a forma como está fazendo o que quer que seja que está fazendo sem mudanças radicais sendo assim assumindo um risco mínimo aplicando o Kanban como uma melhoria em sua jornada.

2.3 SCRUMBAN

A princípio aventurar-se em utilizar tal método pode ser um choque cultural até mesmo para aqueles que já estão familiarizados com metodologias de gerenciamento para desenvolvimento de software, mas para aqueles que já possuem noção tanto do Scrum quanto do Kanban sua compreensão se torna bem mais fácil.

O Scrumban foi criado por Corey Ladas em 2008 quando lançou sua primeira publicação sobre o assunto, um ano depois, em 12 de janeiro de 2009, lançou seu primeiro livro (*Scrumban – Essays on Kanban Systems for Lean Software*) oficializando a fundação desta nova metodologia de desenvolvimento de software.

Como Meydam (2009) cita em sua obra relatando sua experiência com o método, Scrumban é basicamente a fusão do Scrum com o Kanban com o intuito de aproveitar o melhor dos métodos.

2.3.1 Scrum X Kanban

Apesar de ambos serem métodos ágeis com foco na entrega do produto funcional para o cliente cada vez melhor e mais rápido, são diferentes, vários fatores explicam o motivo de tais diferenças, por exemplo, a própria disposição geográfica de cada um, já que o Scrum veio do Manifesto Ágil que teve sua origem nos Estados Unidos da América enquanto o Kanban veio o STP originário da Toyota no Japão.

Analisando as diferenças entre o Scrum e o Kanban os seguintes itens foram levantados (Toombs, 2012):

Tabela 1 – Diferenças entre o Scrum e Kanban.

Scrum	Kanban
Iterações cronologicamente planejadas prescritas	Iterações opcionais
Usa da “ <i>Sprint Velocity</i> ” como métrica padrão para planejamento e evolução	Usa do “ <i>Lead Time</i> ” como métrica padrão para planejamento e evolução
Times multifuncionais prescritos	Times multifuncionais opcionais, times especialistas é permitido
Atividades devem ser divididas para que então caibam dentro de uma <i>Sprint</i>	Sem prescrição de tamanho
Gráfico de “ <i>Burndown</i> ” prescrito	Nenhum diagrama prescrito
Trabalho em progresso limitado indiretamente (via “ <i>Sprint</i> ”)	Trabalho em progresso limitado diretamente (pelo fluxo dos processos)
Estimativas de atividades prescritas	Estimativas de atividades opcionais
Impossível adicionar atividades durante a iteração	Novas atividades podem ser adicionadas enquanto houver capacidade
O “ <i>Sprint Backlog</i> ” pertence a um único time.	O quadro de Kanban é dividido por múltiplos times e indivíduos.
Prescritos 3 papéis (PO/SM/Time)	Nenhum papel prescrito
O quadro de Scrum reinicia a cada <i>Sprint</i>	O quadro de Kanban é persistente
Prescreve a priorização no “ <i>Sprint Backlog</i> ”	Priorização de atividades é opcional

Fonte: Kanban as a Tool in the Agile Toolbox (TOOMBS, 2012, Tradução nossa).

Vale ressaltar que na Tabela 1, Toombs não considerou a afirmação feita por Anderson (2010) quanto à descrição de papéis e hierarquia conforme o capítulo 2.2.2, o que não é errôneo considerando que Anderson fez esse adendo após ter formulado e oficializado o método, logo Toombs pode ter feito tal afirmação baseado numa literatura anterior a tal declaração do fundador do método.

2.3.2 Scrum e Kanban

O Scrumban vem da união desses dois métodos que apesar de diferentes partilham de similaridades e conceitos em comum, senão dificilmente seria possível fazer uma união tão proveitosa ao ponto ocorrer à oportunidade de criar de um novo método.

Segundo Toombs (2012) a união dos métodos é possível devido as seguintes similaridades:

- Ambos são Lean (STP) e Agile (Manifesto Ágil).
- Utilizam fluxo de produção “puxado” (*pull*).
- Limitam trabalho em progresso (WIP).
- Usam da transparência para direcionar o processo de melhoria.
- Focam em entregar software funcional cada vez mais rápido.
- São baseados em times auto-gerenciáveis.
- Requer a divisão do trabalho em parcelas.
- Tem otimização contínua através de planejamento baseado em dados empíricos (*Sprint Velocity \ Lead Time*).

Segundo Toombs (2012) a limitação explícita do WIP é a diferença chave entre o *Story Board* do Scrum e o *Quando de Kanban*.

Em sua publicação, Corey Ladas (2008), dá sugestões de pontos focais onde é interessante utilizar-se de uma abordagem mais voltada ao Scrum ou mais voltada ao Kanban.

- Uma das técnicas é definir um limite de multitarefas, você pode ter um princípio simples como termine o trabalho atual antes de iniciar o próximo (que seria basicamente o *One Piece Flow*) ou algo mais elabora como uma regra que diz: “tente trabalhar numa única tarefa por vez, mas caso fique bloqueado, então você pode trabalhar num segundo item, mas não mais.”.
- Outra técnica seria associar os responsáveis pelas tarefas em tempo real, ao invés de sair da reunião de planejamento com todas as tarefas já associadas a um responsável limitando e tornando linear o andamento da produção, deixe para que aquele que o time decida no decorrer do fluxo quem deve

assumir cada tarefa. É uma boa maneira inclusive de fomentar a multifuncionalidade o aprendizado entre os integrantes do time.

- Outra melhoria a ser aplicada é utilizar o conceito do *Story Board* do Scrum, mas entre a coluna do *Sprint Backlog* e a coluna de *WIP* adicionar o estágio de “*Ready*” ou “Preparado”. A coluna de “Preparado” vai conter itens que assim como resto do *Sprint Backlog* estão pendentes, mas por ter alta prioridade devem ser os próximos a serem executados, como nenhuma tarefa ainda foi associada a ninguém, o primeiro integrante que ficar disponível irá assumi-la ao invés de pegar um outro item qualquer do *Sprint Backlog*, isso permite desassociar o processo de priorização das tarefas com o de atribuição das mesmas simplificando-os. A fila de “Preparado” deverá ter um limite e o mesmo deverá ser pequeno já que seu único propósito é indicar qual tarefa deve ser a próxima a ser executada.

De acordo com Toombs (2012), esta é, sem dúvidas, uma maneira sofisticada de empregar as ferramentas e que tal orquestração e resultados são frutos de uma clara evolução ao invés de simples mandamentos.

2.3.3 Implantação

De acordo com Toombs (2012), várias tentativas de implantação do Scrumban sofreram resistência, foram terrivelmente sofridas ou até mesmo abandonadas.

Questão já discutida previamente por Anderson (2010) quanto à natureza humana de resistir ao fato de sofrer uma mudança radical interna, quando a mesma pode ocorrer de forma gradual pela mudança do ambiente.

Então considerações devem ser feitas ao implantar o Scrumban, primeiramente deve-se implantar o Scrum caso o mesmo já não esteja presente e então utilizar a habilidade do Kanban de ser implantado gradualmente a fim de unir as duas ferramentas até de fato estar consumado o Scrumban.

Para a inserção bem sucedida do Kanban, Anderson (2010) definiu três **princípios fundacionais**, cinco **propriedades de núcleo** e doze **passos a serem seguidos**.

Princípios fundacionais:

- Comece com o que você tem agora.
- Concorde em seguir de maneira incremental a mudança evolucionária.
- Respeite os processos, papéis, responsabilidade e títulos atuais.

Propriedades de núcleo:

- Visualize o fluxo de trabalho.
- Limite o trabalho em progresso.
- Administre o fluxo.
- Torne as políticas de processos explícitas.
- Aperfeiçoe de forma colaborativa (usando modelos e metodologia científica).

Passos a serem seguidos:

- Concorde com uma série de objetivos para introduzir o Kanban.
- Mapeie o traçado do valor.
- Defina algum ponto onde você irá controlar a entrada. Defina a demanda e seus respectivos *Stakeholders*.
- Defina uma série de tipos de trabalho baseado nos pedidos oriundos dos *Stakeholders* da demanda.
- Análise a demanda para cada tipo de pedido.
- Reúna-se com os *Stakeholders* da oferta e da demanda.
- Crie um mural de quadro/cartões para monitorar o traçado do valor que estiver controlando.
- Opcionalmente, crie um sistema eletrônico para desempenhar a mesma função que a acima citada.
- Acorde com o time de ter uma reunião de pé frente ao mural; convide os *Stakeholders* (presença não obrigatória).
- Acorde de haverem regularmente operações de reuniões de revisão/apresentação e análises retrospectivas dos processos; convide os *Stakeholders* (presença não obrigatória).
- Eduque o time quanto ao novo mural, WIP e o sistema de demanda “puxada”. Nada mais em seus mundos deve ter sido mudado.

2.3.3.1 Ciclo Produtivo

Uma dúvida provavelmente ainda paira no ar, deve ter ficado claro que para uma implantação bem sucedida do Scrumban é necessário primeiro fazer presente o Scrum e em seguida gradualmente aplicar o Kanban para auxiliar no gerenciamento do fluxo da produção, mas o Scrum trabalha com iterações enquanto o Kanban é contínuo, o que fazer? Singleton (2012) explica:

“Estes simples gráficos mostram como nós podemos unir o Scrum (com suas entregas periódicas) e o Kanban (com entregas muito frequentes) e combina-los no “Scrumban”, processo o qual possui entregas periódicas e é um aperfeiçoamento do processo normal do Scrum. Ideal para times distribuídos e próximo da forma real como muitos deles trabalham agora.” (SINGLETON, 2012, Tradução nossa).

Vejamos então quais as definições que Singleton (2012) tem a apresentar.

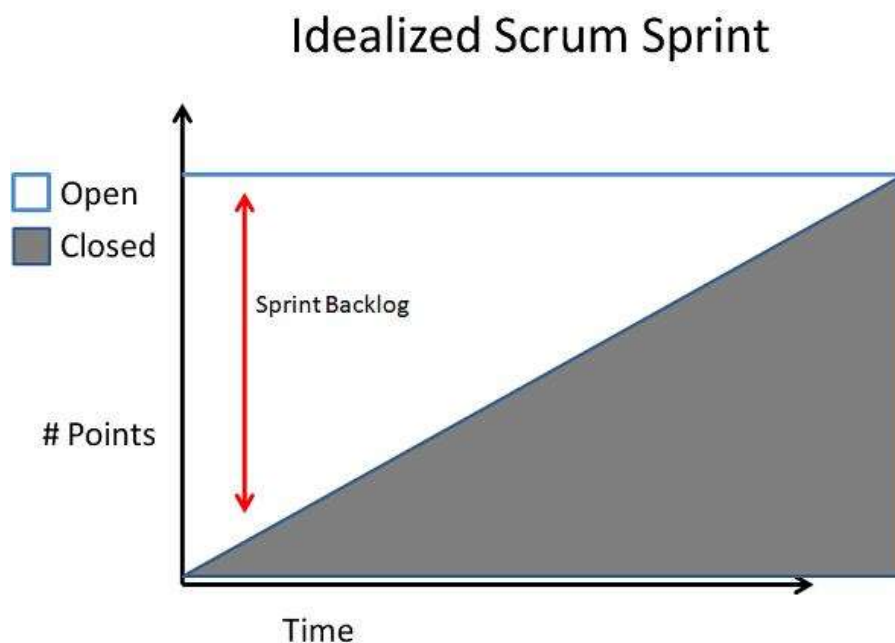


Figura 15 – Ciclo ideal de uma Sprint segundo o Scrum (SINGLETON, 2012).

Como se pode observar na figura 15, tem-se um exemplo de um ciclo de *Sprint* ideal, em branco temos as tarefas pendentes e em cinza as realizadas, no eixo vertical as tarefas (medidas em *Story Points*) e no eixo horizontal a linha do tempo, veja que no ponto zero temos o *Sprint Backlog* em sua totalidade antes do início das atividades.

Segundo Singleton (2012) você raramente verá um projeto seguir essa linha perfeita de realização de tarefas, geralmente algumas tarefas são adicionadas no início e outras têm de ser removidas no final, podem ocorrer vários gargalos quando se chega perto do momento da entrega.

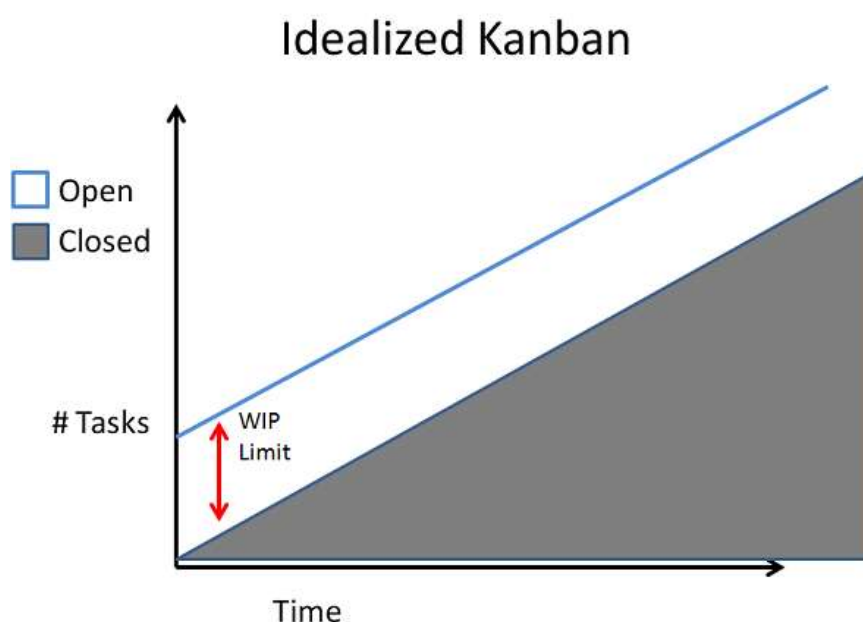


Figura 16 – Fluxo ideal de trabalho segundo o Kanban (SINGLETON, 2012).

Na figura 16 vemos que há diferença na métrica das tarefas pendentes que são consideradas pela quantidade e não *Story Points* isso segundo Singleton (2012) ocorre porque como o Kanban usa a filosofia Lean a ideia é não ter tarefas por aí esperando alguém trabalhar nelas, o Kanban limita a quantidade de tarefas independentemente do seu tamanho, assim cada tarefa atual terá alguém trabalhando nela. Uma tarefa que requer um dia para ser realizada e uma tarefa que requer nove dias para ser realizada vão requerer a mesma quantidade de atenção, em contraste com o Scrum onde então a tarefa seria fragmentada em nove tarefas

de um dia sequenciais para se adequar ao sistemas. O limite de *WIP* do Kanban é uma maneira simples e efetiva de garantir que cada tarefa será concluída o mais rápido possível, não requerendo estimativas e providenciando uma linha natural de trabalho para tarefas que pudessem vir a durar mais de uma iteração.

Singleton (2012) afirma que os processos do Kanban têm várias vantagens e são ótimos caso você trabalhe com serviços de atualização *online* e tenha entregas diárias. Entretanto a maioria dos times não consegue fazer entregas diárias. Eles enviam seus produtos para serviços de distribuição *online* mensalmente ou tem clientes corporativos que aceitam atualizações quinzenais ou que gravam mídias físicas anualmente. Estes times necessitam de lançamentos periódicos estáveis. Para tais times, é sugerido o processo Scrumban.

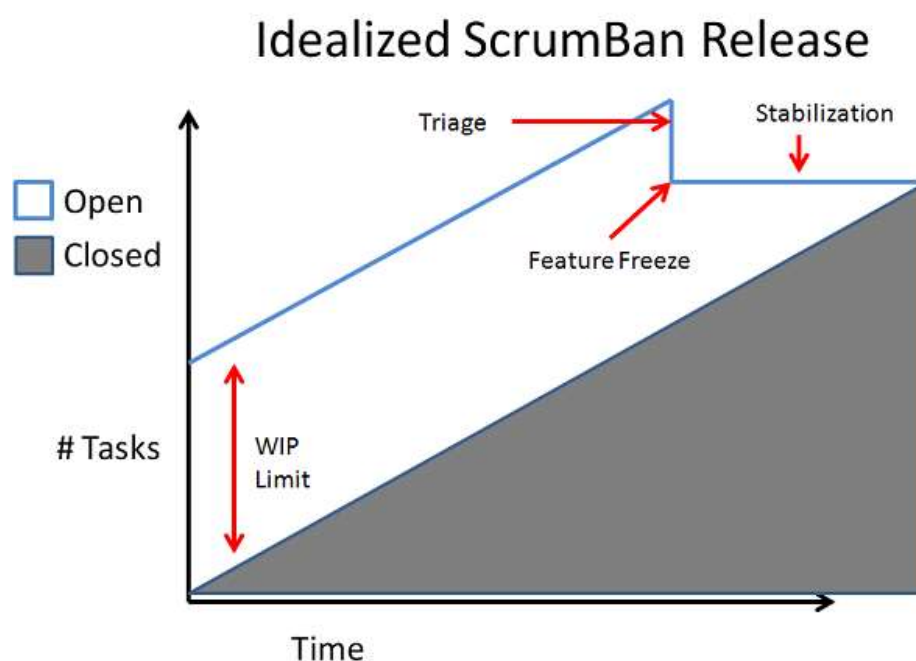


Figura 17 – Ciclo ideal de iteração segundo o Scrumban (SINGLETON, 2012).

Singleton (2012) explica os conceitos relacionados ao ciclo incremental, iterativo e evolucionário do Scrumban. Conforme a figura 17 que são:

- **Triage:** No processo de **Triagem** devem-se identificar as tarefas que não serão completas até o fim da iteração e retirar-las (retorná-las do *Product*

Backlog). Isto é essencial para que o time consiga entregar uma parcela completo de produto ao final da iteração.

- **Feature Freeze:** No período de **Congelamento de Funcionalidades** deve-se parar de adicionar tarefas para a lista de trabalho atual. Então você pode estabilizar a entrega realizando o que restar da lista de tarefas e problemas incompletos.
- **Stabilization:** Por fim vem a **Estabilização** momento em que as tarefas são finalizadas ou removidas e problemas são resolvidos até que se tenha em mãos uma versão estável para entrega. Isso caracteriza o fim o iteração.

Caso seu time consiga aplicar este tipo de iteração realizando o processo de entrega da Triagem até a Estabilização num curto espaço de tempo e sem empecilhos você pode usufruir de duas vantagens imediatas:

- 1) Você não precisara estar realizando estimativas e fazer mais sessões de *Sprint Planning* para as próximas *Sprints*, sendo assim acaba por ganhar um dia extra de desenvolvimento.
- 2) Você pode adicionar a correção de erros a qualquer momento. No processo de Scrum ideal é desagradável a forma como os erros precisam ser separados numa fila diferenciada de trabalho.

Noronen (2011) num relato otimista e detalhado de como foi sua experiência pessoal como *Product Owner* em um projeto que utilizava o Scrum e resolveu migrar para o Scrumban (O ciclo de mudança mais natural e recomendado) revela informações importantes que complementam as considerações finais a serem feitas quanto ao Scrumban que são:

Daily Meetings continuam ocorrendo de forma natural como é pregado no próprio Scrum.

Sprint Planning mantém-se ativa, mas agora com as mudanças feitas pelo ponto de vista do Kanban, as estimativas quanto às atividades a serem desenvolvidas não são mais obrigatórias a partir do ponto que todas deverão ser feitas o mais rápido possível, a reunião acaba tomando um caráter mais de visualização total do trabalho a ser desenvolvido e definição inicial da prioridade de execução das tarefas lembrando que as mesmas podem mudar no decorrer da interação.

Sprint Review como o fluxo de produção do Kanban força o *Product Owner* a estar em constante contato com o *Scrum Master* acompanhando e avaliando o

andamento das tarefas, tais eventos só mantem seu padrão se outras pessoas envolvidas como *stakeholders* e usuários desejarem uma apresentação formal do que foi desenvolvido na *Sprint*, senão é somente um momento onde o *Product Owner* realizará o processo de aprovação do que foi desenvolvido no determinado período de tempo daquela interação, sem a obrigatoriedade do time lhe apresentar o que foi feito.

Sprint Retrospective têm se a opção de continuar com as retrospectivas acontecendo conforme prega o Scrum após cada iteração ou se for de desejo do time em um período maior de tempo, é dada a possibilidade de realizá-las durante o processo de desenvolvimento junto com as *Daily Meetings* ou até mesmo se o time julgar que tal processo devido a sua forma de trabalho não ajuda na melhoria e evolução do time também se tem a liberdade de não realizar mais as retrospectivas.

Noronen (2011) então anuncia que obteve os seguintes resultados ao aplicar o Scrumban durante dois meses:

- A moral do time é visivelmente melhor que antes, o time realmente gostou da forma de trabalho.
- A velocidade do time manteve-se quase a mesma durante as *Sprints*.
- Os desenvolvedores sentem que eles podem de fato terminar aquilo que estão fazendo e encontrar melhores soluções do que antes.
- O analista de qualidade acredita que tem realizado seu trabalho com maior qualidade.
- Houveram dificuldades quanto a transferência de informação entre membros do time em diferentes fases (por exemplo de desenvolvedor para teste), o que foi resolvido adicionando demonstrações internas.
- O time em geral parece mais disposto a melhorar e encontrar melhores caminhos para fazer processos que lhe melhor sirvam.

De acordo com Noronen (2011) as seguintes lições aprendidas foram aprendidas:

- O Scrumban parece ser mais apropriado para ambientes onde os membros do time têm afazeres no projeto e fora dele. Um bom exemplo é o desenvolvimento de um produto onde há manutenção e trabalho de suporte adicional ao desenvolvimento natural.
- Scrumban parece ser muito mais adequado para times que não podem ter múltiplas funcionalidades entre todos os papéis. Por exemplo, o fato de poder

alocar um analista de qualidade para escrever código. Sendo assim sua aplicação seria ideal em ambientes onde a diferença entre as habilidades técnicas é grande, como desenvolvimento de jogos.

- Scrumban requer mais dedicação por parte do *Product Owner*. Sua atividade no projeto deve ser muito maior, maior ao ponto de dispensar a necessidade de *Sprint reviews* formais.

3. **SERIOUS GAMES**

Antes de adentrarmos o vasto universo dos *serious games*, precisamos de fato entender do que tal termo se trata, o que de início já é um desafio, numa simples busca no Google pelo termo “*serious games*” 152,000,000 resultados foram encontrados [06-11-2012], inúmeros autores e proeminentes pesquisadores da área tem suas próprias definições para o termo, não há um consenso literário sobre o mesmo, afortunadamente existe um grupo dedicado ao estudo e promoção dos *serious games* intitulado **Serious Games Initiative** que dá a seguinte sucinta descrição para o termo: Jogos voltados para outros propósitos além do entretenimento (SERIOUS GAMES INITIATIVE, 2002).

Mas qual a diferença entre um jogo voltado para o entretenimento e um *serious game*? Podemos ver os fatores que segregam os *serious games* dos jogos voltados ao setor do entretenimento na tabela abaixo.

Tabela 2 - Diferenças entre jogos para entretenimento e serious games.

	Serious Games	Jogos para Entretenimento
Tarefa X riqueza na experiência	Foco na solução de problemas.	Preferência por experiências ricas.
Foco	Importância dos elementos no aprendizado.	Divertir-se.
Simulações	Suposições são necessárias para simulações funcionais.	Processos de simulação simplificados.
Comunicação	Deve refletir uma comunicação natural (exemplo: não perfeita).	Comunicação é geralmente perfeita.

Fonte: *Serious Games – An Overview* (SUSI, JOHANNESSON e BACKLUND 2007, Tradução nossa).

Os dados acima tem sido alvo de várias discussões dentro da comunidade de *serious games* sendo assim não se recomenda adota-los como valores absolutos, mas servem de base para um melhor entendimento inicial do assunto.

E como seria o desenvolvimento de tal tipo de jogo? Podemos ter uma noção segundo o estudo realizado por Alana Marques de Moraes apresentado a seguir:

“No processo de elaboração dos *serious game* deve existir planejamento pedagógico subsidiando o desenvolvimento, além da delimitação de características como roteiro, conceituação artística (*game design*), jogabilidade (*gameplay*) e definição da interface (MACHADO *et al.*, 2009), importantes na construção do *Game Bible* do projeto. A etapa de planejamento dos jogos é um processo complexo, pois envolve várias ciências trabalhando de forma colaborativa para alcançar um objetivo em comum (WONG *et al.*, 2007). Entretanto, ao se falar dos *serious games* esta complexidade aumenta e surge um apoio extra de profissionais da área a qual o conteúdo do jogo se relaciona (BECKER; PARKER, 2007). A especificidade dos *serious games* demanda um envolvimento de profissionais de áreas distintas, além da Computação e Artes (ZYDA, 2005). Esta parte da equipe norteia os conteúdos que devem ser explorados e as estratégias de apresentação dos mesmos para se atingir o intuito proposto pelo jogo. Os *serious games* permeiam por diversas áreas do conhecimento.” (MORAIS, 2011, p. 23).

É notável o tamanho do esforço e as múltiplas frentes de conhecimento que se fazem necessárias para a elaboração de tal tipo de trabalho, frente à tamanha complexidade é comum haverem questionamentos sobre a pertinência da existência de jogos senão para o entretenimento. De acordo com Susi, Johannesson e Backlund (2007) através dos *serious games* é possível transmitir experiências e realizar interações práticas de forma que não seria possível na vida real ou que não seria possível de se reproduzir sem danos e efeitos colaterais negativos.

Segundo uma matéria da Business Wire (2012) com Robert Brown representante da Serious Games Association (SGA) estima-se que a indústria de *serious games* tem uma receita de 2 a 10 bilhões de dólares, dependendo de quanto do mercado total for abrangido de acordo com o levantamento feito por Sue Bohle, diretora executiva na SGA.

De acordo com autores da LUDUS (2011) comunidade dedicada ao estudo de *serious games*, os mesmos são conhecidos através de outros termos como: jogos educacionais, simulação, realidade virtual, jogos de propósito alternativo, entretenimento educativo, aprendizado baseado em jogos digitais, simulações imersivas para aprendizado, jogos de impacto social, jogos persuasivos, ambientes de aprendizado sintético, *games for change*, *games for good*, *game-based “X”*.

Através dos estudos realizados por Susi, Johannesson e Backlund (2007) os *serious games* são aplicados em várias áreas como educação, idealismo, política, corporativismo, ciência, militar, religião, arte e em particular a área saúde a qual envolve o escopo deste trabalho onde nos aprofundaremos no capítulo a seguir.

3.1 SERIOUS GAMES PARA SAÚDE

Segundo Susi, Johannesson e Backlund (2007) a aplicação de *serious games* na área da saúde tem se tornado cada vez mais comum e hoje existe uma vasta quantidade de aplicações.

O uso dos *serious games* é efetivo, pois através de hardwares específicos você pode tratar não somente de questões psíquicas e psicológicas, mas também de forma física o usuário final, seja ele um paciente em reabilitação ou um futuro cirurgião em treinamento. Rego, Moreira e Reis (2011) relatam uma série de benefícios atrelados ao uso de *serious games* na área da saúde exemplificando:

- **Os jogos estão se tornando mais motivadores para os pacientes que as terapias tradicionais** devido ao fato que os jogos geralmente possuem uma história ou outra forma de contextualização além de uma série de desafios que motivam os pacientes a realizar determinados objetivos através de tarefas que são planejadas para a reabilitação. O paciente fica imerso em diferentes situações onde ele tenta cumprir objetivos propostos pelo jogo e assim é distraído quanto as suas incapacidades e ao fato que de é uma atividade de reabilitação. Além do cumprimento do escopo que é a recuperação do paciente o jogo proporciona imersão, desafio, motivação, diversão, sensações as quais ele não sentiria seguindo uma sequencia repetitiva de tarefas para parte de seu plano de reabilitação.
- **Jogos estão se tornando mais acessíveis para a população geral.** Os sistemas computadorizados estão cada vez mais disseminados e acessíveis aos usuários em geral, de forma que alguns dispositivos como videogames (consoles), computadores portáteis, televisores, etc... Ao mesmo tempo em que as pessoas tem se familiarizado cada vez mais quanto a sistemas de

informação e tecnologias computacionais o que promove a acessibilidade aos jogos de computador.

- **O surgimento de novas interfaces está mudando os paradigmas da interação do usuário** com as aplicações, além dos tradicionais *mouse* e teclado, novas formas de entrada de dados como gestos, resposta de força, resposta de balanço, reconhecimento de expressões faciais e reconhecimento de voz, isso considerando só as mais relevantes das novas modalidades de entrada de dados disponíveis. Com estas novas tecnologias é possível à criação de aplicações que explorem de forma muito mais natural e livre a cinética humana a partir do ponto que corpos artificiais (controles, manches, etc..) para a interação homem-máquina aumentado à qualidade e eficiência.
- **O avanço da internet permite o tratamento e acompanhamento a distancia e sociabilização**, as pessoas cada vez mais têm acesso à internet e cada vez de melhor qualidade o que possibilita, por exemplo, tratamento de reabilitação onde o médico a distância acompanha os resultados do paciente e adapta seu plano de exercícios de casa eliminando a necessidade de locomoção de ambas as partes para o hospital ou clinica gerando enorme conforto e economia. Adicionalmente o fato dos pacientes estarem conectados na rede de computadores permite a introdução de componentes sociais que podem ser explorados para a criação de cenários onde os pacientes podem interagir de forma cooperativa ou competitiva à distância.

Estes relatos dados por Rego, Moreira e Reis (2011) são em grande maioria voltados para o campo específico de reabilitação de pacientes, mas serve de base para compreenderem-se os inúmeros benefícios que tais tipos de aplicações podem trazer.

Tratando então sobre as áreas que podem ser abrangidas pelos *serious games* quando aplicados à saúde Susi, Johannesson e Backlund (2007) listam oito campos sendo estes: aptidão física, educação na saúde / autocuidado, terapia de distração, recuperação e reabilitação, treinamento e simulação, diagnóstico e tratamento de doenças e condições mentais, funcionamento cognitivo e controle mental e emocional.

Alguns desses campos acabam tendo tratativas e escopos similares. Baseando-se em estudos realizados por Machado, Moraes e Nunes (2009), podemos resumir estes oito campos em quatro categorias principais:

1. Auxiliares de terapia
2. Promoção da saúde e condicionamento físico
3. Monitoramento da saúde
4. Treinamento

A seguir nos aprofundaremos pouco mais em cada uma dessas categorias principais e veremos exemplos de *serious games* relacionados às tais.

3.1.1 Auxiliares de terapia

Os jogos auxiliares de terapias são voltados a pessoas em processo de reabilitação e tratamentos físicos ou psicológicos (MACHADO, MORAES e NUNES, 2009).

A tecnologia de Realidade Virtual Aumentada (RVA) tem grande aplicabilidade nesse campo, aumentando de forma significativa o nível de imersão do paciente no tratamento, a The Virtual Reality Medical Center (VRMC) clínica norte americana especializada no uso de realidade virtual para tratamento de fobias e distúrbios mentais (Figuras 18 e 19) aponta as seguintes vantagens ao utilizar-se de um tratamento a base de RVA:

- Custo consideravelmente menor do que exposição em ambiente real.
- Total controle da experiência “fóbica”, sem sair das dependências da clínica.
- Segmentos de qualquer fobia podem ser repetidos, permitindo o paciente gradualmente reduzir seu medo ou ansiedade.
- Elimina os riscos potenciais de dano físico.
- A terapia é conduzida dentro da clínica oferecendo ao paciente confidencialidade eliminando preocupações com constrangimento público
- Pessoas que já demonstraram resistência a métodos tradicionais de tratamento podem considerar a Terapia de Exposição à Realidade Virtual.



Figura 18 – Paciente durante tratamento de fobia relacionada à altura (VRMC, 2011).



Figura 19 – Visual dentro do jogo da paciente sob tratamento da fobia de altura (VRMC, 2011).

Não somente a RVA têm sido utilizado em tal tipo de tratamento, dispositivos móveis também servem de plataforma para *serious games* como o Fear of Spiders Treatment HD (Figura 20) disponível na Apple Store para iPad [07-11-2012], *serious game* com o escopo de auxiliar no tratamento da Aracnofobia (medo de aracnídeos).

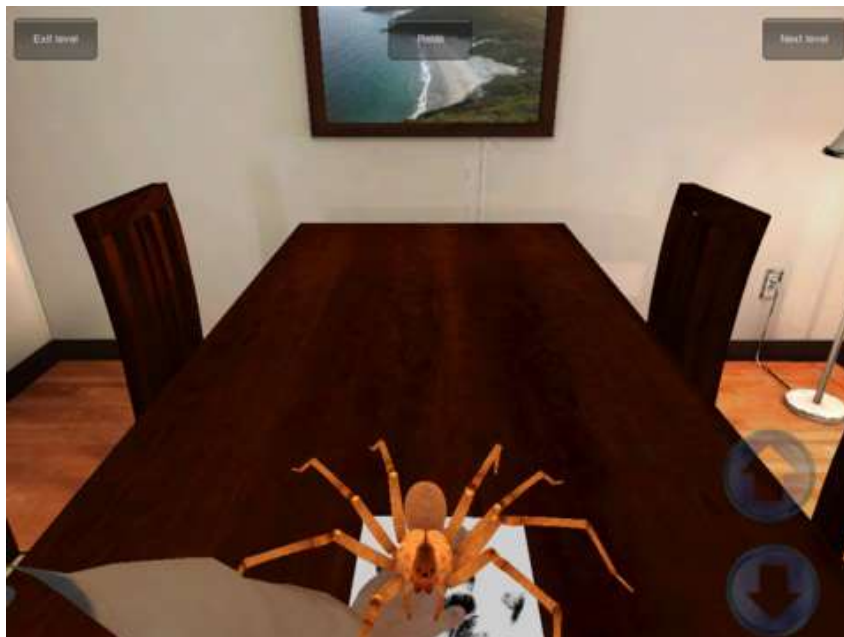


Figura 20 – Imagem do jogo Fear of Spiders Treatment HD disponível para iPad (iTUNES, 2011).

Tais aplicações também estão disponíveis para consoles e computadores, o simulador de guerra Full Spectrum Warrior (Figura 21) disponível para as plataformas Playstation 2 e PC, é uma referência de *serious game* para treinamento de estratégias militares pelo exército norte americano, segundo Susi, Johannesson e Backlund (2007) o mesmo é utilizado para diagnosticar e tratar veteranos de guerra com Transtorno de Estresse Pós-traumático (TEPT).



Figura 21 – Full Spectrum Warrior, utilizado na diagnose e tratamento do TEPT em veteranos de guerra do exército norte americano (MADGAMER, 2012).

3.1.2 Promoção da saúde e condicionamento físico

Esta categoria de *serious game* para saúde tem como escopo conscientizar e auxiliar as pessoas a cuidarem de sua própria saúde em alguns casos o aprimoramento de seu condicionamento físico, é uma das mais amplamente distribuídas entre a população em geral devido à procura e acessibilidade atual dos consoles.

Nesta categoria encaixam-se os famosos *Exergames* (Jogos voltados para exercícios físicos) que de acordo com Johnson (2008) tiveram seu início em 1982 com o periférico The Joyboard (Figura 22) produzido pela Amiga para o console Atari, um controlador em forma de plataforma que através de quatro sensores digitais analisa o balanço e equilíbrio do jogador quando sobre a plataforma, na época somente um jogo foi feito para tal periférico o “Mogul Maniac”, um jogo que simulava a experiência de esqui num circuito de slalom.



Figura 22 – The Joyboard primeiro controlador voltado para *Exergames* (JOHNSON, 2008).

Lazarus (2010) fez um levantamento histórico comentado sobre os *Exergames* o qual é apresentado abaixo (Tradução nossa):

- 1982 – Amiga lança o **The Joyboard**, um periférico para um jogo de esqui.
- 1982 – Atari conceitualiza o projeto **Puffer**, uma bicicleta de exercício controlado para os sistemas 400/800 ou 5200 (abandonado e nunca lançado).
- 1986 – RacerMate lança o **CompuTrainer** para o console NES, uma ferramenta motivacional para exercícios onde o usuário pedala por estradas (muito caro).
- 1988 – Namco lança o **Family Trainer Power Pad** para o console NES, um tapete controlador completo com botões gigantes.
- 1989 – Mattel e Nintendo lançam o **Power Glove** para o console NES, um controle em formato de luva robótica (usuários relataram ser relativamente inoperável).
- 1989 – Namco lança o **Dance Aerobics**, um jogo de aeróbica para o Power Pad e NES.
- 1992 – CyberGear Inc lança o **Tectrix VR Bike and Climber**, uma bicicleta e escada deslizante orientada por software (difícil de manter e dominar).
- 1995 – Nintendo e Life Fitness lançam o **Extertainment System**, uma bicicleta ergométrica com correspondentes jogos na tela (U\$3,500 etiqueta de preço).
- 1996 – Namco lança o **Prop Cycle**, um jogo de bicicleta ergométrica.

- 1998 – Konami lança o **Dance Dance Revolution** (e a era dos *Exergaming* de fato começa).
- 2001 – A companhia britânica Exertris desata da Microsoft para lançar o **Exertris Interactive Exercise Bike**, um jogo de ciclo de treino interativo para Windows CE e XP.
- 2003 – Cat Eye Fitness lança o **GameBike**, uma bicicleta de corrida para PlayStation 1 e 2.
- 2004 – Respondesign lança o **Yourself! Fitness**, um treinador virtual para PlayStation 2 e Xbox.
- 2005 – **Bodypad** é lançado, um pacote de controles para joelhos, cintura, cotovelos e mãos para o PlayStation.
- 2007 – **Gamercise** lança o *power stepper* e pedais periféricos, um sistema de alimentação elétrica que requer a movimentação dos pés por parte do jogador para alimentar consoles.
- 2007 – Fischer-Price lança o **Smart Cycle**, um sistema de bicicleta ergométrica eletrônica miniaturizada para crianças de 3 a 6 anos.
- 2008 – Nintendo lança o **Wii Fit**, um tapete para treino personalizado para o Wii
- 2009 – Electronic Arts lança o **EA Sports Active**, um periférico de malhação avançado para o Wii
- 2010 – Sony lança o **Move**, um controlador de movimento para o PlayStation 3.
- 2010 – Microsoft lança o **Kinect**, um sistema de controle de movimento para o Xbox 360.

Essa categoria de *serious game* não se limita ao condicionamento físico da pessoa, também há jogos voltados para a educação e promoção de hábitos e comportamentos saudáveis, tais jogos são denominados *MindHabits* (MINDHABITS, 2011).

Um exemplo conhecido de *MindHabit* é o *The Incredible Adventures of the Amazing Food Detective* (Figura 23) conjunto de *serious games* da Kaiser Permanente® que tem como objetivo a educação alimentar



Figura 23 – Imagem do jogo Snacktown Snackdown da série The Incredible Adventures of the Amazing Food Detective (KAISER PERMANENTE®, 2012).

3.1.3 Monitoramento

De acordo com os estudos realizados por Machado, Moraes e Nunes (2009), o monitoramento da saúde vem ganhando destaque a partir do advento de aparatos tecnológicos como marcadores biológicos. Tais marcadores (sensores) são utilizados para observar o aumento da atividade cerebral nas pessoas quando expostas aos *serious games*. Este campo tem um enfoque maior na análise do efeito dos jogos sobre as pessoas atuando na terapia da distração e funcionamento cognitivo.

Os *serious games* aplicados na terapia da distração funcionam como ferramentas para ajudar pacientes a suportar a dor ou reduzir seu medo e ansiedade antes de intervenções médicas, um exemplo é o jogo Free Dive (Figura 24) da BreakAway (2012). Em seu vídeo promocional os desenvolvedores envolvidos no jogo junto a profissionais da área da saúde afirmam através de estudos por eles realizados, que as crianças ficam em média 60% mais tolerantes a dor enquanto jogam o Free Dive.



Figura 24 – Imagem do jogo Free Dive utilizado na terapia da distração (BREAKAWAY, 2012).

Tanto em formato digital quanto tradicional (por exemplo, cartas e tabuleiros) os *serious games* voltados ao funcionamento cognitivo tem como escopo treinar a memória do usuário e desenvolver habilidades analíticas e estratégicas segundo estudos realizados por Susi, Johannesson e Backlund (2007).

A Lumosity (2012), que é uma parceria entre várias instituições diferentes é um exemplo de produtora de tais tipos *de serious games* oferecendo mais de 35 jogos e exercícios voltados para o desenvolvimento cognitivo da pessoa.

3.1.4 Treinamento

Considerando os estudos feitos por Susi, Johannesson e Backlund (2007), nesta categoria os *serious games* são utilizados como ferramentas de treinamento e simulação podendo ter diversas aplicações sendo comumente mais aplicados na capacitação e treinamento de futuros profissionais da área da saúde com o objetivo de fornecer treinamento realista para aumentar a difusão de procedimentos inovadores e menos invasivos enquanto diminui a curva de aprendizado do futuro profissional, não se limitando ao tratamento de humanos como, por exemplo, no jogo

Pet Pals: Animal Doctor (Figura 25) da Legacy Interactive (2009) onde o jogador toma o papel de um veterinário realizando diagnose e tratamento de diversos tipos de animais.



Figura 25 – Pet Pals: Animal doctor simulador de veterinária (LEGACY INTERACTIVE, 2009).

Através dos relatos de Leggett (2009) é notável que assim como nos *serious games* auxiliares de terapia e outros tipos de simuladores fora da área da saúde, a RVA é amplamente utilizada para simulações e treinamentos (Figura 26). Devido às limitações existentes no treinamento de procedimentos médicos, o uso de aplicações desta natureza é capaz de prover meios efetivos de treinamento por meio da reprodução de situações reais livre de qualquer risco ou desperdício de recursos.



Figura 26 – RVA no treinamento e simulação de intervenções cirúrgicas (LEGGETT, 2009).

3.1.5 Métodos ágeis e *serious games*

Um *serious game* em formato digital possui elementos que podem caracterizá-lo como um software, logo, métodos de desenvolvimento de software também são aplicáveis ao mesmo. Sendo assim podemos estar aplicando métodos ágeis no desenvolvimento de *serious games*. É comum encontrarmos obras literárias falando sobre o assunto, mas provavelmente devido a restrições comerciais e legais é de certa forma escassa a quantidade de material aberto ao público falando de jogos que se utilizaram de métodos ágeis em seu desenvolvimento, principalmente quando o mesmo tem fins lucrativos.

Dentre os poucos exemplos encontrados, o mais completo e informativo relato a respeito da aplicação de um método ágil no desenvolvimento de um *serious game* encontrado é uma publicação feita por membros de um projeto denominado “TIE Project” na Universidade de Salerno, Itália.

Segundo Bulfico, Francese, Lettieri et al. (2011) o “TIE Project” fundado pela Regione Campania com o escopo de desenvolver atividades *on-line* relacionadas à educação ambiental nas escolas primárias e o *serious game* que foi desenvolvido por uma equipe envolvida neste projeto denominado “Pappi World”.

Um relato interessante foi quanto à regra de negócio estabelecida sobre os pontos que deveriam ser abordados pelo projeto, envolvendo a base do programa educacional na hierarquia de resíduos que são os seguintes tópicos:

- Prevenção, incluindo ações pela redução da quantidade de resíduos gerados;
- Recuperação de recursos, incluindo reuso, reprocessamento, reciclagem e recuperação de energia;
- Descarte, tratando todas as opções da forma mais ambientalmente responsável.

Em particular o projeto era constituído por duas frentes de desenvolvimento, sendo estas:

1 - Palestras multimídia sobre todos os três tópicos e os testes automatizados avaliação relacionados.

2 – Um *serious game* on-line em realidade virtual com o objetivo disseminar esses conceitos, envolvendo os alunos no aprendizado através de atividades práticas.

Os autores da publicação foram imbuídos de desenvolver a segunda tarefa a qual foi descrita e detalhada na mesma.

Pappi World (Figura 27) é um mundo virtual 3D em RVA com múltiplos níveis de interação que incorpora elementos de *Real Time Strategy* (Estratégia em Tempo Real) e tem como escopo educar crianças principalmente na faixa etária de 8 a 10 anos das escolas primárias sobre recolhimento de resíduos e reciclagem de lixo. Devido o time ser multifuncional e o projeto ter alto grau de incerteza quanto os requisitos foi empregado o uso do Scrum desde o pré-jogo nas seções de planejamento envolvendo os conceitos anteriormente vistos como *Product Backlog* e *Sprint Planning*, passando pelo desenvolvimento e suas iterações (*Sprints*) até sua conclusão no pós-jogo, que segundo os autores não é definitiva, pois há interesse em ampliar o jogo para também educar as crianças quanto à economia de energia elétrica e fazer evoluções baseadas no retorno dado pelos usuários sobre o jogo.

As funcionalidades nas quais a jogabilidade se baseia são divididas por cinco níveis sendo cada nível uma camada que envolve o nível inferior do jogo, estes níveis são respectivamente o Pappi World, o arquipélago (cada um representara uma escola), a ilha (representando as classes), o vilarejo (representa a terra do aluno) e por fim os Mini Jogos.

As atividades executadas pelos jogadores se encontram nos níveis do vilarejo e dos Mini Jogos, tendo modalidades de caráter interativo e colaborativo.



Figura 27 – Cena do *serious game* voltado à educação ambiental, Pappi World (BULFICO, FRANCESE, LETTIERI et al, 2011).

O Jogo foi desenvolvido no ambiente NetBeans na plataforma Java encima da *engine* (motor de processamento de um jogo) jMonkey envolvendo várias tecnologias como Blender, RedDwarf, MySQL entre outras. O jogo foi completo após seis interações, é relatado que uma das maiores dificuldades encontradas na primeira iteração foi à falta de familiaridade com as tecnologias e a falta de documentação e informações quanto às ferramentas de código-livre utilizadas, fato anteriormente comentado como comum na grande maioria dos projetos novos.

4. SCRUMBAN APLICADO A JOGOS

O desenvolvimento de um jogo requer uma série de passos e processos assim como o de um software comum, Nos capítulos anteriores foi observada a aplicação dos métodos ágeis Scrum, Kanban e Scrumban em suas formas naturais para o qual cada um foi projetado. Tendo como foco deste trabalho o Scrumban veremos então em seguida alguns pontos diferenciais que devem ser considerados ao aplicar tal método no desenvolvimento de um jogo que possui algumas diferenças quanto ao desenvolvimento de outros tipos de software.

De acordo com Keith (2010), assim como qualquer outro software um jogo, para que possa ser projetado adequadamente, necessita de requisitos, Keith (2010) dá alguns exemplos de requisitos mínimos de um jogo:

- Oito a doze horas de conteúdo de jogo *single-player* (um jogador).
- Data de lançamento fixa específica, por exemplo, natal ou co-lançamento com outra obra como um filme ou série de TV.
- Funcionalidades mínimas requeridas, tais como múltiplos jogadores online para um jogo de tiro em primeira pessoa, as quais devem obrigatoriamente ser lançadas com o jogo.

Tais requisitos necessitam ser levantados e então planejados sempre considerando os recursos em mãos principalmente financeiros e humanos e se os mesmos são suficientes para cobrir os requisitos mínimos necessários, para planejamento a longo termo um conjunto de funcionalidades mínimas precisa ser definido, para um jogo de tiro em primeira pessoa Keith (2010) dá o seguinte exemplo: jogabilidade *single-player*, IA, armas e múltiplos jogadores online.

O desenvolvimento de um jogo requer uma conceituação de projeto diferente das outras, Keith (2010) explica que é necessário separar os processos de produção em estágios quando trabalhando no projeto de um jogo e as três razões principais para isso são:

- **Editores necessitam de conceitos definidos:** Para conseguir aprovação do editor chefe (a qual inclui *marketing* e geralmente a aprovação do proprietário da franquia ou licença), os desenvolvedores precisam criar um conceito detalhado no início do projeto. Eles não poderão vagar para muito longe desta visão no decorrer do projeto.

- **Jogos precisam entregar pelo menos oito horas de jogabilidade:** Jogos tipicamente oferecem de oito a doze horas de jogabilidade em *single-player*. Jogos narram histórias que necessitam de uma grande porção de produção de conteúdo que deve ser feita com as mecânicas descobertas durante o estágio de pré-produção.
- **Data única de lançamento:** Para jogos de larga escala, só há somente uma data de lançamento ao final de 24 ou mais meses de ciclos de produção. Os testes de conformidade de infraestrutura geralmente são adiados até o final.

Dada então a necessidade de particionar à estrutura do projeto é necessário entender quais são esses estágios e como o Scrumban encaixará se sobre os mesmos, tópicos abordados a seguir.

4.1 ESTÁGIOS NO PROJETO DE UM JOGO

Keith (2010) explica que essa necessidade de dividir o projeto em estágios é uma característica peculiar na produção de jogos independente da metodologia a ser utilizada para gerenciar seu desenvolvimento, ele diz que a definição desses estágios pode variar dependendo da forma como o time vai desenvolver o jogo, mas comumente a grande maioria dos projetos de jogos segue o padrão de quatro estágios, sendo estes: o *Concept* (Conceito) seguido da *Pre-Production* (Pré-Produção) que leva a *Production* (Produção) encerrando na *Post Production* (Pós-Produção), mas como o desenvolvimento de um jogo envolve uma série de atividades como codificação, geração de ativos e conteúdo, conceptualização, *design*, otimização entre outros processos no decorrer do projeto isso não significa que os estágios vão ocorrer de forma uniforme divididos perfeitamente em *Sprints* homogêneas.

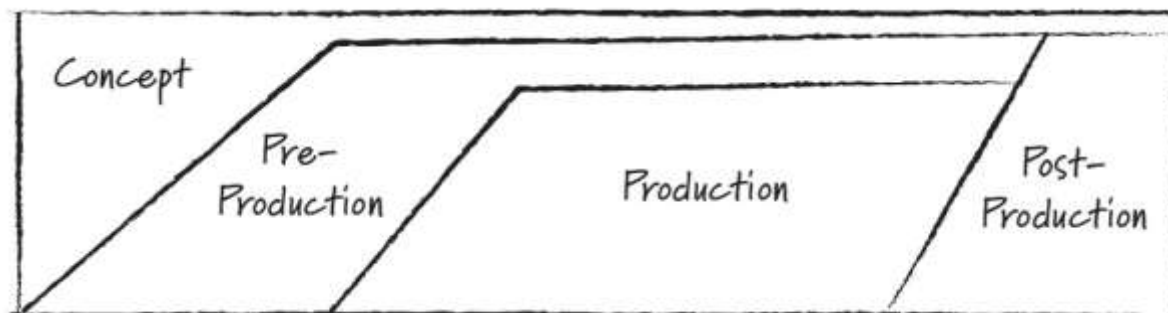


Figura 28 – Exemplificação visual da presença dos estágios durante um projeto (KEITH, 2010).

Seguindo com as colocações feitas por Keith (2010), os estágios não são isolados na linha do tempo do projeto, por exemplo, os maiores esforços quanto à conceptualização estão concentrados no início do projeto, mas conforme as necessidades forem emergindo no decorrer do projeto ocorrerão refinamentos.

Na figura 28 vemos um exemplo típico do comportamento de cada estágio no decorrer de um projeto de jogo devida sua heterogeneidade natural nos processos.

Uma forma de gerenciar os estágios tornando-os compatíveis com o modelo de interações oferecido pelo Scrum é trabalhar com o conceito de lançamentos.

Os lançamentos (Figura 29) são um conjunto de *Sprints* que tem como escopo estar entregando um conjunto majoritário de funcionalidades e (ou) conteúdo.

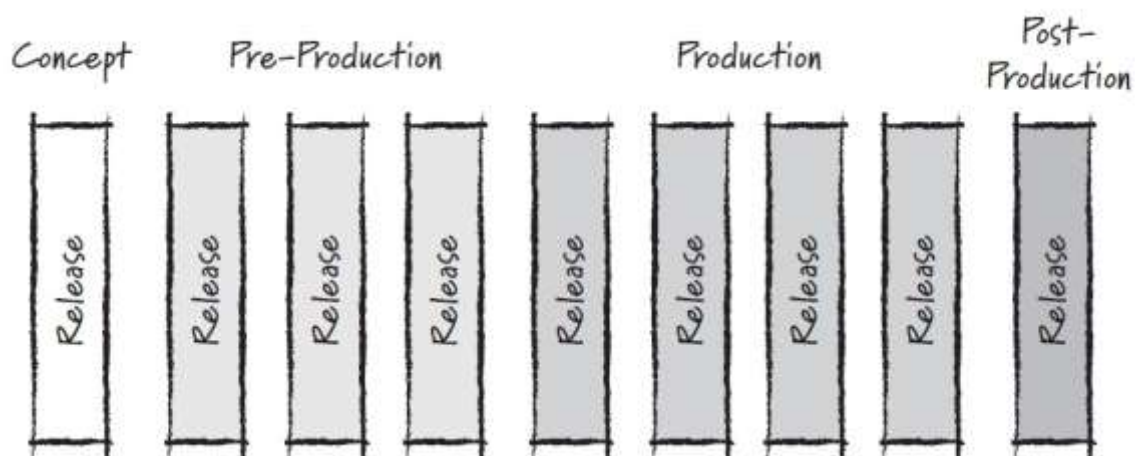


Figura 29 – Conjunto de lançamentos num projeto de múltiplos anos de duração (KEITH, 2010).

Vejamos então como o Scrumban será aplicado em cada estágio nos subtópicos seguintes.

4.1.1 Conceito

Keith (2010) explica que o estágio de Conceito é o primeiro num projeto de jogo vindo antes da pré-produção. O desenvolvimento do conceito é puramente iterativo. Ideias são geradas, possivelmente ocorre prototipificação. Este estágio é geralmente possui um cronograma fechado para entregar um ou mais conceitos de plano de desenvolvimento para obter a aprovação do editor chefe ou detentor da licença que atuará ou deverá contratar alguém para desempenhar em seu lugar o papel de *Product Owner* no decorrer do projeto para garantir que o time não vai se dispersar em relação ao conceito que foi definido e aprovado.

As *Sprints* iniciais que serão diretamente relacionadas ao conceito serão mais curtas e seu *Sprint Backlog* será em grande parte composto por tarefas de estudo e outras atividades relacionadas à curva de aprendizado, o objetivo principal dessas *Sprints* será gerar conhecimento para o time e seus *stakeholders* e não diretamente valor para o usuário como pregam os métodos ágeis, tal passo é necessário para que o time consiga de forma bem sucedida gerar valor para o usuário durante a produção do jogo final.

4.1.2 Pré-Produção

Segundo Keith (2010) neste estágio o time explora o que é divertido e como eles vão gerar conteúdo para servir de base durante a produção. Também são criados níveis e outros tipos de conteúdo que representam a produção da qualidade.

Este estágio é interativo e incremental, de forma iterativa o time vai descobrindo os fatores atrelados à diversão e de forma incremental adaptam o planejamento do desenvolvimento com o conhecimento adquirido.

O comportamento das *Sprints* atreladas a pré-produção é similar ao conceito, mas neste momento o valor de fato começa a ser gerado, pois o objetivo dessas *Sprints* é voltado para as funcionalidades majoritárias.

4.1.3 Produção

De acordo com Keith (2010) neste estágio o time foca-se na criação da experiência de jogo utilizando as mecânicas principais e os processos descobertos durante o estágio de pré-produção.

Este estágio foca na eficiência e melhorias incrementais. Os times trabalham menos a iteração das mecânicas descobertas na pré-produção, pois agora estão produzindo uma grande porção de conteúdo baseado nelas. Mudar estas mecânicas durante o processo de produção gera muitos desperdícios e gastos. O que caracteriza como crítica as mecânicas estudadas e definidas durante a pré-produção devido ao impacto direto que vão ter em todo o processo de produção.

4.1.4 Pós-Produção

Keith (2010) explana que neste estágio o conteúdo foi levado a um nível de qualidade entregável, o time então vai se forçar em polir a experiência de jogo.

Neste estágio o jogo é melhorado de forma incremental. Em seguida o jogo é submetido a testes de infraestrutura nas plataformas as quais será distribuído, mesmo que o processo de teste esteja presente em todas as fases do projeto neste momento são analisados fatores de caráter mais introspectivo e que muitas vezes não podem ser realizados durante o desenvolvimento devido ao custo atrelado a tais processos. A pós-produção tem seu início declarado no lançamento da versão *alpha* do jogo (primeira versão estável e jogável) tendo sua sequência na versão *beta* (versão polida apresentável ao usuário, mas que ainda pode conter erros e vulnerabilidades) tendo por fim a versão final que é lançada e distribuída no mercado.

Aparte de suas peculiaridades e objetivos específicos todas as *Sprints* de todos os estágios seguem os ritos prescritos pelo Scrumban relacionados ao planejamento das atividades, o controle da produção e seu fluxo, as métricas de desempenho, retrospectivas, revisão e apresentação dos resultados de cada *Sprint*.

4.2 SCRUMBAN PARA *SERIOUS GAMES* NA ÁREA DA SAÚDE

Finalmente chegamos ao momento em que aplicamos o método ágil Scrumban no desenvolvimento de um *serious game* voltado para a área da saúde, para que possamos simplificar e tornar mais acessível o conteúdo que será abordado neste capítulo, o foco será expor os pontos diferenciais do desenvolvimento de *serious game* na área da saúde aplicando o Scrumban em relação ao capítulo anterior onde o mesmo é aplicado em um jogo tradicional.

Analisando as publicações de Gestwicki e McNely (2012), Pulman (2009) e Klopfer, Perry, Squire, et al. (2003), é notável que durante o desenvolvimento de *serious games* especialmente os voltados para a área da saúde os estágios de produção e a aplicação do Scrumban podem decorrer da mesma forma que em um jogo tradicional, a diferença essencial está na composição dos **recursos humanos** e no **planejamento** do *serious game* e suas consequências na produção.

Analisando os diferenciais relacionados à composição dos **recursos humanos**, fica clara a necessidade de o time não ser somente composto por profissionais relacionados ao desenvolvimento técnico do jogo como programadores, sonoplastas, artistas gráficos e analistas da qualidade, faz-se necessária a presença de um profissional diretamente ligado à área médica a qual será abordada pelo *serious game*, tal profissional ou profissionais precisa(m) ter experiência suficiente para conseguir descrever os processos envolvidos, o contexto e a teoria por trás do escopo principal abordado pelo jogo pelo ponto de vista técnico da área da saúde e de forma clara compartilhar essas informações com os profissionais dedicados ao desenvolvimento do jogo tanto para auxiliar no direcionamento da produção quanto a análise dos resultados e sua real efetividade para o escopo segundo a área médica.

Aprofundando mais ainda nos relatos de Pulman (2009), fica claro o papel fundamental que o *Scrum Master* terá de desempenhar facilitando a comunicação e

entendimento entre os profissionais ligados diretamente ao desenvolvimento do jogo e a área técnica da saúde que estará presente no suporte.

Também é evidenciado que para que a análise crítica do produto e sua efetividade e valor para o usuário sejam garantidos o *Product Owner* precisa ter conhecimento e experiência tanto com os processos envolvidos no desenvolvimento de *serious games* quanto com a área da saúde, em alguns casos específicos pode ser necessária à presença de um especialista para dar suporte na avaliação e o processo de aprovação.

As diferenças relacionadas ao **planejamento** envolvem em sua base o conceito apresentado por Pulman (2009) Resultados de Aprendizagem Esperados (*ILO - Intended Learning Outcomes*), que é o escopo a ser satisfeito, este conceito é voltado para a parte de treinamento e educação analisando se o usuário está de fato absorvendo com sucesso as informações transmitidas pelo jogo, mas sua essência pode ser replicada para os outros tipos de *serious games* na área da saúde tomando o paradigma de que é essencial que o *serious game* cumpra com sucesso o escopo para o qual foi projetado, por exemplo, se o jogo se propõe a auxiliar no tratamento do TEPT o jogo de fato deve auxiliar o tratamento de alguma maneira, não basta somente conter os elementos relacionados a tal, eles de fato precisam ser testados e evidentemente produzirem resultados positivos no tratamento do paciente.

De acordo com os estudos realizados por Marin, Navarro e Lawrence (2011), deve-se atentar aos seguintes fatores quanto ao escopo de um projeto de *serious game* para saúde:

1. Audiência: Como o foco é um *serious game* para saúde, haverá uma audiência específica a ser atendida por tal e é necessário saber se o jogo está adequado para esta audiência.
2. Objetivo: Esta dimensão está relacionada à qual área do corpo humano se beneficiará do ato de jogar o jogo, qual o tipo do *serious game* dentro das categorias de atuação na saúde e qual o propósito principal de suas melhorias sejam estas físicas ou cognitivas.
3. Interação: A forma a qual o usuário vai interagir com o jogo. Alguns sistemas comerciais utilizam-se de entradas como controles, teclado, mouse, etc... alguns tem seus periféricos próprios ou necessitam que o usuário vista-se de sensores a serem reconhecidos por uma plataforma.

4. Tecnologia: Uma descrição breve dos componentes e técnicas que tornarão possível o funcionamento do jogo.
5. Recursos especiais para a idade apropriada: Melhora ou mecanismo que faz o jogo adequado para determinada idade como, por exemplo, ajustar a dificuldade do jogo dinamicamente.
6. Direcionado ao Lar: O jogo pode ser desenvolvido para ser usado em centros médicos ou clínicas de reabilitação, mas vale ressaltar que a possibilidade de ter acesso a tal tecnologia do lar reduz a necessidade de viagem aos centros médicos e clínicas de reabilitação.
7. Resposta: Esta dimensão está relacionada à resposta dada para o usuário. Alguns jogos representam as ações do jogador somente de forma visual ou sonora ou háptica para notificá-lo sobre suas ações.
8. Métrica: Esta dimensão está relacionada às métricas que podem ser obtidas durante ou após o jogo para ajudar os especialistas a determinar melhorias através de avaliação médica.
9. Relatório de Progresso: Características do jogo que registrem os resultados (pontuação, desempenho, etc...).

O conceito de planejamento de um *serious game* é reforçado no trecho abaixo:

“O estímulo das funções cognitivas, a motivação e a possibilidade de construção de novos conhecimentos são elementos fundamentais em um *serious game*. Por se tratar de uma aplicação de propósito específico, seu planejamento demanda o envolvimento de profissionais da área com a qual o conteúdo se relaciona. Assim, se um jogo é voltado para educação médica, é essencial a participação de profissionais de medicina na equipe de desenvolvimento. Estes irão auxiliar a delinear o escopo do jogo, bem como as maneiras mais adequadas de abordar os conteúdos.” (MACHADO, MORAES, NUNES et al. 2010).

Através de tal definição fica claro que é necessária a presença ativa de um profissional da área médica no desenvolvimento do projeto.

Ao analisar os processos envolvidos no planejamento de um *serious game* nos deparamos com quatro frentes de informações que compõem um conjunto de documentações denominada *design bible*.

Segundo Machado et al. (2010) a *design bible* é composta pelo roteiro, conceituação artística (*game design*), jogabilidade e interface.

Tais elementos não são exclusivos dos *serious games*, mas possuem suas peculiaridades. Os profissionais envolvidos no projeto tanto diretamente em seu desenvolvimento quanto na área técnica da saúde atuarão no planejamento conforme suas especificações para que o *serious game* voltado para a saúde seja concebido com sucesso.

5. CONCLUSÃO

Neste trabalho foi apresentado um estudo a respeito da possibilidade de aplicação do método ágil Scrumban no desenvolvimento de um *serious game* voltado para a área da saúde. Seu desenvolvimento teve uma abrangência multidisciplinar abordando paradigmas de certa forma ainda obscuros dando abertura para uma série de trabalhos futuros descritos posteriormente.

Durante a análise inicial onde foi constatada a presença evolutiva e contínua dos métodos ágeis e levantada a possibilidade de sua aplicação no desenvolvimento de *serious games* em específico na área da saúde três hipóteses foram levantadas, afortunadamente as mesmas tiveram parte no decorrer do trabalho.

O *serious game* “Pappi World” abordado no capítulo 3.1.5. serve de exemplo onde a aplicação de métodos ágeis no desenvolvimento de *serious games* provê resultados positivos e efetivos, o projeto foi concluído e publicado de forma bem sucedida deixando ainda em aberto a proposta de expansão do mesmo.

Através dos relatos de Meydam (2009) e Noronen (2011) que colocaram em prática a criação de Corey Ladas, o Scrumban, foi possível constatar que o mesmo é um equilíbrio entre os dois métodos ágeis de grande ascensão na área de desenvolvimento de software, Kanban e Scrum.

No capítulo 4.2 através da análise de colocações e relatos de vários autores presenciemos o emprego de métodos ágeis ser compatível com os processos de desenvolvimento de um *serious games* na área da saúde dentro de suas necessárias adaptações.

Podemos então considerar que o objetivo inicialmente proposto de entender os processos envolvidos no desenvolvimento de um *serious game* para a área da saúde quando aplicado um método ágil, especificamente o Scrumban, foi alcançado.

No decorrer do trabalho nota-se que o método ágil Scrumban é capaz de servir de base organizacional para um projeto de *serious games* voltado para a área da saúde, por possuir a estrutura necessária para se adequar as necessidades específicas demandadas, mesmo em cenários não favoráveis superando as inúmeras dificuldades que possam estar atreladas sem a necessidade abrir mão de seus conceitos primários herdados de seus ancestrais diretamente ligados ao manifesto ágil e a filosofia Lean.

5.1 TRABALHOS FUTUROS

A primeira e provavelmente a mais ambiciosa e dificultosa seria aplicar de forma prática o Scrumban no desenvolvimento de um *serious game* voltado para a área da saúde, tal iniciativa requer uma série de recursos, principalmente humanos, por envolver o esforço de vários profissionais de múltiplas áreas do conhecimento humano, além de ser uma iniciativa pioneira a nível global.

Outra opção, menos ambiciosa, seria a de aplicar o Scrumban no desenvolvimento de um jogo independente de seu propósito, algo mais próximo da realidade acadêmica por não demandar tantos recursos quanto à primeira sugestão principalmente por eliminar a necessidade de envolver profissionais de outras áreas e recursos estruturais complexos envolvendo outros periféricos além dos que já acompanham um computador pessoal tradicional, não teria o mesmo caráter pioneiro da primeira opção, mas com a devida documentação e registro do decorrer do projeto possui potencial para prover benefícios acadêmicos principalmente no Brasil onde analisando seu cenário atual ainda poderia ser considerada uma iniciativa pioneira.

Por fim, outra sugestão de estudo seria a de juntamente a um acadêmico ou profissional da área da saúde aprofundar-se nos estudos da aplicação do Scrumban no desenvolvimento de *serious games* na área da saúde procurando por formas inovadoras de utilizar-se dos *serious games* na saúde que pudessem ser viabilizadas devido à agilidade e flexibilidade do Scrumban além de identificar possíveis novos papéis, atribuições e técnicas exclusivas ou até mesmo uma nova gama de conhecimentos ainda desconhecidos.

6. REFERÊNCIAS BIBLIOGRÁFICAS

ABORDAGENS PRÁTICAS DE REALIDADE VIRTUAL AUMENTADA, 11, 2009, PUCRS. MACHADO, Liliâne S. ; MORAES, Ronei M. ; NUNES, Fátima L.S. *Serious games para Saúde e Treinamento Imersivo*. Porto Alegre: SBC, 2009. Disponível em: <http://www.de.ufpb.br/~labteve/publi/2009_svrmc.pdf>. Acesso em 4 de outubro de 2012.

ANDERSON, David. *Kanban*. [S.l.]: Blue Hole, 2010.

ANDRADE, Maria Margarida de. *Introdução à metodologia do trabalho científico*. 9ª ed. São Paulo: Atlas, 2009.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. Citação: NBR-10520/ago - 2002. Rio de Janeiro: ABNT, 2002.

_____. Referências: NBR-6023/ago. 2002. Rio de Janeiro: ABNT, 2002.

AUGMENTED HUMAN INTERNACIONAL CONFERENCE, 2, 2011, Odaiba. REGO, Paula Alexandra. ; MOREIRA, Pedro Miguel. ; REIS, Luís Paulo. *New Forms of Interaction in Serious Games for Rehabilitation*. Tokyo: IGI, 2011.

COMPUTER SUPPORT FOR COLLABORATIVE LEARNING 2005: THE NEXT 10 YEARS!, 2005, [S.l.]. KLOPFER, Eric. ; PERRY, Judy. ; SQUIRE, Kurt. ; MING-FONG, Jan. ; STEINKUEHLER, Constance. *Mystery at the museum: a collaborative game for museum education*. Taipei: Routledge, 2005. Disponível em: <<http://education.mit.edu/papers/handheldpapers/Mystery at the Museum- CSCL.pdf>>. Acesso em: 15 de outubro de 2012.

DISTRIBUTED MULTIMEDIA SYSTEMS, 17, 2011, Convitto della Calza. BULFICO, Ida. ; FRANCESE, Rita. ; LETTIERI, Marcio. ; LISCIO, Luca. ; PASSERO, Ignazio. ; TORTORA, Genny. *The TIE Project: Agile Development of a Virtual World Serious Game on Waste Disposal*. Florence: DBLP, 2011. Disponível em:

<http://www.ksi.edu/seke/Proceedings/dms11/DET/7_Rita_Francesse.pdf>. Acesso em 05 de outubro de 2012.

GESTWICKI, Paul. ; MCNELLY, Brian. *A case study of a five-step design thinking process in educational museum game design*. [S.l.: s.n], 2012. Disponível em: <http://meaningfulplay.msu.edu/proceedings2012/mp2012_submission_37.pdf>. Acesso em: 02 de outubro de 2012.

KEITH, Clinton. *Agile Game Development with Scrum*. Boston: Pearson Education, 2010.

KNIBERG, Henrik. *Lean from the Trenches: Managing Large-Scale Projects with Kanban*. Frisco: The Pragmatic Programmers, 2011.

MACHADO, Liliane dos Santos. ; MORAES, Ronei Marcos de. ; NUNES, Fatima de Lourdes dos Santos. ; COSTA, Rosa Maria Esteves Moreira da. Serious Games Baseados em Realidade Virtual para Educação Médica. *Revista Brasileira de Educação Médica*, v.35, n.2, p. 254-262, 04 de outubro de 2010.

MARCHWINSKI, Chet. ; SHOOK, John. *Lean Lexicon* 4ª ed. Cambridge: Lean Enterprise Institute, 2008.

MORAIS, Alana Marques de. *Planejamento e desenvolvimento de um serious game voltado ao ensino de saúde bucal em bebês*. [dissertação de mestrado] – UFPB. João Pessoa, 2011.

PRIES, Kim H. ; QUIGLEY, Jon M. *Scrum Project Management*. Boca Raton: CRC PRESS, 2011.

SCHWABER, Ken. *Agile Project Management with Scrum*. [S.l.]: Microsoft Press, 2004.

SERIOUS GAMES ON THE MOVE, 2008, Anglia Ruskin Univesity. PULMAN, Andy. ; SHUFFLEBOTTOM, Mark. *A Virtual Infection Control Simulation: The Development*

of a Serious Game in the Health-Care Sector. Cambridge: INSPIRE, 2009. Disponível em: <http://www.inspire.anglia.ac.uk/assets/uploads/networks/issue12/virtual_infection_control.pdf>. Acesso em 06 de outubro de 2012.

SUSI, Tarja. ; JOHANNESSON, Mikael. ; BACKLUND, Per. *Serious Games — An Overview*. University of Skövde: DiVA, 2007. Disponível em: <<http://his.diva-portal.org/smash/get/diva2:2416/FULLTEXT01>>. Acesso em: 07 de outubro de 2012.

THE FOURTH INTERNATIONAL CONFERENCE ON ADVANCES IN HUMAN-ORIENTED AND PERSONALIZED MECHANISMS, TECHNOLOGIES AND SERVICES, 4, 2011, UCA. MARIN, Jaime Garcia. ; NAVARRO, Karla Felix. ; LAWRENCE, Elaine. *Serious Games to Improve the Physical Health of the Elderly: A Categorization Scheme*. Barcelona: IARIA, 2011.

WIEMEYER, Josef. ; KLIEM, Annika. Serious games in prevention and rehabilitation—a new panacea for elderly people?. *European Group for Research into Elderly and Physical Activity (EGREPA)*, v.9, n.8, p. 41-50, 8 de dezembro de 2011.

ANDERSON, David J. Kanban – Lack of roles is strength. Disponível em: <http://agilemanagement.net/index.php/Blog/kanban_-_lack_of_roles_is_a_strength/>. Acesso em: 01 de novembro de 2012. 15h37.

BREAKAWAY. Free Dive. Disponível em: <<http://www.breakawaygames.com/serious-games/solutions/healthcare/>>. Acesso em: 09 de outubro de 2012. 16h40.

BUSINESS WIRE. Spend on Serious Games Growing Steadily; Now a Multi-Billion Dollar Industry. Disponível em: <<http://www.businesswire.com/news/home/20120823006124/en/Spend-Games-Growing-Steadily-Multi-Billion-Dollar-Industry>>. Acesso em: 06 de novembro de 2012. 13h00.

CARLOS, Miguel. Artefatos do Scrum: Product Backlog. Disponível em: <<http://projetoseti.com.br/gestao/gp-scrum/artefatos-do-scrum-product-backlog/>>. Acesso em: 30 de outubro de 2012. 13h13.

COHN, Mike. Training for Scrum Task Board Use. Disponível em:
<<http://www.mountangoatsoftware.com/scrum/task-boards>>. Acesso em: 29 de outubro de 2012. 02h05.

COTTMEYER, Mike. ; STEVENS, Dennis. Kanban for Agile Teams. Disponível em:
<http://www.versionone.com/White_Papers/kanban/>. Acesso em: 01 de novembro de 2012. 13h34.

GLOGER, Boris. Scrum Tools | Taskboard. Disponível em:
<<http://borisgloger.com/2008/09/11/scrum-tools-taskboard/>>. Acesso em: 30 de outubro de 2012. 11h20.

ITUNES. Fear of Spiders Treatment HD. Disponível em:
<<https://itunes.apple.com/br/app/fear-of-spiders-treatment-hd/id418167243?mt=8>>. Acesso em: 17 de outubro de 2012. 16h00.

JOHNSON, Joel. From Atari Joyboard to Wii Fit: 25 years of “exergaming”. Disponível em: <<http://gadgets.boingboing.net/2008/05/15/from-atari-joyboard.html>>. Acesso em: 18 de outubro de 2012. 13h50.

KAISER PERMANENTE®. Snacktown Snackdown. Disponível em:
<<http://info.kaiserpermanente.org/richmedia/kidWisdom/snacktown/index.html>>. Acesso em: 16 de novembro de 2012. 15h41.

LADAS, Corey. Scrum-ban. Disponível em:
<<http://leansoftwareengineering.com/ksse/scrum-ban/>>. Acesso em: 01 de novembro de 2012. 19h19.

LAZARUS, Melanie M. The History of Exergames. Disponível em:
<<http://www.healthgamers.com/2010/exergaming/the-history-of-exergames/>>. Acesso em: 08 de outubro de 2012. 13h13.

LEGACY INTERACTIVE. Pet Pals: Animal doctor. Disponível em:
<<http://www.amazon.com/gp/product/B000QYLGEY/?tag=indonvetercom-20>>.
Acesso em: 08 de outubro de 2012. 18h30.

LEGGETT, Hadley. Virtual Reality Could Keep You From Being a Surgical Guinea Pig. Disponível em: <<http://www.wired.com/wiredscience/2009/08/virtualsurgery/>>.
Acesso em: 08 de outubro de 2012. 17h40.

LUDUS. Chapter 1: What a Serious Game is? Definitions and theoretical frame work for Serious Games. Disponível em: <http://www.serious-gaming.info/@api/deki/files/57/=Chapter_1.pdf>. Acesso em: 06 de novembro de 2012. 14h00.

LUMOSITY. Engaging exercises. This is your brain on exercise. Disponível em:
<<http://www.lumosity.com/why-lumosity/engaging-exercises>>. Acesso em: 08 de novembro de 2012. 17h20.

MADGAMER. Full Spectrum Warrior – Ten Hammers. Disponível em:
<<http://www.madgamer.sk/pouzite-hry/playstation-2/akcna/full-spectrum-warrior-ten-hammers>>. Acesso em: 07 de novembro de 2012. 18h00.

MEYDAM, Jens. Scrum(ban). Disponível em:
<<http://www.slideshare.net/jmeydam/scrumban-jmeydam>>. Acesso em: 01 de novembro de 2012. 22h21.

MINDHABITS. MindHabits. Disponível em: <<http://www.mindhabs.com/>>. Acesso em: 08 de novembro de 2012. 14h32.

MORDEZKI, Miguel. ¿Qué es Scrum? Disponível em:
<<http://www.cafeproyectos.com/practicass-lean-y-agil-en-gestion-de-proyectos/una-practica-agil-scrum/>>. Acesso em: 31 de outubro de 2012. 06h16.

MURRAY, Martin. Introduction to Kanban. Disponível em:
<<http://logistics.about.com/od/tacticalsupplychain/a/Introduction-To-Kanban.htm>>.
Acesso em: 01 de novembro de 2012. 11h50.

NORONEN, Heikki-Pekka. Scrumban makes my teeth laugh. Disponível em:
<<https://agilebackblog.wordpress.com/>>. Acesso em: 12 de novembro de 2012.
15h39.

OXFORD ADVANCED LEARNER'S DICTIONARY. Lean. Disponível em:
<<http://oald8.oxfordlearnersdictionaries.com/dictionary/Lean>>. Acesso em: 01 de
novembro de 2012. 13h26.

PATTON, Jeff. Kanban Development Oversimplified. Disponível em:
<http://agileproductdesign.com/blog/2009/kanban_over_simplified.html>. Acesso em:
01 de novembro de 2012. 12h26.

PRIMO, Glauco. User Stories – O que são? Como Usar? Disponível em:
<<http://blog.myscrumhalf.com/2011/10/user-stories-o-que-sao-como-usar/>>. Acesso
em: 30 de outubro de 2012. 10h42.

ROOCK, Stefan. Kanban: Definition of Lead Time and Cycle Time. Disponível em:
<<http://stefanroock.wordpress.com/2010/03/02/kanban-definition-of-lead-time-and-cycle-time/>>. Acesso em: 01 de novembro de 2012. 16h27.

SAMPAIO, Marcus Costa. Casos de Uso. Disponível em:
<<http://www.dsc.ufcg.edu.br/~sampaio/cursos/2007.1/Graduacao/SI-II/Uml/diagramas/usecases/usecases.htm>>. Acesso em: 29 de outubro de 2012.
20h59.

SERIOUS GAMES INITIATIVE. The Serious Games Initiative. Disponível em:
<<http://www.seriousgames.org/>>. Acesso em: 06 de novembro de 2012. 12h21.

SIMOES, Roberto. Pilares do Scrum e Timeboxes. Disponível em:
<<http://www.scrumrj.org/tag/transparencia/>>. Acesso em: 31 de outubro de 2012.
08h16.

SINGLETON, Andy. Scrum + Kanban = ScrumBan, an Easy Scrum Upgrade.
Disponível em:
<<http://blog.assembla.com/assemblablog/tabid/12618/bid/87904/Scrum-Kan.aspx>>.
Acesso em: 01 de novembro de 2012. 20h27.

STRICKLER, Jon. Agile on a Single Page. Disponível em:
<<http://agileelements.wordpress.com/>>. Acesso em: 10 de outubro de 2012. 23h59.

THOMPSON, Kevin. Scrum FAQ. Disponível em:
<http://www.cprime.com/about/scrum_faq.html>. Acesso em: 17 de outubro de 2012.
01h59.

TOOMBS, David E. Kanban as a Tool in the Agile Toolbox. Disponível em:
<<http://pt.scribd.com/doc/85438867/Kanban-as-a-Tool-in-the-Agile-Toolbox>>. Acesso
em: 01 de novembro de 2012. 12h56.

TOYOTA. History. Disponível em: <<http://www.toyotageorgetown.com/history.asp>>.
Acesso em: 01 de novembro de 2012. 12h14.

VRMC. The Virtual Reality Medical Center. Disponível em:
<<http://www.vrphobia.com>>. Acesso em: 07 de novembro de 2012. 14h00.