



**INSTITUTO FEDERAL
SANTA CATARINA**

**CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS
CURSO SUPERIOR DE GESTÃO DA TECNOLOGIA DA
INFORMAÇÃO**

Frederico Dubiel

**ScrumBoard:
um jogo para o auxílio do
ensino de Scrum**

**Florianópolis – SC
2017**

Ficha de identificação da obra elaborada pelo autor.

Dubiel, Frederico

ScrumBoard: um jogo para o auxílio do ensino de SCRUM
/ Frederico Dubiel ; orientação de Emerson Pessoa. - Florianópolis,
SC, 2018.

92 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal
de Santa Catarina, Câmpus Florianópolis. CST
em Gestão de TI. Departamento Acadêmico de Saúde
e Serviços.

Inclui Referências.

1. Scrum. 2. Agil. 3. Educação. I. Pessoa, Emerson.
II. Instituto Federal de Santa Catarina. Departamento
Acadêmico de Saúde e Serviços. III. Título.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA TECNOLOGIA DA
INFORMAÇÃO**

FREDERICO DUBIEL

ScrumBoard: um jogo para o auxílio do ensino de SCRUM

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Tecnólogo em Gestão da Tecnologia da Informação.

Professor Orientador:
Prof. Emerson Pessoa, Dr.

FLORIANÓPOLIS - SC
2017/2

ScrumBoard: Um jogo para o auxílio do ensino de SCRUM

FREDERICO DUBIEL

Este trabalho foi julgado adequado para obtenção do Título de Tecnólogo em Gestão da Tecnologia da Informação e aprovado na sua forma final pela banca examinadora do Curso Superior de Tecnologia em Gestão da Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis-SC, 04 de Dezembro de 2017.



Prof. Julio Cesar da Costa Ribas, Dr.
Coordenador do CST em Gestão da Tecnologia da Informação
Instituto Federal de Santa Catarina

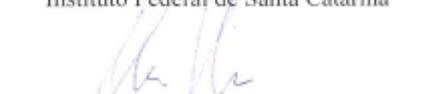
Banca Examinadora:



Prof. Emerson Pessoa, Dr
Orientador
Instituto Federal de Santa Catarina



Felipe Cantorio Soares, Dr
Instituto Federal de Santa Catarina



Glauco Cardozo, Mestre
Instituto Federal de Santa Catarina

*À minha mãe e à minha mulher, que me ajudaram a levantar a cada tropeço
neste caminho*

AGRADECIMENTOS

Agradeço, primeiramente, a minha mãe e esposa que me ajudaram e apoiaram em todo caminho. Agradeço a minha família toda pelo carinho e inspiração. Agradeço aos amigos e colegas do meu grupo de orientação que estiveram ao lado dando suporte. Ao meu orientador que me guiou durante este trabalho, aconselhando e apoiando.

*“Carry on my wayward son/
For there'll be peace when you are done”*

Kerry Livgren.

RESUMO

O presente trabalho trata da prototipação de uma ferramenta lúdica, na forma de um jogo de tabuleiro, para o auxílio ao ensino do Scrum dentro de um ambiente de sala de aula. Tendo em consideração que os alunos quando passam por uma aula teórica têm dificuldade em estabelecer uma conexão entre o ensinado e o que é encontrado do dia a dia de trabalho na utilização do Scrum.

Para a criação desta ferramenta utilizou-se da revisão sistemática no intuito de encontrar ferramentas já existentes para esta finalidade e, a partir da avaliação destas, construir um protótipo funcional. Utilizou-se a metodologia de prototipação rápida para construir um modelo de testes e alcançar um conjunto de regras e elementos que fossem coerentes e adequados para o ensino do Scrum.

O jogo foi aplicado a uma turma do IFSC e os resultados, baseados no modelo MEEGA+, foram avaliados como positivos para o uso do ScrumBoard como ferramenta lúdica de auxílio para o ensino do Scrum.

Palavras-chave: Ágil. Educação. *Scrum*

ABSTRACT

The present work deals with the prototyping of a ludic tool, in the form of a board game, to aid the teaching of Scrum within a classroom environment. Taking into consideration that the students in a theoretical class have difficulty in establishing a connection between what is taught and what is found of the day to day work in the use of Scrum.

In order to create this tool, a systematic review was used to find the existing tools with this same purpose and, from the evaluation of these tools, to build a functional prototype. Rapid prototyping methodology was used to construct a test model and to achieve a set of rules and elements that were coherent and appropriate for teaching Scrum.

The game was applied to an IFSC class and the results, based on the MEEGA + model, were evaluated as positive for using the ScrumBoard as a ludic tool for teaching Scrum.

Key-words: Agile. Education. Scrum.

LISTA DE ILUSTRAÇÕES

Figura 1: Uso de cartas para auxílio ao ensino de filosofia.....	14
Figura 2: Representação dos grupos de processos PMBoK.....	21
Figura 3: Matriz de processos do PMBoK.....	24
Figura 4: Representação do modelo Cascata.....	25
Figura 5: Representação do modelo V.....	26
Figura 6: Pesquisa do Standish Group sobre sucesso de projetos versus metodologia aplicada	29
Figura 7: Representação do Scrum, com suas reuniões, papéis e artefatos.....	34
Figura 8: Figura exemplificando o modelo iterativo de prototipação rápida.....	40
Figura 9: Figura exemplificando questionário e relatórios sobre o MEEGA+.....	41
Figura 10: Tabuleiro do Scrumble.....	47
Figura 11: Representação de protótipo de baixa fidelidade para Iteração I.....	52
Figura 12: Representação da carta de Trabalhador.....	54
Figura 13: Protótipo de carta de tarefa.....	54
Figura 14: Protótipo de Bug.....	55
Figura 15: Protótipo exemplo de carta de Impedimento.....	56
Figura 16: Protótipo exemplo da carta de ação.....	56
Figura 17: Legenda para carta de Tarefa.....	59
Figura 18: Fotos da aplicação do jogo.....	62

LISTA DE ABREVIATURAS E SIGLAS

<u>MEEGA</u>	<i>Model for Evaluatign Educational Games</i>
PO	<i>Product Owner</i>
PMBok	<i>Project Management Book of Knowledge</i>
SM	<i>ScrumMaster</i>

SUMÁRIO

1.introdução.....	13
1.1.JUSTIFICATIVA.....	13
1.2.DEFINIÇÃO DO PROBLEMA.....	13
1.3.OBJETIVOS.....	14
1.3.1.Objetivo Geral.....	14
1.3.2.Objetivos Específicos.....	14
1.4.ESTRUTURA DO TRABALHO.....	14
2.Fundamentação Teórica.....	13
2.1.EDUCAÇÃO LÚDICA.....	13
2.1.1.Jogos no ensino de Engenharia de Software.....	18
2.2.GERENCIAMENTO DE PROJETOS.....	19
1.1.MODELOS DE PROCESSOS PRESCRITIVOS OU CLÁSSICOS.....	24
2.3.METODOLOGIAS ÁGEIS.....	27
2.4.SCRUM.....	30
2.4.1.Sprint.....	30
2.4.2.Artefatos.....	31
2.4.4.Reuniões.....	34
2.5.Procedimentos Metodológicos.....	38
2.6.REVISÃO SISTEMÁTICA.....	38
2.7.PROTOTIPAÇÃO RÁPIDA.....	39
2.8.AVALIAÇÃO PELO MODELO MEEGA+.....	41
3.DESENVOLVIMENTO DO TRABALHO.....	42
3.1.REVISÃO SISTEMÁTICA.....	42
3.2.SCRUM CARD GAME.....	42
3.3.SCRUMBLE.....	45
3.4.LEGO CITY.....	48
3.5.PROTOTIPAÇÃO DO JOGO.....	51
3.5.1.Iteração I.....	51
3.5.2.Iteração II.....	53
3.5.3.Iteração III.....	55
3.5.4.Iteração IV.....	57
4.ScrumBoard.....	57
4.1.ITENS DO JOGO.....	57
4.2.REGRAS.....	58
4.2.1.Preparação.....	58
4.2.2.Início da <i>sprint</i>	59
4.2.4. Fim da <i>sprint</i>	61
4.2.5.Cartas de Ação X Impedimento.....	61
4.2.6.Bugs.....	61
4.3.AVALIAÇÃO DO SCRUMBOARD.....	62
5.CONCLUSÕES.....	68
Referências.....	69
APÊNDICE A – Avaliação Modelo MEEGA+.....	72
Apêndice B – Elementos do ScrumBoard.....	75

1. INTRODUÇÃO

O presente trabalho pretende sugerir uma ferramenta–auxiliar para reduzir a falta de aderência da teoria das metodologias ágeis ao que é encontrado no dia a dia. Não se trata de dizer que a teoria é transmitida de forma incorreta, nem de que a realidade utiliza regras diferentes das que são pregadas na teoria. Contudo, tanto no ensino formal quanto em cursos de metodologias ágeis, o que é visto em sala de aula conflita com a realidade.

1.1. JUSTIFICATIVA

O objetivo deste trabalho é criar um jogo de tabuleiro para mostrar, em um ambiente tolerante a erros e de maneira rápida e lúdica, o funcionamento do *Scrum* de forma prática aprimorando, assim, o ensino-aprendizagem de metodologias ágeis, em específico do *framework Scrum*. Dessa forma, visa-se beneficiar alunos e professores que estejam estudando e ou ministrando o *framework*.

Jogos de forma geral, são utilizados como alternativas de educação, e a utilização de um jogo favorece o trabalho em equipe e a resolução de problemas de forma lúdica e cumprimento de um, ou mais, objetivos.

A adoção e compreensão do ágil, e especificamente do *framework Scrum* para criação de um processo de desenvolvimento pode ser de difícil absorção para a organização ou mesmo compreensão individual. Entre os motivos, na perspectiva de COHN (2010, p.5): “[...] o estado final é imprevisível; o *Scrum* é pervasivo; o *Scrum* é dramaticamente diferente [...]”.

1.2. DEFINIÇÃO DO PROBLEMA

O conteúdo das metodologias ágeis, quando ministrado apenas teoricamente, costuma não ter uma aderência de conhecimento entre o teórico e o prático. Quando estudado apenas teoricamente, o conteúdo costuma ser abstrato, uma receita de bolo a ser seguida e com informações longe da realidade do aluno. Mesmo quando é exercitada uma parte prática, esta costuma conter apenas um processo sem as atribuições do dia a dia de uma organização real, ou ainda realizar tarefas que não estão em contato com a realidade de produção de uma organização de desenvolvimento de *software*.

Essa dificuldade acarreta problemas de adaptação dos alunos quando eles saem do escopo acadêmico para entrarem no mercado de trabalho. Observa-se nesse ponto que existe uma lacuna entre os alunos que estão sendo formados e o que o mercado necessita [...] (SILLER; BRAGA, 2013, p. 152).

Desta forma como podemos compreender a que medida a utilização de ferramentas lúdicas

auxilia o aprendizado de metodologias ágeis?

1.3. OBJETIVOS

1.3.1. Objetivo Geral

Criar protótipo de um jogo de tabuleiro, para auxiliar o ensino do *framework Scrum*.

1.3.2. Objetivos Específicos

- Revisar jogos já utilizados para prática de ensino de metodologias ágeis.
- Construir uma versão inicial com regras simplificadas.
- Evoluir da versão inicial para uma versão com regras que compreendam papéis de *ScrumMaster* – SM, Time e *Product Owner* – PO.
- Realizar uma aplicação piloto.
- Avaliar a influência da utilização do jogo na educação.

1.4. ESTRUTURA DO TRABALHO

Este trabalho está estruturado em cinco capítulos. O primeiro capítulo foca na fundamentação teórica, explicando os conceitos que foram utilizados para a construção do trabalho, como a educação lúdica, metodologias de engenharia de software, gerenciamento de projetos, métodos ágeis e o *Scrum*. O capítulo dois apresenta as metodologias de trabalho utilizadas, como revisão sistemática e prototipação rápida. O terceiro capítulo trata do desenvolvimento do trabalho, de suas iterações de criação para o *ScrumBoard* tomar forma. No quarto capítulo, é apresentado o jogo completo, suas regras e mecânicas, e a avaliação feita com uma turma piloto. O trabalho encerra com o quinto capítulo, que apresenta as conclusões sobre o trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. EDUCAÇÃO LÚDICA

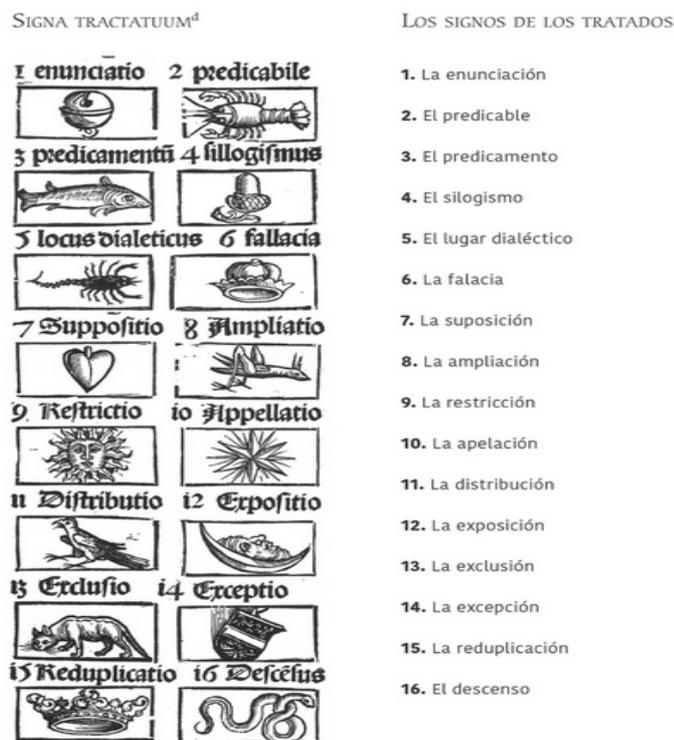
O lúdico se refere àquilo que vem da brincadeira, do jogo, do que nos tira da realidade. Lúdico tem origem do grego *Ludos* e significa jogar, simular. A educação lúdica é um dos métodos utilizados para o processo de ensino-aprendizagem, em diversas disciplinas e em variadas faixas etárias.

Desde a infância, são utilizados jogos e brincadeiras para que a criança se socialize e possa se familiarizar com regras ou com determinada situação. Jogos fazem parte da estratégia em diversas fases do ensino.

Outras ferramentas lúdicas, são utilizadas para evidenciar pontos específicos, metáforas, para auxiliar o aluno a alcançar o conhecimento desejado pelo professor. Jogos de empresa, por exemplo, são utilizados para simular tomadas de decisão, evidenciar a necessidade do planejamento e pensamento estratégico.

Pensadores e professores de diversas áreas utilizaram o lúdico para o ensino. Kishimoto (1990) cita Thomas Murner (1475 - 1537), um frade franciscano, que desenvolveu um jogo de cartas para o ensino da filosofia com o uso de imagens, visto que seus alunos tinham dificuldades para compreender os textos em espanhol disponíveis. Em sua obra, Delgadillo e Puente (2017, p.8) apresentam uma ilustração do jogo de cartas para o ensino da filosofia de Murner, conforme demonstrado na imagem abaixo:

Figura 1: Uso de cartas para auxílio ao ensino de filosofia



Fonte: Delgadillo e Puente (2017)

Ou, ainda, como podemos observar:

“... Portanto, as atividades lúdicas são excitantes, mas também requerem um esforço voluntário, uma vez que mobilizam esquemas mentais. Sendo uma atividade física e mental, a ludicidade aciona e ativa as funções psico-neurológicas e as operações mentais, estimulando o pensamento. Com isso integram as várias dimensões da personalidade: afetiva, motora e cognitiva. Como atividade física e mental que mobiliza as funções e operações, a ludicidade aciona tanto a esfera motora, quanto a cognitiva, e à medida que gera envolvimento emocional, apela para a esfera afetiva.”(AECV; MJMS, 2012 p.40)

O jogo não é meramente um meio de entretenimento, visto que várias doutrinas educacionais observam a utilização de jogos de tabuleiro, a exemplo do xadrez, como forma de exercício lógico e matemático. O jogo de xadrez, diretamente, não possui regras matemáticas ou lógicas aplicadas, porém para que seja jogado, é necessário que os jogadores,

observem as possíveis ações a partir do estado do tabuleiro atual, quais os possíveis movimentos do adversário, levando em conta a jogada realizada, extraindo as jogadas mais prováveis, considerando uma estratégia de jogo, até a captura do rei adversário. Exercita, portanto, o pensamento estratégico dos participantes indiretamente durante o processo da partida.

O jogo não precisa ser uma forma de exercício abstrato, ele pode ser usado como meio de simulação do conteúdo que está sendo ensinado. Assim, permite que o aluno possa observar as situações em que o conteúdo se aplica, permitindo-lhe interagir com este ambiente simulado e livre de estresse. Ao jogador é oferecida a oportunidade de tomar decisões que não tomaria na vida real, podendo refazer suas ações de infinitas maneiras para observar o resultado, e extrair assim um conhecimento profundo e, já com experiência, saber quais ações acarretaram resultados e compreende o porquê destes resultados.

Como exemplo de jogos de simulação pode-se observar jogos ou competições de empresas, os quais são utilizados em diversas instituições acadêmicas. Coloca-se os participantes em contato com trabalho em equipe, observação de dados, tomada de decisão, gerenciamento de riscos, em um ambiente simulado. Baseados numa situação apresentada pelo jogo, os participantes tomam suas decisões e fazem suas jogadas conforme regras previamente estipuladas. O jogo então, devolve o resultado das jogadas, considerando as decisões dos praticantes e o próprio conjunto de regras existentes.

Desta forma, se o participante tomar uma decisão que acarrete uma má jogada, nada de pior do que ter perdido o jogo ocorrerá. Além disso, seja o resultado qual for, o participante, pode observar o resultado das suas ações no jogo, e pode concluir o porquê do resultado e tentar novamente observando um novo conjunto de decisões e suas consequências.

A utilização do lúdico como estratégia de ensino não se prende apenas ao ensino infantil. A aplicação de brincadeiras simples, é utilizada, e pode ser amplamente praticada em qualquer fase de ensino, sem restrições com relação a faixa etária que esteja sendo aplicada, pois como refere Santos (2000 p.19) “[...]A ludicidade é, então, uma necessidade do ser humano em qualquer idade e não pode ser vista apenas como diversão. [...]”.

Na mesma linha de pensamento Lopes afirma:

“É muito mais eficiente aprender por meio de jogos e, isso é válido para todas as idades, desde o maternal até a fase adulta. O jogo em si, possui componentes do cotidiano e o envolvimento desperta o interesse do aprendiz, que se torna sujeito ativo do processo [...]” (LOPES, 2001, p. 23. apud Cortez 2000)

O ato de jogar, não termina com a infância. O homem tem necessidade de se divertir, fazendo isso de diversas maneiras, desde jogos de cartas, até o jogo de futebol do final de semana. A ascensão de jogos de tabuleiros modernos, que pode ser observada em vários estabelecimentos especializados, é outra evidência de que continuamos jogando, e de que, conforme amadurecemos, a necessidade de sofisticação do jogo também cresce. Pode-se então, fazer uso desta característica para utilizar o jogo como uma ferramenta que é tão aceita quanto necessária para o participante/aluno.

A esse respeito, Santos afirma: (2000, p.19) “[...]O jogo não é somente um divertimento ou uma recreação. Os jogos são atividades naturais que satisfazem a atividade humana, e é necessário seu uso dentro da sala de aula.[...]”.

Na mesma linha, Tavares comenta que:

“Para Piaget (1978), os jogos educacionais se caracterizam por ter duas funções, fixar e ou melhorar a fixação dos conteúdos em estudo e ajudar no equilíbrio emocional da criança. Como citado anteriormente a questão emocional está ligada aos problemas relacionados com o aprendizado, o jogo pode suprir essas necessidades.” (PIAGET, 1978 apud TAVARES 2013, p.17)

A utilização de jogos educacionais é um facilitador para a geração do conhecimento pelo aluno. Acompanhado do conteúdo, o jogo pode ser utilizado, para auxiliar a construção do conhecimento pelo aluno, fazendo que este valide o conhecimento aprendido, coloque em cheque os conceitos construídos e valide as conclusões que abstraiu durante o processo. Podendo, e devendo, ainda, corrigir erros e acertos durante as etapas do jogo o conteúdo que adquiriu.

Através de uma visão construtivista:

“O uso de modelos pedagógicos e de jogos educacionais que levem o aluno a refletir sobre o conteúdo por intermédio do raciocínio próprio com interações que exercitem o conhecimento adquirido torna-se, então, ferramenta muito importante[...]” (KRASILCHICK, 2004 apud CONGRESSO NACIONAL DE EDUCAÇÃO, 2015 p.2).

Ou seja, através da vivência, ainda que obtida por meio da simulação proporcionada por um jogo ou dinâmica, é possível construir conhecimento. É da combinação entre a situação concreta e o conhecimento prévio adquirido no jogo ou fora dele, que se pode construir um conhecimento próprio e não simplesmente tentar absorver um conhecimento vindo de um narrador e tentar memorizar um conjunto de diretrizes. Possibilita-se, dessa forma, a compreensão das diretrizes passadas, de acordo com a vivência do sujeito de acordo com o próprio contexto, trazendo para a realidade dele próprio.

O quadro abaixo apresenta uma adaptação das características de jogos educacionais, conforme destaca Domingues et al. (2014).

Quadro 1 - Características de jogos educacionais	
Co-design	O Jogador vê suas ações interferindo no estado do jogo.
Ferramentas inteligentes e conhecimento distribuído	As situações e regras apresentadas pelo jogo, podem ser aplicadas na realidade.
Frustração Prazerosa	Os desafios dentro do jogo colocam os jogadores atuando no limite de suas habilidades
Identidade	O jogador assume uma persona que possui características para atuar dentro do jogo. Ao assumir esta persona ele se engaja no jogo para cumprir seus objetivos.
Informação sob demanda e no momento certo	As informações e regras do jogo são apresentadas ao longo da necessidade e decorrência do jogo, e não em um único bloco massivo de regras
Pensamento sistêmico	Os jogadores são levados a pensar nas relações entre ações e situações do jogo e como elas interferem neste sistema
Problemas bem ordenados	Os problemas apresentados aos

jogadores, auxiliam a criação de soluções de problemas futuros dentro do jogo.
--

Considerando a realidade de ensino de engenharia de *software* para o ensino de métodos de desenvolvimento, o jogo enquanto abstração, permite que o participante possa se libertar de limitações da realidade em que vive. No ambiente do jogo, observa-se a situação apresentada de forma indireta, desprovida da pressão existente em um ambiente real.

Jogos, principalmente os associados a simulação (*mimicry*) permitem que se observe as consequências dos atos, pois demonstram os efeitos de causa e consequência das ações realizadas. Porém sem os impactos negativos envolvidos em uma situação de erro, na qual a decisão tomada não é computada, pelo sistema: regras do jogo, como a mais correta, ou coerente com a situação.

2.1.1. Jogos no ensino de Engenharia de Software

Navarro, Baker e Van der Hoek (2004) descrevem a problemática do ensino da engenharia de *software* nas instituições acadêmicas, visto a diferença das habilidades e conhecimento entre o que é ensinado e o requerido. Isso se dá pelo fato que as aulas teóricas são de um ensino passivo e quando há uma parte prática, estas são restritas pelo modelo acadêmico.

Para Coelho (2011, p. 307), “O jogo traz consigo aspectos da vida diária como: competições, jogos de sorte, problemas desafiadores que precisam de força, outros de raciocínio(...)”.

Segundo Callois apud Coelho (2011), jogos podem ser de quatro tipos, *Agon*, *Alea*, *Mimicry* e *Ilinx*. Onde *Agon*, define a condição da competição, definição de um vencedor ou um grupo vencedor sobre os demais, a partir das habilidades que possui. *Alea* são os tipos de jogos que dependem da sorte, da aleatoriedade da situação, da lançada dos dados, onde a emoção de cada jogada ou decisão está lançada nas probabilidades e possibilidades de um conjunto de cartas. *Mimicry* são os jogos onde um jogador assume um determinado papel,

vive uma experiência simulada, onde ele pode utilizar de qualquer solução que sua imaginação permita para entrar no jogo, e interagir naquele novo mundo. *Ilinx* são as brincadeiras ou jogos que dão sensação de vertigem, onde o jogador busca um desequilíbrio ou ação físico ou psicológico para dar sentido a brincadeira.

Esta classificação não é excludente, ou seja, um jogo não precisa pertencer a somente uma classificação, ele pode ser competitivo e ter elementos de sorte, por exemplo jogos de cartas, pode ser uma simulação de ter estímulo físico, por exemplo *Paintball*, Ele pode ter, elementos de *Agon*, *Alea* e *Mimicry*, como ocorre nos jogos de *Role Playing Games* – RPG. Jogos que possuem mais elementos de diferentes classificações tendem a ser mais estimulantes, enquanto jogos que dependem apenas de um elemento de classificação, tendem a ficar enfadonhos e repetitivos ao longo do tempo.

2.2. GERENCIAMENTO DE PROJETOS

Antes de entender no que consiste o gerenciamento de projetos, e qual o impacto disso no desenvolvimento de software, vamos definir o que é um projeto. Tudo o que desenvolvemos pode ser considerado Projeto? Segundo Heldman (2005), Projetos são únicos; são temporários; são considerados como concluídos quando as metas forem alcançadas. O *Project Management Institute* (2008) afirma que cada projeto cria um produto ou serviço únicos, que o projeto deve ser finalizado quando as metas forem alcançadas, ou se observar que estas não têm como ser entregues, ou quando o projeto for determinado como desnecessário.

Portanto, podemos entender que gerenciamento de projetos é o conjunto de ações, ferramentas, técnicas, artefatos e conhecimentos com o objetivo de conduzir um projeto do seu início ao fim, com sucesso e com eficácia.

Gerenciamento de projetos está ligado a uma série de processos que auxiliam o planejamento, a execução, o controle e a finalização de projetos, de maneira que este possa ser executado da melhor forma possível, ou mesmo identificar se o projeto não é mais viável ou não é mais necessário, sem a necessidade de exaurir recursos. Para Bomfin, Nunes e

Hastenreiter (2012, p.65) “Gerenciamento de projetos pode ser capaz de detectar e controlar custos e prazos mantendo a competitividade a fim de superar as expectativas dos clientes.”

Segundo o PMBoK para que um projeto seja bem-sucedido, a equipe envolvida deve:

- Selecionar os processos apropriados necessários para cumprir os objetivos do projeto;
- Usar uma abordagem definida que possa ser adotada para atender aos requisitos;
- Cumprir os requisitos para atender às necessidades e expectativas das partes interessadas e obter um equilíbrio entre as demandas concorrentes de escopo, tempo, custo, qualidade, recursos e riscos, para gerar o produto, o serviço ou o resultado especificado.

Podemos observar então que, por mais que o PMBoK apresente uma série de boas práticas, artefatos, técnicas e conhecimentos, cada gerenciamento pode ser tão único quanto o projeto pelo qual é responsável. Ele exige apenas que haja um processo definido a ser seguido e nada impede que este processo seja alterado no decorrer do projeto, desde que haja uma correta observação e organização de como e porque estes processos existem.

Para que o gerenciamento de projetos ocorra da melhor maneira, existe um profissional especializado no gerenciamento de projetos, este profissional deve estar munido das técnicas, ferramentas e conhecimentos citados para que tenha a possibilidade de gerenciar o projeto de forma eficaz. De acordo com o *Project Management Institute* (2008 p. não informada) “O gerente de projetos é a pessoa designada pela organização executora para atingir os objetivos do projeto”. Este profissional deve ser multifacetado, podendo operar em várias etapas do projeto. Dele também se espera que tenha uma boa comunicação, pois lidará com todos os interessados e os papéis que estão comprometidos com o projeto. Portanto, ele deverá ser capaz de lidar e fazer com que estes lados possam se comunicar.

O PMBoK divide o gerenciamento de projetos em cinco grupos de processos, sendo eles:

Grupos de processo de iniciação: que compreendem os processos que auxiliam a definir e iniciar o projeto.

Grupos de processo de planejamento: São os processos que definem quais tarefas, artefatos e necessidades devem avaliadas para definir como o projeto será executado.

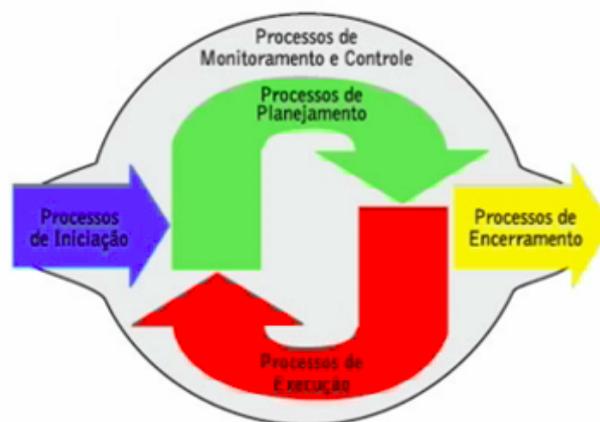
Grupos de processo de execução: Neste grupo estão os processos que são responsáveis pela execução do projeto, a fim de que os requisitos deste sejam satisfeitos.

Grupos de processo de monitoramento: São os processos utilizados para avaliar o projeto, avaliar sua situação, e iniciar mudanças necessárias, caso seja necessário.

Grupos de processo de encerramento: Aqui se encontram os processos que encerram o projeto, formalizando seu término com todas as partes envolvidas.

Os grupos de processo se inter-relacionam conforme a imagem abaixo:

Figura 2: Representação dos grupos de processos PMBoK



Fonte: Devmedia (2017)

O PMBoK define nove áreas de conhecimento, que compreendem os processos de gerenciamento de projeto, sendo elas:

Escopo: São os processos responsáveis por determinar todo o trabalho necessário para

a finalização do projeto.

Tempo: Processos que compõe a definição, planejamento e controle dos tempos das atividades contidas no do projeto, incluindo processos de definição das atividades envolvidas, o sequenciamento destas atividades, determinando o caminho crítico do projeto. Estimar os recursos de esforço, tempo e capital envolvidos para a execução destas atividades. E finalmente controlar o cronograma e *baseline* do projeto.

Custos: Esta área contempla os processos envolvidos nas estimavas, *baselines*, orçamento, controle e avaliação da realização dos custos ao longo do seu desenvolvimento.

Qualidade: Processos que determinam e controlam as políticas, os objetivos e as responsabilidades sobre os parâmetros de satisfação sobre o projeto, bem como o planejamento, a realização e o controle da qualidade do projeto.

Recursos Humanos: Consiste em identificar quais habilidades e funções se fazem necessárias para a execução do projeto, bem como a mobilização e a validação da disponibilidade do pessoal envolvido. Desenvolver, caso necessário as capacitações, acompanhar e monitorar o desenvolvimento dos membros da equipe.

Comunicações: Processos que asseguram que as informações do projeto sejam distribuídas a quem seja necessário. Identificar as partes interessadas, planejar como e em qual periodicidade estas informações são distribuídas. Gerenciar as expectativas das necessidades de informações relacionadas as partes interessadas. E informar o desempenho de projeto a quem interessa e necessita desta informação.

Riscos: Nesta área agrupam-se os processos responsáveis por identificar os riscos envolvidos no projeto, analisar o real impacto e qual a medida quantitativa e qualitativa destes para o projeto, planejar quais as melhores respostas para os riscos identificados e monitorar os riscos para a aplicação das soluções desenvolvidas.

Aquisições: São os processos necessários para adquirir produtos, serviços ou resultados (terceirizações). Envolve as ações do planejamento e identificação de quais itens devem ser adquiridos. Planejar estas aquisições tanto na dimensão do tempo de projeto,

quanto no escopo dos custos envolvidos.

Integração: Compreende os processos envolvidos para realizar o gerenciamento do projeto, pois envolve os processos que se responsabilizam pela formalização da abertura do projeto, do desenvolvimento do plano de projeto; orientar e gerenciar a execução deste mesmo projeto; controlar e monitorar as mudanças do projeto e, por fim, realizar a formalização do projeto. Ou seja, é a área do conhecimento que permeia todo o gerenciamento de projeto, certificando-se de sua iniciação, execução e encerramento completo.

As referidas áreas de conhecimento podem ser bem observadas na matriz apresentada abaixo:

Figura 3: Matriz de processos do PMBoK

© 2010, Márcio d'Ávila

	Iniciação	Planejamento	Execução	Controle	Encerramento
Escopo		Coletar requisitos. Definir escopo. Criar EAP		Verificar e controlar escopo	
Tempo		Definir atividades. Estimar sua sequência, duração e recursos. Criar cronograma		Controlar cronograma	
Custos		Estimar custos. Definir orçamento		Controlar custos	
Qualidade		Planejar qualidade	Realizar garantia da qualidade	Controlar qualidade	
Recursos Humanos		Planejar RH	Mobilizar, desenvolver e gerenciar equipe		
Aquisições		Planejar aquisições	Conduzir aquisições	Administrar aquisições	Encerrar aquisições
Comunicações	Identificar partes interessadas	Planejar comunicações	Distribuir informações. Gerenciar expectativas das partes interessadas	Reportar desempenho	
Riscos		Identificar riscos. Planejar sua gestão e resposta. Analisar quali- e quantitativamente.		Monitorar e controlar riscos	
Integração	Desenvolver TAP	Desenvolver plano de gerenciamento do projeto	Orientar e gerenciar a execução	Monitorar e controlar trabalho e mudanças	Encerrar projeto ou fase

Fonte: D'Ávila (2015)

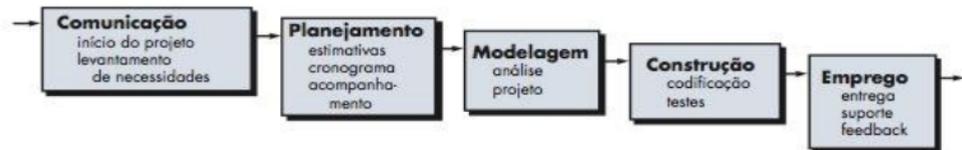
1.1. MODELOS DE PROCESSOS PRESCRITIVOS OU CLÁSSICOS

Para Pressman (2011), os métodos prescritivos, também chamados de clássicos, geram um trabalho de organização, fornecendo um roteiro claro e direto para a execução do trabalho dentro da engenharia e desenvolvimento de software. Os modelos sequenciais, cascata e em V são os modelos mais antigos.

O desenvolvimento de *software* apoiou-se nas fontes das demais engenharias para criação de um processo de planejamento e execução de projeto. Para Bassi Filho(2008), o

modelo cascata foi o primeiro a ser utilizado dentro da engenharia de *software*. Neste modelo o processo de desenvolvimento, desde a concepção até a entrega e finalização do projeto, é seguido de forma linear, conforme indica imagem abaixo.

Figura 4: Representação do modelo Cascata



Fonte: Pressman (2011)

Neste modelo, é necessário saber, desde a concepção do projeto, todos os requisitos necessários para sua entrega. De posse destas informações os analistas terão em mão os dados necessários para realizar o planejamento e análise do *software*, e a partir destas gerar os artefatos adequados para execução, desenvolvimento e garantia de qualidade do *software*.

Entretanto, Pressman alerta que:

“Os modelos de processo sequenciais, tais como o de cascata e o modelo V, são os paradigmas da engenharia de software mais antigos. Eles sugerem um fluxo de processo linear que, frequentemente, é inadequado para considerar as características dos sistemas modernos...” Pressman (2011 p.78)

2.3. METODOLOGIAS ÁGEIS

Metodologias ágeis são um conjunto de técnicas e processos baseados nos doze princípios do manifesto ágil. Segundo Oliveira (2003), os dois principais motivos que levaram ao desenvolvimento de metodologias ágeis foram:

- Alta frequência com que os projetos de software deixavam de cumprir seus cronogramas e extrapolavam seus orçamentos;
- Dificuldade de uso das metodologias pesadas.

As metodologias ágeis foram desenvolvidas para tornar mais eficaz a execução de projetos de *software*, principalmente em equipes de pequeno e médio porte. Para Bassi Filho (2008), quando o processo de desenvolvimento de *software* começou a tomar forma, baseou-se nos métodos tradicionais, formando assim um padrão de processo. Porém observou-se também um grande percentual de fracassos nos projetos de *software*.

Alguns profissionais, para evitar o referido fracasso, começaram a desenvolver e adotar métodos e técnicas opostos aos paradigmas tradicionais, no intuito de auxiliar o desenvolvimento. Assim, encontraram nestas técnicas, ferramentas que se mostraram mais eficientes, as quais acabaram por ser aprimoradas ao longo de vários projetos, retirando certos formalismos do processo, evitando burocracia e documentação excessivas.

Um grupo destes profissionais reuniu-se em 2001 com o objetivo de desenvolver uma nova metodologia, composta destas novas ideias. Após dois dias de discussões, concluíram que o desenvolvimento de *software* é muito complexo para ser definido por um único processo. Ao invés disso chegaram ao consenso de que existem alguns princípios importantes para obter bons resultados. E assim assinaram um manifesto, ao qual deram o nome de Manifesto Ágil, e publicaram seus doze princípios conforme exposto abaixo:

Manifesto Ágil:

Indivíduos e interações 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

Princípios:

- Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de *software* de valor.
- Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
- Entregar *software* funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
- Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.
- Construir projetos ao redor de indivíduos motivados, dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
- O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
- *Software* funcional é a medida primária de progresso.
- Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
- Contínua atenção à excelência técnica e bom *design*, aumenta a agilidade.
- Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
- As melhores arquiteturas, requisitos e *designs* emergem de times auto-organizáveis.
- Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

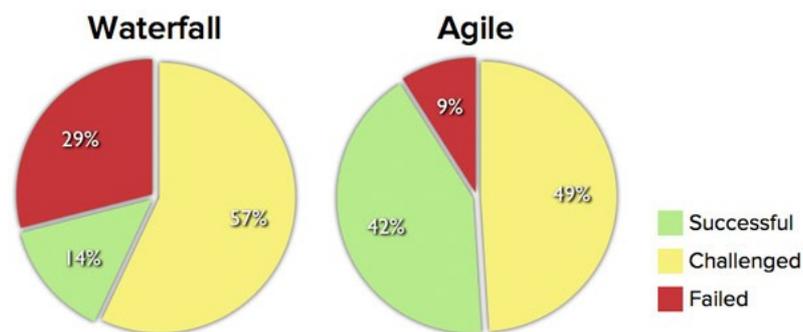
Observa-se, que o ágil não nega a existência, ou mesmo a necessidade de ferramentas,

processos, documentação, contratos, ou o plano. Apenas dá mais valor, àquilo que traz retorno de investimento ao *stakeholder*, àquilo que ele tem necessidade de retorno com o *software* a ser desenvolvido: A interação dos indivíduos envolvidos, colaboração entre cliente e equipe de desenvolvimento e respostas as mudanças para que haja a maior quantidade de *software* funcional, no menor tempo possível.

A partir deste manifesto e de seus princípios, desenvolveram-se várias metodologias, técnicas para lidar com o desenvolvimento de *software*, a estas técnicas se dá o nome de metodologias ágeis.

Pôde ser observado uma adoção das metodologias ágeis para o desenvolvimento de *software* em várias partes do mundo. Visto como uma solução para os problemas encontrados nos projetos. Esta mudança de paradigma tem sido vista com bons olhos pelas empresas, e mostrado resultado, conforme a pesquisa realizada pelo *Standish Group* em 2012, ver figura abaixo. FONTE: Cohn2012

Figura 6: Pesquisa do Standish Group sobre sucesso de projetos versus metodologia aplicada



Source: The CHAOS Manifesto, The Standish Group, 2012.

Fonte: Cohn (2012)

Através deste estudo, pode-se concluir que a utilização destas metodologias, realmente traz algum benefício para o desenvolvimento de software, uma vez que houve uma diferença significativa entre as taxas de sucesso das duas visões.

Entre as várias metodologias ágeis podemos citar como alguns dos expoentes do movimento ágil, o *Extreme Programming*, que é focado no desenvolvimento de *software*, e o *Scrum* que é um *framework* de desenvolvimento de projetos.

2.4. SCRUM

Scrum é um *framework* de metodologia de trabalho. Dizemos que é um *framework* e não uma metodologia, pois ele não diz como deve ser feito o trabalho, ou processo dentro dele. O *Scrum* indica a utilização reuniões, melhores práticas, artefatos e papéis dos participantes. Dentro destas reuniões e do processo de desenvolvimento do *software*, quaisquer ferramentas, técnicas ou boas práticas de desenvolvimento podem, e devem, ser utilizadas para assegurar a entrega, com qualidade das funcionalidades do *software*.

O *Scrum*, tem como origem as ideias de desenvolvimento de produto, vindas de Nonaka e Takeuchi.

“Nesse artigo, intitulado ‘The new new product development game(‘O novo novo jogo do desenvolvimento de produtos’), Takeuchi e Nonaka descreveram uma abordagem holística, na qual equipes de projetos são compostas de pequenas equipes multifuncionais, trabalhando com sucesso rumo a um objetivo comum, que os autores compararam à formação Scrum do Rugby” (PHAN; PHAN, 2011, p.41)

Segundo Oliveira, Freitas e Cunha (2014), o nome *Scrum*, vem de um movimento do jogo Rugby, no qual todo o time trabalha de forma aglomerada para transportar a bola pelo campo, e, com isso, atingir o objetivo do jogo. Esta definição do nome define um dos pilares do *Scrum*, ele é uma metodologia que depende e precisa de uma equipe para que funcione. Não há espaço para heróis. Ou o trabalho é feito pela equipe, e esta obtém sucesso, ou a equipe falha em realizar a entrega do projeto.

2.4.1. Sprint

A *sprint* é uma *timebox* fechada, ou seja, com um tamanho de tempo definido, e dentro desta *timebox*, o desenvolvimento das funcionalidades a serem entregues serão realizadas. O objetivo da *sprint* é ter ao final dela, uma nova parte do produto entregável ao cliente.

Geralmente, se usa 1 semana ou 2 semanas como tempo padrão para execução do

trabalho. A seleção deste tempo tem impactos na quantidade de trabalho entregue, por *sprint*.

”O time trabalha isoladamente por um período fixo de tempo chamado de Sprint. Ao final do período, o time se compromete a entregar um incremento no produto atual, a partir de uma lista de itens que contém os seus requisitos, e possivelmente algo mais[...]” Oliveira (2002, p. 27)

Caso todas as tarefas estimadas sejam terminadas até o fim da *sprint*, considera-se que ela foi bem-sucedida, e esta será avaliada pelo *Product Owner* (PO) na reunião de revisão.

No caso da estimativa de trabalho ter subestimado as tarefas da *sprint*, cabe ao time, com apoio do *ScrumMaster*, escolher entre, realizar melhoras nas tarefas já realizadas, como incrementar os testes automatizados, melhorias de arquitetura do sistema, ou mesmo tirar o tempo para estudo dos próximos passos, ou descansar.

Contudo, havendo problemas, eventuais impedimentos, tarefas subestimadas, ou quaisquer outros tipos de obstáculos e a *sprint* atrasar, caberá ao *ScrumMaster* comunicar ao PO os problemas, e cancelar a *sprint*. Para que seja criada uma nova, realizar uma nova reunião de *sprint* e avaliar os problemas encontrados. As tarefas que não forem completadas voltam para o *Product Backlog*, são re-priorizadas pelo PO e executadas pela equipe, quando a prioridade da tarefa for alcançada.

2.4.2. Artefatos

Product Backlog

O *Product Backlog* é uma lista dos requisitos e funcionalidades, que são de responsabilidade do PO priorizar com o Retorno de Investimento dos *Stakeholders* do projeto, pois o desenvolvimento das funcionalidades, baseadas nos requisitos será realizada na ordem definida pelo PO.

Para Oliveira (2003), as tarefas mais prioritárias, por se localizarem no início da fila a serem executadas, são as que estão sendo mais detalhadas no momento. Portanto tem um grau menor de incerteza, associado ao seu desenvolvimento.

Sprint Backlog

O *Sprint Backlog*, consiste na lista de requisitos que devem ser desenvolvidos na *sprint* que está sendo trabalhada. Estes requisitos ganham um maior detalhamento durante a reunião de *sprint*, durante a qual são divididos em tarefas e distribuídos pela equipe.

As tarefas que foram comprometidas ao final da reunião de *sprint* não devem ser alteradas. Quaisquer mudanças de escopo necessárias, devem aguardar uma nova *sprint*.

2.4.3. Papéis

O *Scrum* possui três papéis, que divide os indivíduos que participam do projeto, utilizando este *framework*. Estes papéis formam, as partes comprometidas com o projeto, que serão responsáveis para trazer o sucesso para o projeto. São eles:

ScrumMaster

Segundo Cohn (2010), este papel tem como responsabilidade manter o processo de desenvolvimento funcional, retirar impedimentos de desenvolvimento da equipe. O SM é responsável por manter o foco do time em alcançar os objetivos da *sprint* e do projeto. O SM responsável também por manter o processo do *Scrum*, mediar reuniões, mediar situações de dúvidas ou problemas entre membros da equipe.

Ainda, segundo o autor supracitado, são características de um bom *ScrumMaster*:

- Responsável
- Humilde
- Colaborativo
- Comprometido
- Poder de Influência
- Conhecimento

Carvalho; Mello (2009) enuncia que o *ScrumMaster* é responsável por cuidar de quaisquer questões necessárias para que a equipe possa se focar apenas no desenvolvimento do software.

Product Owner

O *Product Owner*, ou PO, é o representante dos *Stakeholders* do projeto, é da responsabilidade dele trazer a opinião do cliente, priorizar as tarefas que devem ser executadas, manter o *Product Backlog* atualizado e priorizado. Ele deve ser o responsável por trazer as opiniões dos *Stakeholders* como uma única e coerente voz e valorizar sempre o retorno de investimento para o projeto e o cliente.

Para Pichler (2010), são consideradas características desejáveis para um *Product Owner*:

- Visionário
- Executor
- Líder
- Trabalhar em equipe
- Comunicador
- Negociador
- Ter autoridade
- Disponível
- Qualificado

Time

O time de *Scrum*, é multifuncional, dentro dele devem conter as pessoas e as habilidades suficientes para execução do projeto. Este time deve ser auto-gerenciável.

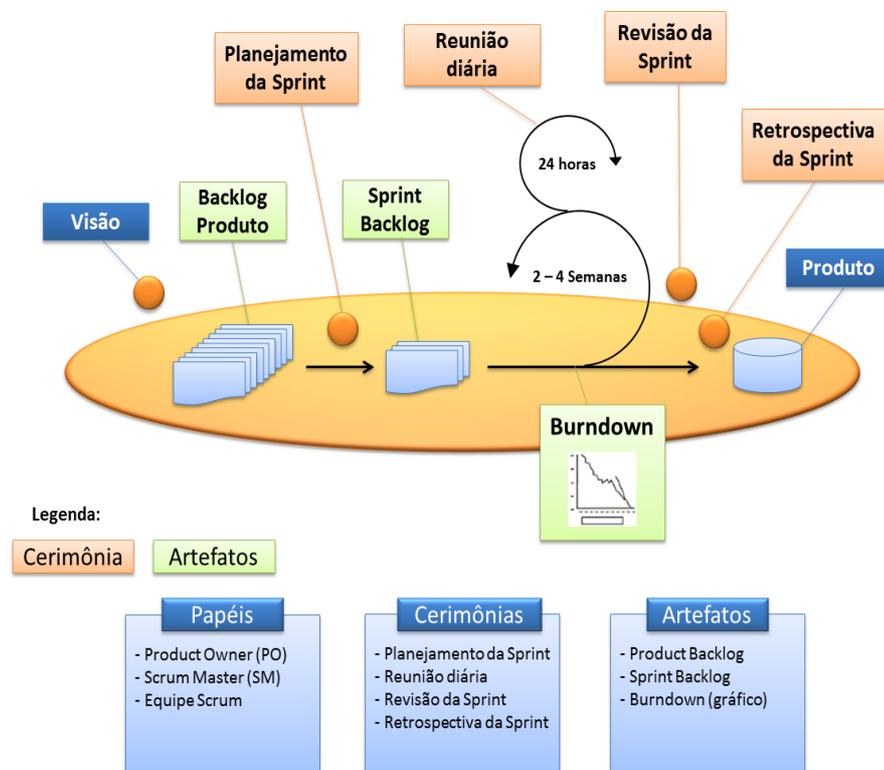
“[...]deve ser multifuncional e auto-suficiente, o que significa que seus membros devem reunir todas as habilidades necessárias para atingir o objetivo, assumindo compromissos e responsabilidades em relação ao projeto.” Bassi Filho (2008 p.40).

Um time de *Scrum* possui uma organização horizontal, todos são igualmente responsáveis pelo sucesso do desenvolvimento do projeto.

2.4.4. Reuniões

O *Scrum* possui quatro reuniões principais, que guiam o processo de desenvolvimento. Estas reuniões, possuem entradas de artefatos e de informações que geram saídas na forma de artefatos, conforme demonstrado abaixo.

Figura 7: Representação do Scrum, com suas reuniões, papéis e artefatos



Fonte: Ettinger (2011)

Reunião de *Sprint*

Objetivo: Definir o *Sprint Backlog* e objetivo da *sprint*;

Participantes: *ScrumMaster*, Time e PO;

Entradas: *Backlog* Priorizado, opiniões especializadas dos participantes;

Saídas: *Sprint Backlog* Priorizados, tarefas a serem executadas definidas e designação de qual integrante do time realizará qual tarefa.

O partir do *Product Backlog* Priorizado, deve ser selecionado um conjunto de tarefas, em que o time se comprometa na entrega até o término da *sprint*.

Durante esta reunião a equipe observa os itens mais prioritários selecionados pelo PO para serem desenvolvidos. Faz uma análise profunda, tirando dúvidas, determinando quais são as tarefas necessárias, para dar como prontos cada um dos itens que serão contemplados na *sprint*.

É de responsabilidade da equipe, determinar quantas tarefas ela deve se comprometer para a *sprint*, levando em conta o tempo e a complexidade das tarefas que serão selecionadas. É também a equipe que determina qual integrante fará qual tarefa, podendo promover assim um rotacionamento de informações e conhecimento do desenvolvimento.

Para Bassi Filho (2008), ao final da reunião de *sprint*, deve haver um *Sprint Backlog*, cujas tarefas devem refletir a meta proposta pelo PO e estar em conformidade com a capacidade de desenvolvimento da equipe.

Reunião diária

Objetivo: Monitoramento da equipe e do andamento da *sprint*;

Participantes: Time e *ScrumMaster*;

Entradas: Resposta às perguntas “O que fiz desde a última reunião?”, “O que farei até a próxima reunião?” e “Quais os impedimentos que tive?”;

Saídas: Lista de impedimentos a serem resolvidos; status do projeto/*sprint*;

Nesta reunião os integrantes da equipe permanecem em pé para responder, cada um, às três perguntas citadas acima. O objetivo é identificar possíveis problemas, o andamento do projeto, informar aos demais participantes em quais partes do projeto estão mexendo. Facilita, assim, a troca de informações pertinentes ou o conhecimento de quais são problemas de

avanço nas tarefas em desenvolvimento.

Oliveira (Oliveira 2003, p.27) afirma que “[...]Através do acompanhamento diário da reunião de acompanhamento, o time e o Scrum Master podem compreender o status atual do projeto e quais são as perspectivas de andamento[...]”.

Neste ponto, algumas observações sobre esta reunião são necessárias. Não se deve aguardar pela reunião diária para informar sobre um impedimento, mas é pertinente que se informe dos impedimentos que ocorreram, para que sua resolução seja disseminada pelo grupo, aumentando a qualidade de tempo de resposta para soluções da equipe.

Esta reunião deve ser feita, preferencialmente, em um lugar de fácil acesso, para que quaisquer pessoas da organização possam observá-la. Isso aumenta e fomenta a transparência do projeto. Porém, somente os integrantes da equipe podem interagir na reunião.

Reunião de Revisão/*Review*

Objetivo: Demonstrar ao PO o andamento da *sprint* e do Projeto;

Participantes: Equipe, *ScrumMaster*, *Product Owner* e convidados do PO;

Entradas: Lista das tarefas finalizadas da *sprint*; demonstração das tarefas; lista das tarefas não executadas da *sprint*;

Saídas: Lista de tarefas dadas como finalizadas; lista de melhorias a serem incorporadas ao *Product Backlog*.

Nesta reunião o time apresenta ao PO, o resultado da *sprint*. Demonstra em uma demo ou ambiente de homologação, as funcionalidades desenvolvidas, apresenta o funcionamento destas e como foram executadas e também quais os problemas encontrados durante a execução do desenvolvimento.

Neste momento, o PO, pode questionar as tarefas, e avaliar o impacto destas no objetivo do projeto. E, ainda, identificar melhorias, ou mesmo identificar, em um pior caso, que a tarefa não atende as necessidades do cliente.

É de prerrogativa do PO determinar se a *sprint* alcançou ao objetivo.

Reunião de Retrospectiva

Objetivo: Identificar melhorias no processo de desenvolvimento;

Participantes: Equipe, *ScrumMaster*, convidados da equipe;

Entradas: Lista de pontos positivos e negativos ocorridos durante a *sprint*;

Saídas: Lista de ações de melhoria e responsáveis para serem executadas.

Este é o momento em que a equipe olha para a *sprint*. Observa o que foi feito durante este tempo, o que foi positivo e deve ser mantido, e o que foi negativo e deve ser mudado. E como mudar para melhorar.

Podem ser discutidos pela equipe os mais variados tópicos. Sempre com o propósito de melhorar o processo de desenvolvimento, observar gargalos e impedimentos crônicos.

Oliveira (2008) nos traz que, o resultado da retrospectiva são três listas:

1. Ações que deram um resultado positivo e devem ser mantidas;
2. Ações que não apresentam resultado, ou geram problemas e devem ser retiradas do processo;
3. Novas práticas a serem implementadas para corrigir os problemas do item dois, ou realizar melhorias no processo.

O *Scrum* é um conjunto de boas práticas focadas no Manifesto Ágil, no qual a cada *sprint* (uma a duas semanas) gera-se um resultado para o produto e executa-se um processo de melhoria contínua do próprio processo. Para Bissi (2007), esses são alguns dos benefícios obtidos com o uso do *Scrum*:

- Diminuição dos riscos;
- Maior integração entre os membros das equipes;
- Rápida solução de problemas;
- Progresso medido continuamente;

- Os clientes se tornam parte da equipe de desenvolvimento;
- Entregas frequentes de funcionalidades;
- Discussões diárias de *status* com a equipe;

2.5. PROCEDIMENTOS METODOLÓGICOS

Para a execução deste projeto de criação de um jogo para o auxílio do ensino do *Scrum*, foram necessários determinar certos passos para que o processo não se torne puramente empírico e possa ter algum valor científico.

Entre os passos deste processo, foi realizada uma revisão sistemática de jogos já existentes para o ensino deste *framework*. Com base nesta busca realizou-se uma avaliação dos pontos positivos e pontos a melhorar destes artefatos. A partir desta avaliação, fez-se a prototipação de um jogo que possa servir como instrumento para o ensino do *Scrum*. Ao finalizar a prototipação, o jogo foi avaliado através de um questionário, para dimensionar as consequências, do uso do jogo nos participantes.

2.6. REVISÃO SISTEMÁTICA

Revisão sistemática é um método para pesquisa e avaliação de um determinado conteúdo, dando um embasamento metodológico científico sobre a forma de pesquisa, diminuindo a probabilidade de que a pesquisa se realize a partir tão somente do bom-senso do pesquisador. Apresentando ao final do processo (Sampaio 2007, p. 84). “[...]um resumo das evidências relacionadas a uma estratégia de intervenção específica, mediante a aplicação de métodos explícitos e sistematizados[...]”.

A revisão sistemática possui alguns passos que devem ser seguidos para uma correta execução de sua metodologia. Segundo Galvão, Sawada e Trevisan (2004) podemos elencar entre estes passos:

Definição de uma pergunta ou objetivo – Nesta fase determina-se quais os termos as serem buscados. A utilização de um termo adequado influencia nos resultados que serão

obtidos. Isso porque, se o termo buscado for demasiado abrangente, os resultados da pesquisa serão extremamente numerosos e podem ter pouca aderência ao tema. Contudo, uma busca muito objetiva, pode vir a retornar poucos resultados para uma avaliação dos resultados. Por isso se deve avaliar e testar.

Busca dos estudos – Aqui são selecionadas quais as fontes de busca a serem utilizadas para os termos selecionados, podendo-se escolher diversas fontes, ou mesmo restringi-las, dependendo dos objetivos da busca. Entre as fontes selecionadas, pode-se citar bibliotecas físicas, motores de busca eletrônicos, periódicos entre outros.

Seleção dos estudos – Ao aplicar o termo selecionado dentro da seleção de fontes de busca obter-se-ão vários artigos e trabalhos que atenderão ao termo, porém nem todos serão do interesse do pesquisador. Devem ser colocados, portanto, critérios de exclusão separando-se parte dos itens recuperados na busca que não serão incluídos para avaliação do pesquisador.

Avaliação crítica – Neste ponto, o pesquisador avalia os resultados obtidos. Averigua se os resultados e métodos obtidos na peça em questão são válidos, para continuar dentro da pesquisa.

Síntese – Com base nos artigos coletados e avaliados nas fases anteriores, o pesquisador cria uma conclusão com as informações encontradas.

2.7. PROTOTIPAÇÃO RÁPIDA

Para o desenvolvimento do jogo, será utilizada a metodologia de prototipação rápida. Para Duarte (2012), a prototipação rápida tem como objetivo a criação de um modelo utilizável do jogo que está em fase de projeto, a partir de materiais de fácil acesso, como papel, tesoura, cartolina, partes feitas a mão, ou mesmo emprestadas de outros jogos. Gera um protótipo simples mas jogável de um jogo físico para ser utilizado para simular a metodologia de desenvolvimento de software focada no ágil e no *Scrum*. Sobre a prototipação rápida:

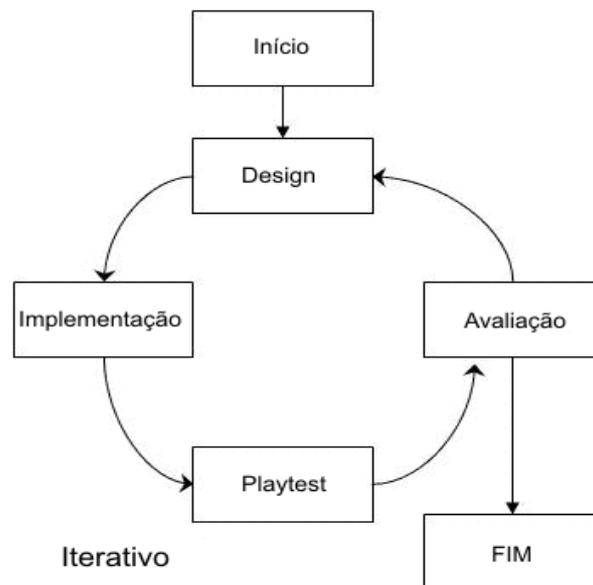
“Além do baixo custo de implementação, um jogo de tabuleiro pode ser modificado imediatamente, mesmo em meio a uma partida, em decorrência da identificação de um problema ou da concepção de uma ideia nova.” Duarte(2012, p. 135).

A Prototipação rápida vem do *design* de interação, possuindo quatro etapas: estabelecer requisitos; criar alternativas de design; prototipar e avaliar o resultado. Para Domingues et Al (2014), estas etapas são iterativas e contínuas, focadas na experiência do usuário, ou seja, o foco é trazer resultado para quem o utilizará.

Após definidas as mecânicas de jogo, suas regras, e elemento, cria-se o material definitivo com a arte e qualidade finais. Enquanto os testes ocorrem, trocar, ou mesmo produzir elementos é muito mais fácil com papel e caneta.

Após algumas iterações, identifica-se melhores regras e formatos de jogo, criar-se-á uma aplicação jogável e com uma melhor apresentação, para uma produção piloto.

Figura 8: Figura exemplificando o modelo iterativo de prototipação rápida



Fonte: Schreiber (2009)

2.8. AVALIAÇÃO PELO MODELO MEEGA+

Com o objetivo de realizar uma avaliação do jogo, como uma ferramenta de apoio pedagógico, utilizaremos o modelo MEEGA+. O modelo MEEGA+ vem a ser uma melhoria a partir do modelo MEEGA - *Model of Evaluation of Educational Games* (Modelo de Avaliação de Jogos Educacionais).

Petri, Von Wangehein e Borgatto (2017) trazem este modelo como uma padronização para avaliar o jogo após a utilização deste por estudantes. Avaliando o jogo em três dimensões principais: motivação, experiência do usuário e a aprendizagem. Esta avaliação se dá através de um questionário com perguntas quantitativas utilizando da abordagem GQM - Goal, Question, Metric (Objetivo, Questão e Métrica).

“O objetivo do modelo MEEGA+ é: analisar jogos educacionais com o propósito de avaliar a percepção da qualidade em termos de experiência do jogador e percepção da aprendizagem do ponto de vista de alunos e instrutores no contexto de cursos superiores da área de computação.” (Petri, Von Wangehein e Borgatto 2017, p. 4).

Petri, Von Wangehein e Borgatto (2017) nos trazem que o MEEGA+ utiliza como tipo de resposta baseado na escala de Likert, utilizando cinco possíveis respostas: 1. “Concordo totalmente”, 2. “Concordo”, 3. “Não concordo e nem discordo”, 4. “Discordo” e 5. “Discordo totalmente”. O questionário recupera também dados demográficos dos alunos auxiliando, a perfilação dos alunos com relação aos resultados de suas respostas. Conforme na figura abaixo exibe um exemplo do questionário e um exemplo de resultado avaliado pelo modelo.

Figura 9: Figura exemplificando questionário e relatórios sobre o MEEGA+



Fonte: SOFTWARE QUALITY GROUP – GQS (2017)

O *Software Quality Group* mantém uma página, onde o modelo pode ser obtido gratuitamente, sob a licença *Creative Commons*, onde disponibilizam, o questionário padrão,

um manual de instruções do questionário e uma planilha de avaliação de dados. Este material ser utilizado para avaliação de jogos educacionais digitais e não-digitais.

Para Petri, Von Wangehein e Borgatto (2017) o modelo avalia o jogo em duas dimensões principais:

A) Experiência do jogador: compreendida por categorias como: usabilidade, confiança, desafio, satisfação, interação social, diversão, atenção focada e relevância.

B) Percepção da aprendizagem: compreendida por categorias como: aprendizagem de curto prazo e Objetivos de aprendizagem.

3. DESENVOLVIMENTO DO TRABALHO

3.1. REVISÃO SISTEMÁTICA

A fim de iniciar a criação de um jogo adequado para a fixação dos conceitos do *Scrum*, foram avaliados três jogos já existentes, os quais têm a mesma finalidade e encontram-se disponíveis na internet, gratuitamente.

Para a presente revisão, utilizou-se como motores de busca o buscador genérico google.com, e o buscador específico para registros acadêmicos scholar.google.com, no período compreendido entre os dias 20 de Setembro de 2017 e 30 de Setembro de 2017.

Foram pesquisados os seguintes termos de busca: *Scrum board game*; *board game Scrum*; *Scrum Game*. Avaliados os retornos para ambos os buscadores, restaram selecionados os jogos *Scrum Card Game*, *Scrumble* e *Lego City* para avaliação da revisão sistemática dos jogos já utilizados.

3.2. SCRUM CARD GAME

Autor: Timofey Yevgrashyn

Link para download: <https://scrumcardgame.com/scrum-card-game-rules/>

O jogo *Scrum Card Game* tem a proposta de trazer uma iteração curta, de três *sprints*,

exibindo as fases, as ferramentas, e as possibilidades do *Scrum*. Para ser compreendido corretamente, é necessário conhecimento teórico prévio dos processos, reuniões e valores do *Scrum*.

Durante a partida, são utilizadas cartas de “*Stories*”, que devem ser finalizadas através de uma simulação do processo de desenvolvimento. Cada *story* possui uma descrição, ordem de priorização e quantidade de “horas” para sua execução. Utiliza-se um *flipchart* para desenhar um *kanban*, no qual as *Stories* da *sprint* em jogo são colocadas no *status* adequado.

Esta atividade é um jogo competitivo entre duas equipes, no qual atribui-se a vitória à equipe que colocar como realizadas mais *Stories*, ou seja, for mais produtiva. Cada equipe deve ter até 6 jogadores. O jogo deve ser ministrado por um mediador/treinador, que explicará as regras e tirará quaisquer dúvidas, ou explicará algum processo.

No início do jogo, cada equipe deverá estar com um *flipchart* ou representação de um *kanban* com as seguintes colunas: A fazer, Fazendo, Realizado.

Para realizar o trabalho da *story*, ou seja, para que ela seja dada como realizada, um jogador lança dois dados de 6 lados; somam-se os resultados e o valor obtido é deduzido da quantidade de horas inscrita na *Story*. Será tida como realizada uma *Story* que contar com zero ou menos horas a serem realizadas e não tiver nenhuma carta de impedimento sobre ela.

Exemplificando: O jogador 1 determina que está trabalhando na *Story* #3, que possui 16 horas. Ele lança dois dados e obtém os resultados 3 e 5, cujo somatório é 8. Esse valor deverá ser diminuído da quantidade original de horas (16 horas). Desta forma, a tarefa foi iniciada e possui ainda 8 “horas” para ser concluída. E esta mecânica deverá ser repetida até que as horas da *story* sejam zeradas.

Outro elemento do jogo são as cartas de sorte (chance), que são divididas em 3 categorias:

- Eventos: Determinam um evento ocorrido e que deve ser aplicado ao jogo, podendo ser positivo – como uma resolução de uma tarefa pendente –, ou negativo – a tarefa a ser entregue precisará de mais horas para ser resolvida.

- Problemas/impedimentos: Determinam um problema que deve ser aplicado a uma *story*, impedindo que ela seja resolvida. Quando comprada por um jogador, esta carta deverá ser aplicada à *story* que ele está desenvolvendo.

- Soluções: São cartas que podem ser guardadas pelos jogadores para solucionar os problemas que estiverem em efeito impedindo que as *stories* sejam resolvidas

O jogo é iniciado com um planejamento de *sprint*, momento em que serão selecionadas pelos participantes quantas *Stories* realizarão. Ao final deste passo, começam a seguir-se os “dias” de jogo. Deve ser anotada a quantidade somada de horas de cada *story* como o planejado da *sprint*.

Cada iteração é composta de três dias. Em cada “dia”, cada um dos jogadores realiza as seguintes ações:

- Seleciona a tarefa a ser desenvolvida;
- Lança dois dados de seis lados;
- Compra uma carta de sorte.

Após três rodadas, é finalizada a *sprint* do jogo. Ao final de cada *sprint*, a quantidade de horas realizadas deverá ser anotada no *kanban*, abaixo do número inicialmente planejado. Realiza-se uma *review*, na qual são mostradas ao mediador as *stories* concluídas. Em seguida, realiza-se uma reunião de retrospectiva, na qual a equipe deverá ser incentivada a determinar formas de melhorar, se necessário, a estratégia de suas jogadas. O jogo termina ao final de três *sprints* (nove rodadas).

Após ler as instruções e jogar algumas partidas foram encontrados os seguintes pontos sobre o *Scrum Card Game*.

Pontos positivos:

- O jogo tem regras simples;
- É rápido para se compreender e jogar;

- Passa por todas as reuniões do *Scrum*;
- Utiliza o desenvolvimento de *Software* como tema do jogo;
- Os problemas que surgem no jogo são resolvidos por decisões dos próprios jogadores.

Pontos negativos:

- A resolução das *stories* é totalmente baseada em sorte: lançamento de dados e compra de cartas;
- Não faz uso dos papéis do *Scrum* com os participantes;
- Não há diferenciação de papéis ou tarefas entre os desenvolvedores. Cada *story* precisa apenas ser concluída, independentemente das fases de desenvolvimento de software (análise, desenvolvimento, testes etc);

3.3. SCRUMBLE

Autor Roman Trocherie,

Link <http://scrumble.pyxis-tech.com/>

Para dar início ao jogo, os jogadores são distribuídos nos papéis de: membros do time de desenvolvimento, que serão responsáveis por executar as tarefas para desenvolvimento do projeto em questão; Product Owner (PO), que será o responsável por determinar quais funcionalidades deverão ser executadas, e com qual prioridade; e, por fim, o ScrumMaster (SM), o qual deve ser uma pessoa que já tenha familiaridade com o *Scrum*, pois ele será o guia do jogo, tirando dúvidas, auxiliando em pontos de controvérsia entre os participantes.

Uma lista de funcionalidades é dada ao PO para que ele selecione quais tarefas serão executadas. Ele poderá ter o auxílio dos demais participantes, porém a palavra final sobre quais tarefas serão executadas será dele.

Cada carta de funcionalidade possui um identificador numérico único e informação da precedência de tarefas a qual, caso exista, informa quais funcionalidades devem estar finalizadas para a completude da tarefa em questão. Na descrição da funcionalidade também

aparece a quantidade de “pontos de valor”, que representam a satisfação do cliente em ter a funcionalidade desenvolvida e entregue.

O jogo contém um conjunto de funcionalidades prontas para serem utilizadas, porém possui um conjunto de cartas em branco para que sejam colocadas tarefas de acordo com o contexto do grupo que está jogando. Desta forma, permite-se que o contexto do grupo de jogadores seja utilizado no jogo.

Uma vez que o PO apresenta as tarefas ao grupo de desenvolvimento, estas são estimadas pelos jogadores utilizando o método T-Shirt Size de estimativa, no qual a complexidade das tarefas é definida por tamanho (PP – P – M – G – GG).

O procedimento do jogo tem início determinando-se o que será feito em uma *sprint* de dez dias de jogo, sendo que, destes dias, um é reservado para as reuniões de retrospectiva, *review* e planejamento, resultando em nove dias de produção de tarefas.

É dado início ao *sprint planning* do jogo. Nesta fase são determinadas, a partir da complexidade das funcionalidades, do débito técnico atual do projeto e da quantidade de pessoas que estão jogando, quantas tarefas deverão ser executadas para uma funcionalidade ser dada como finalizada.

Calculado o número de tarefas da cada funcionalidade, é colocado um marcador no tabuleiro no círculo de tarefas (em amarelo), informando a quantidade de tarefas a serem executadas. E um marcador de início no dia 1, quantidade 0.

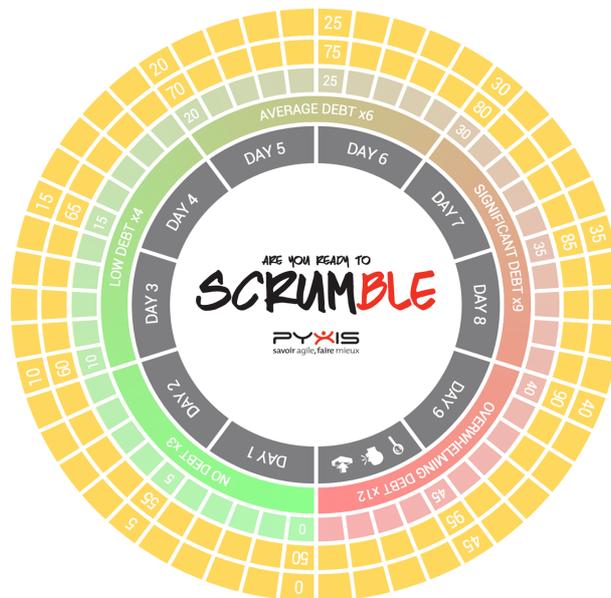
Para cada dia de jogo, cada jogador de desenvolvimento, rola um dado de 6 lados, e avança no tabuleiro (figura abaixo) a quantidade de casas correspondente ao valor lançado. Há dois casos especiais: quando tira 1, o jogador tem o direito de lançar o dado novamente; e, quando obtém 6, sorteia-se uma carta de problema, que terá um efeito negativo a ser aplicado. Cada dia também é sorteada uma carta com uma informação, positiva, neutra ou negativa que deve ser aplicada ao jogo.

O jogador que lança os dados têm duas opções de movimento, pode avançar na execução das tarefas, ou “pagar” o déficit técnico. O débito técnico representa problemas no

desenvolvimento – seja na forma de tarefas inacabadas ou de problemas acumulados na arquitetura do sistema que está sendo construído – e impacta diretamente na quantidade de tarefas a serem desenvolvidas para cada funcionalidade.

Ao final da *sprint* o PO avalia o que foi entregue e retira uma carta de “*Review*” para cada funcionalidade finalizada. O efeito da carta pode ser aplicado imediatamente ou durante o restante do jogo, de acordo com o que está descrito nesta. Para cada cinco tarefas não entregues na *sprint*, é gerado 1 ponto de déficit técnico a ser incorporado no tabuleiro.

Figura 10: Tabuleiro do Scrumble



Fonte: Trocherie (2014)

A quantidade total de *sprints* fica a critério do *ScrumMaster*. Deve-se levar em consideração que não existe um momento em que se declara vitória no jogo.

Pontos positivos:

- O jogo apresenta várias similaridades com a realidade, demonstradas pelos problemas que surgem no dia a dia;

- Mostra o impacto do débito técnico;
- Tarefas não realizadas causam impacto no andamento do jogo;
- Utiliza do contexto de desenvolvimento de software para o *story telling* do jogo;

Pontos negativos:

- Regras complexas para determinação das tarefas e impacto de débito técnico;
- Falta da participação dos jogadores nos papéis de PO e SM que não têm papel atuante;
- O PO tem poucas responsabilidades dentro do jogo;
- Focado no débito técnico e o impacto deste no projeto;
- Tempo de preparação pode ser longo, visto que o PO deve avaliar as tarefas e os jogadores têm que decidir a complexidade de cada funcionalidade;
- Todos participantes são de igual impacto no jogo, caso um tenha uma jogada ruim, o impacto é o mesmo para qualquer participante da equipe.

3.4. LEGO CITY

Autor: Alexey Krivitsky

Link: https://www.scrumalliance.org/system/resource_files/0000/3689/Scrum-Simulation-with-LEGO-Bricks-v2.0.pdf

O jogo tem como objetivo a elaboração de um projeto de cidade, no qual cada funcionalidade pode ser um prédio, casa ou mesmo uma rua, entroncamento. As equipes são divididas e avaliam os requisitos de cada prédio a ser construído.

O jogo possui como papéis ativos o *Product Owner* (PO), o *ScrumMaster* e o time de desenvolvimento.

O papel do *Product Owner* é desempenhado pelo ministrante do jogo, o qual servirá como “anfitrião” e explicará as regras.

Já o papel de *ScrumMaster* fica a cargo, quando houver, de jogadores mais experientes que participem da partida. Estes participantes são responsáveis em orientar os demais em como o jogo deve ser executado.

Por fim, o time de desenvolvimento será constituído pelos participantes que construirão a “cidade” a partir das especificações do *Product Owner*, de acordo com a priorização e o escopo por ele passados.

O tabuleiro utilizado representa um mapa com ruas e áreas para serem colocadas construções de prédios e outras estruturas.

Durante a preparação para o jogo, são selecionados os tipos de prédios a serem criados a partir de bloco de *legos*. Sugere-se que se tenha uma diversidade e quantidade suficiente para que os participantes possam ter algumas *sprints* para executar o trabalho.

Para a preparação, deve-se disponibilizar um “tabuleiro”, geralmente com uma ou mais folhas de *flipchart*, que crie um mapa da cidade, onde ficarão os prédios a serem construídos.

Antes do início, é sugerido que os participantes avaliem a estimativa de cada prédio a ser construído, baseado nas especificações dos mesmos. Para o início de cada *sprint*, cada time deve estimar quantos prédios construirão na primeira *sprint*, e deixar esta informação anotada.

São dados três minutos para o planejamento da *sprint* a ser executada, durante os quais as tarefas são estimadas, e sua execução planejada, quais materiais serão usados, onde no mapa será colocado cada um dos itens etc.

É dado início a uma *sprint* de sete minutos para construção do que foi comprometido no planejamento. A *sprint* transcorre ininterruptamente durante este processo, um marcador com o tempo deve preferencialmente estar visível para os participantes

Após este tempo, são disponibilizados cinco minutos para a reunião de *Review*, na qual o *Product Owner* pergunta sobre as tarefas executadas, e faz observações sobre a qualidade das execuções.

Todas as tarefas não finalizadas são colocadas novamente no *backlog* de execução, podendo ser estimadas para o planejamento da próxima *sprint*. E então o ciclo tem início novamente.

É sugerido que se utilize de três *sprints* para a execução do jogo. Ao fim destas iterações, o anfitrião do jogo pergunta aos participantes as observações destes sobre o processo, sobre como foram feitas as estimativas, o quão acuradas estas foram, qual a importância do papel do *ScrumMaster* e do PO. Para assim os participantes possam externalizar e compreender como o *Scrum* pode funcionar a partir da observação de uma simulação.

Pontos positivos:

- Passa pelos pontos principais do *Scrum*;
- Incentiva a observação e execução de estimativas.

Pontos negativos:

- Não apresenta todas as reuniões do *Scrum*;
- Os papéis de *ScrumMaster* e *Product Owner* são realizados por participantes que já tenham alguma experiência.

Abaixo quadro comparativo entre os jogos avaliados e sua aderência ao *Scrum*:

Quadro 2 – Quadro comparativo dos jogos avaliados e as reuniões e papéis do <i>Scrum</i>			
Reunião	Scrum Card Game	Scrumble	Lego City
Sprint Planning	Presente	Presente	Presente
Reunião Diária	Presente	Presente	Não
Retrospectiva	Presente	Presente	Não
Review	Presente	Presente	Presente
Papeis			

ScrumMaster	Não aplicável	Presente, mas deve ser uma pessoa que já familiarizada com o framework	Presente, mas ou é representado pelo aplicador do jogo, ou por participantes experientes, auxiliando no processo
Product Owner	Representado pelo Jogo	Presente, mas somente atua em duas fases do jogo	Presente, porém na forma do aplicador do jogo
Time	Representado pelos participantes	Representado pelos participantes	Representado pelos participantes

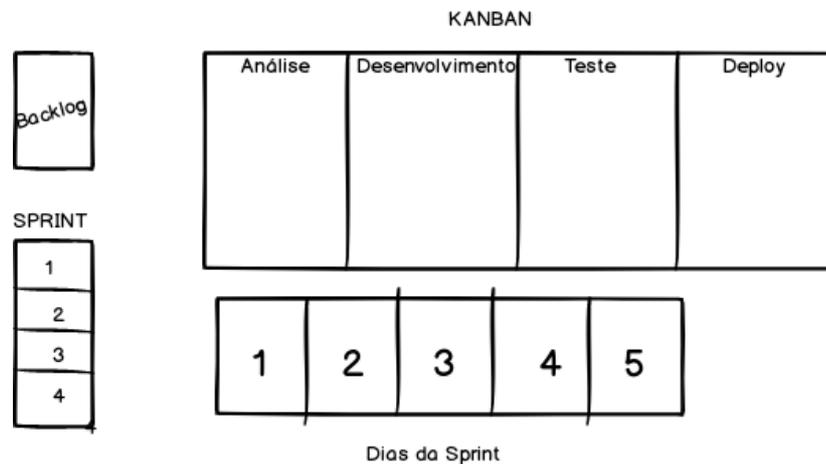
3.5. PROTOTIPAÇÃO DO JOGO

3.5.1. Iteração I

Resumo das regras

A primeira iteração compunha de um tabuleiro, conforme a figura abaixo, contendo um *kanban*, espaços para marcadores da *sprint* corrente, qual o dia da *sprint* atual e um espaço para uma pilha de cartas que representam o *backlog* a ser desenvolvido.

Figura 11: Representação de protótipo de baixa fidelidade para Iteração I



Fonte: Autor

Cada *story* do *backlog* contém apenas um número representando a dificuldade da tarefa a ser executada. Estes números vão de 1 a 13 seguindo a escala de Fibonacci (1, 2, 3, 5, 8, 13).

Cada jogador é representado por um peão genérico, que informa qual tarefa está sendo executada e em qual etapa do desenvolvimento na área do tabuleiro do *kanban*.

No início do jogo, as cartas são embaralhadas e colocadas sobre o monte, com as faces para baixo, para que não seja possível observar o valor da próxima carta. Os jogadores devem comprar tantas cartas, quando acharem necessário, preferencialmente o suficiente para que sejam executadas durante uma única *sprint*. Porém as tarefas devem ser executadas na ordem de compra, não sendo permitidas mudanças na ordem de execução.

A tarefa, para ser dada como concluída, deve passar pelas três fases de desenvolvimento exibidas no *kanban*: análise, desenvolvimento e teste.

Para passar cada tarefa de fase, é necessário que o jogador lance um dado de seis lados. O valor obtido é diminuído do valor da carta, caso chegue a zero ou menos a tarefa

pode passar para a próxima coluna.

De início, coloca-se um marcador na primeira *sprint*, e outro marcador no primeiro dia.

Início do turno se dá definindo pelos participantes quais peões, trabalharão em quais tarefas. Coloca-se o peão sobre a carta na área do *kanban* em que a carta está. Para passar para a próxima etapa, o jogador determina uma “alocação de trabalho”, representando a quantidade de esforço além do “normal” para cumprir com a tarefa em mãos.

Após tomada esta decisão, cada jogador lança um dado para obter a quantidade de trabalho executada. Subtrai-se do valor da carta de tarefa o valor obtido no dado. Caso a subtração resulte zero ou menos, passa para a próxima coluna, caso seja maior que um, a tarefa permanece na coluna e o resultado fica sendo o valor da carta para o próximo turno.

Quando todos os jogadores tiverem lançado seus dados, move-se o marcador de dia para o próximo dia. Esta operação se repete até alcançar o final dos dias (5). Neste momento, move-se o marcador de *sprint* para a próxima casa, e volta-se o marcador de dias para a primeira casa.

O jogo finaliza, ao final de quatro *sprints*, ou se a quantidade de cartas de tarefas forem todas movidas para a coluna *deploy*.

3.5.2. Iteração II

Baseado no *playtest* da primeira interação, foi considerado que uma melhoria, seria a especialização dos trabalhadores. Cada trabalhador tem um modificador relacionado a uma coluna do *kanban* (Análise, desenvolvimento e teste), conforme apresentado na figura 13. Cada jogador, no início do jogo sorteia ou escolhe uma persona, cada persona possui diferentes modificadores para cada fase de entrega da tarefa (Análise, desenvolvimento e teste).

Figura 12: Representação da carta de Trabalhador

Programador Pleno
Análise +0
Desenvolvimento +2
Teste +1

Fonte: Autor

Outra modificação colocada, foi que a tarefa teria uma dificuldade para cada fase de entrega, para passar para a próxima fase, a dificuldade de cada fase deve ser zerada até a tarefa ser considerada entregue conforme demonstrado na figura abaixo.

Figura 13: Protótipo de carta de tarefa

Tarefa 1		
Análise	Desenv	Teste
3	8	2

fonte: autor

Ao lançar o dado para consumir os pontos da tarefa, o modificador do trabalhador é somado ao resultado dos dados. Portanto, se o jogador estiver trabalhando em uma tarefa que

esteja em desenvolvimento obter um 6 nos dados, e estiver com um modificador (+2), uma tarefa de até 8 pontos é considerada pronta e passada para a próxima fase.

3.5.3. Iteração III

Nesta iteração, também foram adicionadas cartas de problema. A cada rodada, é retirada de uma pilha de cartas de problema para todo o grupo. Cada problema pode ser ou um *bug* ou um impedimento. A figura abaixo traz uma representação do protótipo da carta de *bug*.

Figura 14: Protótipo de Bug

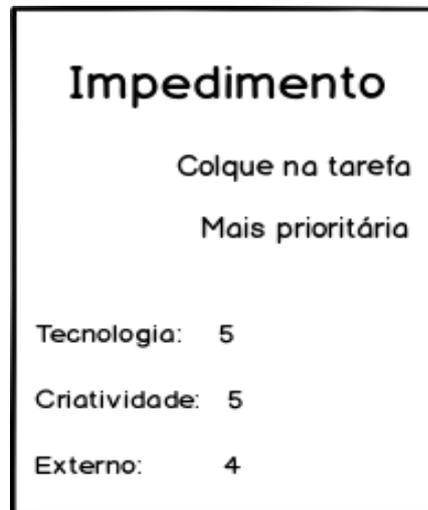
BUG		
Análise	Desenv	Teste
0	3	1

Fonte: autor

Quando a carta for um bug, deve ser tratado como uma nova tarefa, a ser colocada no *backlog* da *sprint*, e possui as mesmas características de uma tarefa. Sendo necessário a entrega do bug para resolução do problema.

As cartas de impedimentos são sorteadas, e no seu corpo informam quais tarefas em execução impedirão de serem finalizadas. Uma tarefa que tenha um impedimento sobre ela não pode ser finalizada até que o impedimento em questão seja removido pelas cartas de ação.

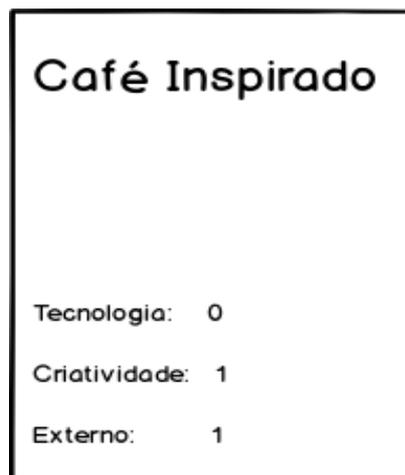
Figura 15: Protótipo exemplo de carta de Impedimento



Fonte: Autor

Outro tipo de carta adicionado foi o de “ação”, esta carta é comprada por cada um dos jogadores, cada carta possui três características. A cada característica é associado um valor.

Figura 16: Protótipo exemplo da carta de ação



Fonte: Autor

Para retirar um impedimento, agrupam-se as cartas de ação, e somam-se seus valores

até chegar ao valor do impedimento para cada valor (Tecnologia, Criatividade e Externo).

3.5.4. Iteração IV

Para equilibrar o jogo, foram colocadas cartas neutras para a ação de revés, para balancear o jogo. Para que não se sorteie problemas ou *bugs* a serem resolvidos em todas as rodadas.

Foram adicionados impedimentos que retornam tarefas da fase de desenvolvimento de testes para a fase de desenvolvimento, simulando a situação em que o *tester* encontra um problema no que foi desenvolvido.

4. SCRUMBOARD

4.1. ITENS DO JOGO

- 1 Tabuleiro de cartas e controle de dias e *sprint* (conforme anexo 1);
- 1 Tabuleiro de *Kanban* do processo de desenvolvimento;
- 5 Cartas de UFA;
- 25 Cartas de ação (Ctrl+C Ctrl +V, Rubber Duck, Ócio Criativo, Café com Inspiração e Stack Overflow);
- 6 Cartas de melhoria da equipe;
- 15 Cartas com tarefas do Projeto;
- 22 Cartas com tarefas do projeto e-commerce;
- 1 Carta de definição do projeto, contendo as necessidades do cliente;
- 28 Cartas de Impedimento;
- 20 Cartas de Bug;
- 5 Cartas de *add on* ao projeto;
- 1 Cartão e 1 peão identificados como Programador Sênior;

2 Cartões e 2 peões identificados como Programador Pleno;
2 Cartões e 2 peões identificados como Programador Júnior;
1 Cartão e 1 peão identificados como Tester Pleno;
1 Cartão e 1 peão identificados como Tester júnior;
8 Cartões com as regras do jogo para facilitar a consulta;
Peças para representação de recompensas.

4.2. REGRAS

4.2.1. Preparação

Colocam-se sobre o tabuleiro de cartas, nos montes representados pelos fundos dos tipos das cartas de ação,

Colocam-se marcadores na primeira casa do contador de dias, e outro no contador de *sprints*

Distribuem-se os papéis de cada jogador (Programador pleno, Sênior etc), podendo ser por sorteio ou a escolha dos jogadores.

Um dos jogadores é sorteado com o papel de *ScrumMaster* e outro jogador é sorteado como *Product Owner*.

O Jogador que representa o *Product Owner* lê para os demais jogadores a carta que descreve o projeto a ser desenvolvido no Jogo (projeto 1). Após a leitura das necessidades do cliente, são lidas as cartas de tarefas do jogo, sendo os desenvolvimentos necessários do projeto.

Com a ajuda dos demais jogadores, o PO ordena as tarefas em ordem de prioridade de entrega de acordo com o que foi compreendido da descrição do projeto. Deve-se respeitar a precedência das tarefas apresentadas em sua carta. Por exemplo na figura abaixo vemos uma representação de carta que para que seja iniciada.

Figura 17: Legenda para carta de Tarefa

#2 ¹	CRUD Edições ²		1 ³ ● ⁴
Análise	Desenv	Teste	
0 ⁵	2 ⁶	1 ⁷	

Fonte: Autor

A carta de tarefa tem as seguintes características:

1. Identificador da tarefa
2. Título da tarefa
3. Quais tarefas são antecessoras da tarefa
4. Quantidade de recompensa a ser ganha quando a tarefa for finalizada
5. Quantidade de pontos que devem ser consumidos na análise
6. Quantidade de pontos que devem ser consumidos no desenvolvimento
7. Quantidade de pontos que devem ser consumidos na fase de teste

4.2.2. Início da *sprint*

Baseado na priorização do PO, a equipe seleciona quais tarefas serão entregues durante a *sprint*. A ordenação feita pelo PO não pode ser desfeita. As tarefas selecionadas são colocadas na coluna “A Fazer” do tabuleiro de *Kanban*.

4.2.3. Início do dia

Cada dia da *sprint* deve passar pelos seguintes passos:

1. Os jogadores compram uma carta de ação cada;
2. Os jogadores determinam quais tarefas executarão. Cada jogador somente pode realizar uma única tarefa, e cada tarefa somente pode ser executada por um jogador. Sendo a exceção quando a carta *Pair Programing* estiver em jogo
3. É comprada uma carta de “Treta”. Caso seja um *bug*, deve ser adicionado ao *backlog*, Caso o *bug* seja prioritário, deve ser colocado no topo da priorização e deve ser a próxima tarefa a ser executadas. Caso a carta seja de um impedimento, a carta deve ser colocada no tabuleiro de *kanban* conforme instruções no corpo desta.
4. Aplicam-se as cartas de ação para retirar os impedimentos, caso seja possível;
5. Cada jogador lança um dado de seis lados, o resultado é somado com o modificador do jogador para a coluna onde a tarefa que está sendo realizada se encontra. Caso o valor obtido seja maior ou igual aos pontos da carta naquela fase, a tarefa é passada para a próxima fase. Senão, os pontos obtidos são retirados da carta para serem consumidos na próxima rodada.

Exemplo: Um Programador Pleno decidiu executar uma tarefa que está em desenvolvimento. E o dado deu um resultado de três. Ele soma o seu modificador de +1 para desenvolvimento, um total de quatro. A carta que está sendo realizada tem cinco pontos para desenvolvimento, A carta não passa para a próxima fase, e resta um ponto que poderá ser testado novamente no próximo turno

6. Move-se o Marcador de dias para o próximo campo
7. Para cada Tarefa finalizada, a equipe ganha marcadores de recompensa de acordo com determinado na carta finalizada. Uma tarefa não pode ser finalizada se houver um impedimento sobre ela. Mas a tarefa pode passar de etapa de desenvolvimento.

4.2.4. Fim da *sprint*

Review

Quando o marcador de dias chega na última casa (5), a *sprint* é dada como encerrada. A equipe valida com o PO se as necessidades do projeto foram atendidas. Caso afirmativo, o jogo é encerrado. Caso o PO afirme que as demais tarefas ainda precisam ser entregues, o jogo começa uma nova *sprint*.

O PO Lança um dado, caso o resultado seja ímpar, uma nova tarefa do monte “ideias” é sorteada para dentro da próxima *sprint*.

Retrospectiva

Para cada tarefa não finalizada da *sprint*, retira-se uma recompensa da equipe.

Para cada *Bug* não resolvido que esteja no *backlog*, retiram-se duas recompensas da equipe.

Com as recompensas restantes, o *ScrumMaster* decide quais melhorias serão compradas e colocadas em jogo. Para comprar uma melhoria, são utilizados os marcadores de recompensas ganhos pela finalização das tarefas. O máximo de melhorias permitidas são três, caso se deseje comprar uma nova melhoria, e já houver três em jogo, deve-se retirar uma das melhorias.

4.2.5. Cartas de Ação X Impedimento

Ambas as cartas, possuem três atributos (Externa, Criatividade e Tecnologia), uma vez que uma tarefa esteja impedida, ela não pode ser considerada finalizada enquanto houver uma carta de impedimento sobre ela. Para retirar o impedimento, a equipe deve somar as pontuações de suas cartas de ação até atingir todos os pontos da carta de impedimento para cada atributo.

4.2.6. *Bugs*

Bugs são representações de problemas que ocorrem no dia a dia de desenvolvimento,

bugs de projetos que estão em andamento e devem ser resolvidas pela equipe. *Bugs* tem o mesmo processo de uma Tarefa

4.3. AVALIAÇÃO DO SCRUMBOARD

Após ministrada uma aula inicial sobre *Scrum*, observados seus princípios, reuniões, papéis e artefatos. Ao final da aula foi oferecido que o jogo *ScrumBoard* fosse aplicado ao grupo de alunos. Dos quais sete participantes, aceitaram participar do jogo.

Figura 18: Fotos da aplicação do jogo



Fonte: Autor

As regras do jogo foram explicadas aos participantes, e disponibilizado o jogo para que começassem a jogar. A partida se encerrou ao final da terceira *sprint*. Com todas as tarefas do projeto 1 realizadas.

Ao final do jogo, foi enviado aos participantes um questionário contendo as perguntas do modelo MEEGA+, com as perguntas específicas de aprendizagem do *ScrumBoard*. Conforme mostrado no quadro abaixo, onde são mostradas as perguntas, agrupadas pelas dimensões que são responsáveis por avaliar.

Quadro 3 - lista de perguntas do Modelo MEEGA+			
Experiencia do Usuário	Usabilidade	Estética	O design do jogo é atraente (tabuleiro, cartas, etc.).
			Os textos, cores e fontes combinam e são consistentes.
	Aprendizibilidade	Eu precisei aprender poucas coisas para poder começar a jogar o jogo.	
		Aprender a jogar este jogo foi fácil para mim.	
		Eu acho que a maioria das pessoas aprenderiam a jogar este jogo rapidamente.	
	Operabilidade	Eu considero que o jogo é fácil de jogar.	
		As regras do jogo são claras e compreensíveis.	
	Acessibilidade	As fontes (tamanho e estilo) utilizadas no jogo são legíveis.	
		As cores utilizadas no jogo são compreensíveis.	
	Confiança	Quando olhei pela primeira vez o jogo, eu tive a impressão de que seria fácil para mim.	
		A organização do conteúdo me ajudou a estar confiante de que eu iria aprender com este jogo.	
	Desafio	Este jogo é adequadamente desafiador para mim.	
		O jogo oferece novos desafios (oferece novos obstáculos, situações ou variações) com um ritmo adequado.	
		O jogo não se torna monótono nas suas tarefas (repetitivo ou com tarefas chatas).	
	Satisfação	Completar as tarefas do jogo me deu um sentimento de realização.	
		É devido ao meu esforço pessoal que eu consigo avançar no jogo.	
		Me sinto satisfeito com as coisas que aprendi no jogo.	
		Eu recomendaria este jogo para meus colegas.	
	Interação Social	Eu pude interagir com outras pessoas durante o jogo	

		O jogo promove momentos de cooperação e/ou competição entre os jogadores.
		Eu me senti bem interagindo com outras pessoas durante o jogo.
	Diversão	Eu me diverti com o jogo.
		Aconteceu alguma situação durante o jogo (elementos do jogo, competição, etc.) que me fez sorrir.
	Atenção Focada	Houve algo interessante no início do jogo que capturou minha atenção.
		Eu estava tão envolvido no jogo que eu perdi a noção do tempo.
		Eu esqueci sobre o ambiente ao meu redor enquanto jogava este jogo.
	Relevância	O conteúdo do jogo é relevante para os meus interesses.
		É claro para mim como o conteúdo do jogo está relacionado com a disciplina.
		O jogo é um método de ensino adequado para esta disciplina.
		Eu prefiro aprender com este jogo do que de outra forma (outro método de ensino).
Percepção da Aprendizagem	Aprendizagem de curto prazo	O jogo contribuiu para a minha aprendizagem na disciplina.
		O jogo foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.
	Objetivos de Aprendizagem	O jogo contribuiu para lembrar os conceitos do papel de ScrumMaster
		O jogo contribuiu para lembrar o conceito do papel de Product Owner
		O jogo contribuiu para compreensão do conceito de <i>sprint</i>
		O jogo contribuiu para compreensão do conceito da reunião diária
		O jogo contribuiu para compreensão do conceito da Review

O jogo contribuiu para compreensão do conceito da Retrospectiva

Todos os sete participantes responderam ao formulário. As respostas dadas por eles estão compiladas nos gráficos abaixo.



Observamos a partir dos resultados coletados nas dimensões do modelo as seguintes considerações:

Usabilidade

Na dimensão de Usabilidade sobre a, no tópico referente ao quão atraente é o jogo, mais de 70% das avaliações concordam que o jogo é atraente. Enquanto 29% ficaram neutras em relação a este tópico, mostrando uma grande aceitação sobre o item.

Confiança

Em relação a dimensão confiança no jogo, teve um sucesso razoável, visto que a concordância na facilidade de compressão do jogo foi perto de 30%, enquanto 14% dos participantes acharam que as mecânicas do jogo não seriam de fácil apreensão.

Sobre a dimensão de Desafio, 85% dos participantes afirmaram que o jogo é desafiador de uma forma adequada, ou seja, não fácil o suficiente para se tornar entediante, nem difícil o suficiente para impedir a continuidade ou tirar a vontade de continuar jogando.

Contudo, com relação a adição de novos desafios de forma contínua durante o jogo, apenas 57% dos participantes, avaliaram o jogo de forma positiva. Os demais participantes, consideraram este tópico de forma neutra.

O que traz o próximo tópico sobre a avaliação se o jogo se torna repetitivo. Trazendo 14% dos avaliados informando que o jogo é de monótono e não traz evoluções durante a evolução da partida. Porém aproximadamente 72% dos avaliados, concordaram em maior ou menor grau que o jogo traz evoluções e novos desafios durante o seu processo.

Satisfação

Sobre a dimensão que trata da satisfação do jogo, 85% dos avaliados, consideraram que o jogo traz um sentimento de realização ao se executar e completar as tarefas do jogo. Enquanto 42% dos avaliados consideraram que evoluíam através do próprio esforço durante o jogo 28% discordaram desta afirmação. Trazendo um ponto a ser questionado sobre como a execução e dinâmica das tarefas deve ser avaliada visando uma melhor avaliação deste ponto

Interação Social

Dos participantes avaliados 100% concordaram que se sentiram bem interagindo com os demais participantes e apenas 14% destes avaliaram de forma neutra que o jogo promove

cooperação ou competição entre os participantes.

Diversão

100% dos avaliados concordaram que se divertiram com o jogo, e que o jogo traz elementos divertidos durante o seu processo.

Atenção Focada

Com relação as afirmações sobre a dimensão de Atenção focada, podemos observar que o jogo foi avaliado positivamente nas afirmações sobre a atenção ter sido capturada no início do jogo, sobre o assunto do jogo e ter perdido a noção do tempo enquanto jogavam. Com concordância de 100% em maior ou menor grau. Apenas 28% avaliaram de forma neutra a perda de noção do ambiente. Trazendo que o jogo teve um bom desempenho nesta dimensão.

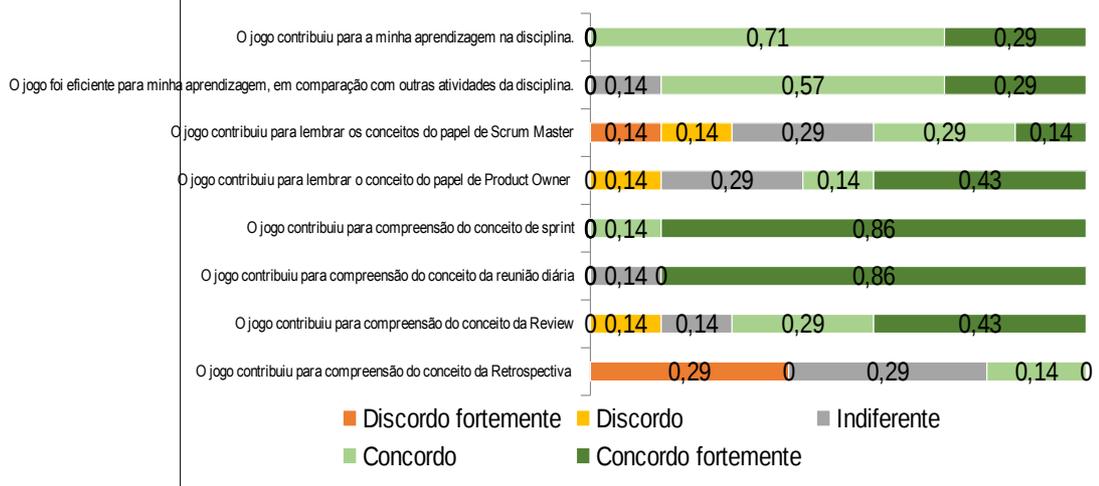
Relevância

Sobre a Relevância com a disciplina e o conteúdo de ensino. Houve uma avaliação positiva sobre o mesmo. E 100 % dos avaliados consideraram que o jogo é uma forma adequada de para ensinar o conteúdo. Tendo apenas 14% dos avaliados discordando que o jogo é uma forma preferível de ensino a outros métodos. Contudo 42% dos avaliados consideraram de forma neutra sobre esta própria afirmação. E apenas 42% afirmaram positivamente sobre a afirmação de que o jogo é uma forma melhor de aprendizagem que outras.

Portanto com relação a Experiência do usuário, podemos avaliar que o jogo tem pontos a serem melhorados nas dimensões Satisfação e Aprendizibilidade, pertencente a dimensão de Usabilidade. Onde os participantes notaram algumas fraquezas sobre como o jogo é passado, em relação as regras e informações iniciais. E em relação a satisfação, podemos visualizar que algumas mecânicas, regras ou mesmo o conteúdo das cartas pode ser revisto para dar maior dinamismo e desafio ao jogo.

Abaixo segue o gráfico com a compilação dos dados sobre a Percepção da Aprendizagem

Percepção da Aprendizagem



Os quesitos sobre aprendizagem de curto prazo obtiveram uma boa conformidade, uma vez que não houve referências negativas. O mesmo pode se dizer sobre os quesitos de aprendizagem sobre o conceito de *sprint* e Reunião Diária.

Contudo, os quesitos sobre a lembrança sobre o papel de *ScrumMaster*. E compreensão sobre as reuniões de *Review* e Retrospectiva não obtiveram uma boa aderência com os participantes. Transparecendo que as regras e mecânicas do jogo sobre estes conceitos.

5. CONCLUSÕES

Através da passagem em nas disciplinas do curso de Gestão de Tecnologias da Informação do IFSC, todas as disciplinas acessadas, em maior ou menor impacto auxiliaram para a construção deste trabalho. Contudo podemos ressaltar as disciplinas de Gerência de Projetos e Metodologia de Projetos, como fator de apresentação das disciplinas envolvidas neste trabalho. E as disciplinas que envolvem o desenvolvimento de programação por dar um vislumbre do processo e dos desafios que a gerência de projetos e o ágil devem ter para mudança de cultura e trazer benefícios para a área de desenvolvimento de software.

Este trabalho teve como objetivo prototipar uma ferramenta de auxílio para o ensino do *Scrum*, e avaliar seu impacto no aprendizado. Através do experimento realizado pudemos observar um grande engajamento por parte dos alunos, com a utilização do jogo.

O *ScrumBoard* mostrou-se uma ferramenta adequada ao ensino, porém com algumas áreas a serem melhoradas em novas versões. Isto porque, baseado nas avaliações realizadas pelos alunos, as dinâmicas relacionadas a Retrospectiva, *Review* e *ScrumMaster* não foram suficientes para melhorar a familiaridade dos jogadores com seus conceitos.

Para próximas versões do jogo ora proposto, poderemos considerar o aprimoramento de regras, bem como a criação de novas, a fim de melhorar a visualização dos papéis de *ScrumMaster* e PO, e tornar mais consistentes as reuniões de *Review* e Retrospectiva. Outra possibilidade a ser considerada consiste em dividir o jogo em partes, específicas para diferentes áreas do *Scrum*, tendo jogos centrados nas Reuniões, ou mesmo nos papéis.

Por fim, pode-se observar que o jogo como ferramenta de apoio ao ensino possibilitou que os alunos visualizassem o conteúdo da aula em um ambiente resistente a falhas, permitindo-lhes observar a mecânica de funcionamento do *Scrum* através de uma ferramenta lúdica. Ao interagirem uns com os outros em um ambiente simulado, foi incentivado o diálogo entre os jogadores para a resolução de conflitos, buscando a melhor solução face as situações apresentadas durante o jogo. Semelhante ao que se encontra no dia a dia de desenvolvimento do *Scrum*.

REFERÊNCIAS

AECV; MJMS. **O LÚDICO COMO FORMA DE APRENDIZAGEM NA EDUCAÇÃO INFANTIL**. 2012. Disponível em:

<<http://editorarealize.com.br/revistas/fiped/trabalhos/8c1b6fa97c4288a4514365198566c6fa.pdf>>. Acesso em: 08 ago 2017.

ARAÚJO, Emanuela Suassuna de; LIMA, Begna Janine da Silva. **UTILIZAÇÃO DE JOGOS COMO RECURSO DIDÁTICO PARA O ENSINO DE BIOLOGIA**. 2015.

Disponível em:

<http://www.editorarealize.com.br/revistas/conedu/trabalhos/TRABALHO_EV045_MD4_SA3_ID2451_10082015121116.pdf>. Acesso em: 08 ago. 2017.

BISSI, Wilson. SCRUM-Metodologia de desenvolvimento ágil. **Campo Digital**, v. 2, n. 1, p. 03-06, 2007.

BOMFIN, David Ferreira; NUNES, Paula Cristine de Ávila; HASTENREITER, Flávio. GERENCIAMENTO DE PROJETOS SEGUNDO O GUIA PMBOK: DESAFIOS PARA OS GESTORES. **Revista de Gestão e Projetos**, [s.l.], v. 3, n. 3, p.58-87, 1 nov. 2012. University Nove de Julho. <http://dx.doi.org/10.5585/gep.v3i3.78>. Disponível em: <<http://revistagep.org/ojs/index.php/gep/article/view/78>>. Acesso em: 23 ago. 2017.

CARVALHO, Bernardo Vasconcelos de; MELLO, Carlos Henrique Pereira. REVISÃO, ANÁLISE E CLASSIFICAÇÃO DA LITERATURA SOBRE O MÉTODO DE DESENVOLVIMENTO DE PRODUTOS ÁGIL SCRUM. In: SIMPÓSIO DE ADMINISTRAÇÃO DA PRODUÇÃO, LOGÍSTICA E OPERAÇÕES INTERNACIONAIS, Não use números Romanos ou letras, use somente números Arábicos., 2009, São Paulo. **Anais...** . São Paulo: Fgv-eaesp, 2009. v. 1, p. 1 - 16. Disponível em: <http://www.simpoi.fgvsp.br/index.cfm?FuseAction=arquivo.monta&ID_EdicaoArquivo=2009>. Acesso em: 18 set. 2017.

COHN, Mike. **Agile Succeeds Three Times More Often Than Waterfall**. 2012. Disponível em: <<https://www.mountangoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall>>. Acesso em: 05 nov. 2017.

COHN, Mike. *Succeeding with agile: software development using Scrum*. Pearson Education, 2010.

D'ÁVILA, Márcio. **PMBOK e Gerenciamento de Projetos**. 2015. Disponível em: <<http://www.mhavila.com.br/topicos/gestao/pmbok.html>>. Acesso em: 28 ago. 2017.

DE ARAÚJO, Emanuela Suassuna; DA SILVA LIMA, Begna Janine. UTILIZAÇÃO DE JOGOS COMO RECURSO DIDÁTICO PARA O ENSINO DE BIOLOGIA.

DELGADILLO, Jorge Medina; PUENTE, Mauricio Beuchot. **Thomas Murner: El Juego de las Cartas de Lógica**. México: Sapientia, 2017. 312 p. Disponível em: <https://www.amazon.com/Thomas-Murner-Traducción-introducción-Mauricio-ebook/dp/B0723987NR#reader_B0723987NR>. Acesso em: 11 nov. 2017.

DEVMEDIA. **Grupos de Processos segundo o PMBoK**. Disponível em: <<https://www.devmedia.com.br/grupos-de-processos-segundo-o-pmbok/27106>>. Acesso em: 28 ago. 2017.

DOMINGUES, Aline Natalia et al. Uso de protótipo em papel no design de um jogo educacional acessível. In: SBGAMES, 13., 2014, Porto Alegre. **Artigo**. Porto Alegre: Art & Design Track – Full Papers, 2014. p. 130 - 138. Disponível em: <[http://www.sbgames.org/sbgames2014/files/papers/art_design/full/A&D_Full_Uso de prototipo em papel.pdf](http://www.sbgames.org/sbgames2014/files/papers/art_design/full/A&D_Full_Uso%20de%20prototipo%20em%20papel.pdf)>. Acesso em: 30 set. 2017.

DUARTE, Luiz Cláudio S.; FEDERAL, Senado. *Jogos de Tabuleiro no Design de jogos Digitais*. SBC-SBGames, 2012.

- ETTINGER, Daniel. **A engrenagem do Scrum**. 2011. Disponível em: <<https://danielettinger.com/2011/04/06/a-engrenagem-do-scrum/>>. Acesso em: 28 ago. 2017.
- GALVÃO, Cristina Maria; SAWADA, Namie Okino; TREVIZAN, Maria Auxiliadora. Revisão sistemática: recurso que proporciona a incorporação das evidências na prática da enfermagem. **Revista Latino-Americana de Enfermagem**, v. 12, n. 3, p. 549-556, 2004.
- HELDMAN, Kim. **Gerência de Projetos: guia para o exame oficial do PMI**. Rio de Janeiro: Elsevier, 2005. 514 p.
- KISHIMOTO, Tizuko Morchida. O brinquedo na educação: considerações históricas. **São Paulo: FDE**, p. 39-45, 1990.
- MESTRA-UFPI, MJMS–Professora. O LÚDICO COMO FORMA DE APRENDIZAGEM NA EDUCAÇÃO INFANTIL. AECV-Aluna–UFPI.
- OLIVEIRA, Artur Pimentel de; FREITAS, Renan Cavichi de; CUNHA, Adilson Marques da. AVALIAÇÕES ANALÍTICAS DE USABILIDADE PARA DESENVOLVIMENTO ÁGIL DE SOFTWARE. In: ENCONTRO DE INICIAÇÃO CIENTÍFICA E PÓS-GRADUAÇÃO DO ITA, 20., 2014, São José dos Campos. **Anais...**. Anais: Instituto Tecnológico de Aeronáutica, 2014. p. 1 - 9. Disponível em: <https://www.researchgate.net/profile/Adilson_Cunha/publication/267327806_AVALIACOES_ANALITICAS_DE_USABILIDADE_PARA_DESENVOLVIMENTO_AGIL_DE_SOFTWARE/links/544aa8430cf2bcc9b1d2f878/AVALIACOES-ANALITICAS-DE-USABILIDADE-PARA-DESENVOLVIMENTO-AGIL-DE-SOFTWARE.pdf>. Acesso em: 11 nov. 2014.
- OLIVEIRA, Ebenezer Silva de. **Uso de Metodologias Ágeis no Desenvolvimento de Software**. 2003. 38 f. Monografia (Especialização) - Curso de Especialização em Informática, Universidade Federal de Minas Gerais, Belo Horizonte, 2003. Disponível em: <<http://www.infobitsolucoes.com/antigos/trabalhos/monografias/Monografia-EbenezerSilvaOliveira.pdf>>. Acesso em: 30 set. 2017.
- PETRI, Giani; VON WANGENHEIM, Christiane Gresse; BORGATTO, Adriano Ferretti. Evolução de um Modelo de Avaliação de Jogos para o Ensino de Computação.
- PHAN, Andrew; PHAN, Phoung-van. **SCRUM em ação: Gerenciamento e Desenvolvimento Ágil de Projetos de Software**. São Paulo: Novatec, 2011. 287 p.
- PICHLER, Roman. **Agile Product Management with Scrum: Creating Products that Customers Love (Adobe Reader)**. Addison-Wesley Professional, 2010.
- PRESSMAN, Roger S.. **Engenharia de Software: uma abordagem Profissional**. 7. ed. Porto Alegre: Amgh, 2011.
- PROJECT MANAGEMENT INSTITUTE (Ed.). **Um guia do conhecimento de gerenciamento de projeto: guia PMBOK**. 4. ed. Pensilvania: Project Management Institute, 2008.
- RF, SAMPAIO. Estudos de revisão sistemática: um guia para síntese criteriosa da evidência científica. 2007.
- SANTOS, Simone Cardoso dos. **A IMPORTÂNCIA DO LÚDICO NO PROCESSO ENSINO APRENDIZAGEM**. 2010. Disponível em: <<http://repositorio.ufsm.br/handle/1/393>>. Acesso em: 08 ago. 2017.

SILLER, Felipe; BRAGA, Juliana Cristina. **Software Educacional para Prática do Scrum**. 2013. Disponível em: <<http://br-ie.org/pub/index.php/wcbie/article/view/2664>>. Acesso em: 30 set. 2017.

SOFTWARE QUALITY GROUP – GQS (Florianópolis). Universidade Federal de Santa Catarina (Org.). **MEEGA+ A model for evaluating educational games**. Disponível em: <<http://www.gqs.ufsc.br/meega-a-model-for-evaluating-educational-games/>>. Acesso em: 30 out. 2017.

TAVARES, P. C. Utilização de jogo educativo como proposta para favorecer o ensino de ciências nas turmas do 8º ano da Escola Municipal Maria Caproni de Oliveira, Município de Carvalhópolis MG. 2013. 2014.

APÊNDICE A – AVALIAÇÃO MODELO MEEGA+

Nome do jogo :SCRUMBOARD

Gostaríamos que você respondesse as questões abaixo sobre a sua percepção da qualidade do jogo para nos ajudar a melhorá-lo. Todos os dados são coletados anonimamente e somente serão utilizados no contexto desta pesquisa. Algumas fotografias poderão ser feitas como registro desta atividade, mas não serão publicadas em nenhum local sem autorização.

Nome do pesquisador responsável: Frederico Dubiel

Local e data _____

Informações Demográficas	
Instituição:	Instituto Federal de Santa Catarina
Curso:	Gestão da Tecnologia da Informação
Disciplina:	
Faixa etária:	<input type="checkbox"/> Menos de 18 anos <input type="checkbox"/> 18 a 28 anos <input type="checkbox"/> 29 a 39 anos <input type="checkbox"/> 40 a 50 anos <input type="checkbox"/> Mais de 50 anos
Gênero:	<input type="checkbox"/> Masculino <input type="checkbox"/> Feminino
Com que frequência você costuma jogar jogos digitais?	<input type="checkbox"/> Nunca: nunca jogo. <input type="checkbox"/> Raramente: jogo de tempos em tempos. <input type="checkbox"/> Mensalmente: jogo pelo menos uma vez por mês. <input type="checkbox"/> Semanalmente: jogo pelo menos uma vez por semana. <input type="checkbox"/> Diariamente: jogo todos os dias.
Com que frequência você costuma jogar jogos não-digitais (de cartas, tabuleiro, etc)?	<input type="checkbox"/> Nunca: nunca jogo. <input type="checkbox"/> Raramente: jogo de tempos em tempos. <input type="checkbox"/> Mensalmente: jogo pelo menos uma vez por mês. <input type="checkbox"/> Semanalmente: jogo pelo menos uma vez por semana. <input type="checkbox"/> Diariamente: jogo todos os dias.

Por favor, **marque uma opção** de acordo com o quanto você concorda ou discorda de cada afirmação abaixo.

Experiência do Jogador					
Afirmações	Marque uma opção conforme sua avaliação				
	Discordo Totalmente	Discordo	Nem discordo, nem concordo	Concordo	Concordo Totalmente
O design do jogo é atraente (tabuleiro, cartas, etc.).	⌘	⌘	⌘	⌘	⌘
Os textos, cores e fontes combinam e são consistentes.	⌘	⌘	⌘	⌘	⌘
Eu precisei aprender poucas coisas para poder começar a jogar o jogo.	⌘	⌘	⌘	⌘	⌘
Aprender a jogar este jogo foi fácil para mim.	⌘	⌘	⌘	⌘	⌘
Eu acho que a maioria das pessoas aprenderiam a jogar este jogo rapidamente.	⌘	⌘	⌘	⌘	⌘
Eu considero que o jogo é fácil de jogar.	⌘	⌘	⌘	⌘	⌘
As regras do jogo são claras e compreensíveis.	⌘	⌘	⌘	⌘	⌘
As fontes (tamanho e estilo) utilizadas no jogo são legíveis.	⌘	⌘	⌘	⌘	⌘
As cores utilizadas no jogo são compreensíveis.	⌘	⌘	⌘	⌘	⌘
Quando olhei pela primeira vez o jogo, eu tive a impressão de que seria fácil para mim.	⌘	⌘	⌘	⌘	⌘
Afirmações	Marque uma opção conforme sua avaliação				
	Discordo Totalmente	Discordo	Nem discordo, nem concordo	Concordo	Concordo Totalmente
A organização do conteúdo me ajudou a estar confiante de que eu iria aprender com este jogo.	⌘	⌘	⌘	⌘	⌘
Este jogo é adequadamente desafiador para mim.	⌘	⌘	⌘	⌘	⌘
O jogo oferece novos desafios (oferece novos obstáculos, situações ou variações) com um ritmo adequado.	⌘	⌘	⌘	⌘	⌘
O jogo não se torna monótono nas suas tarefas (repetitivo ou com tarefas chatas).	⌘	⌘	⌘	⌘	⌘
Completar as tarefas do jogo me deu um sentimento de realização.	⌘	⌘	⌘	⌘	⌘
É devido ao meu esforço pessoal que eu consigo avançar no jogo.	⌘	⌘	⌘	⌘	⌘
Me sinto satisfeito com as coisas que aprendi no jogo.	⌘	⌘	⌘	⌘	⌘

Eu recomendaria este jogo para meus colegas.	☒	☒	☒	☒	☒
Eu pude interagir com outras pessoas durante o jogo.	☒	☒	☒	☒	☒
O jogo promove momentos de cooperação e/ou competição entre os jogadores.	☒	☒	☒	☒	☒
Eu me senti bem interagindo com outras pessoas durante o jogo.	☒	☒	☒	☒	☒
Eu me diverti com o jogo.	☒	☒	☒	☒	☒
Aconteceu alguma situação durante o jogo (elementos do jogo, competição, etc.) que me fez sorrir	☒	☒	☒	☒	☒
Houve algo interessante no início do jogo que capturou minha atenção.	☒	☒	☒	☒	☒
Eu estava tão envolvido no jogo que eu perdi a noção do tempo.	☒	☒	☒	☒	☒
Eu esqueci sobre o ambiente ao meu redor enquanto jogava este jogo.	☒	☒	☒	☒	☒
O conteúdo do jogo é relevante para os meus interesses.	☒	☒	☒	☒	☒
É claro para mim como o conteúdo do jogo está relacionado com a disciplina.	☒	☒	☒	☒	☒
O jogo é um método de ensino adequado para esta disciplina.	☒	☒	☒	☒	☒
Eu prefiro aprender com este jogo do que de outra forma (outro método de ensino).	☒	☒	☒	☒	☒

Percepção da Aprendizagem					
Afirmações	Marque uma opção conforme sua avaliação				
	Discordo Totalmente	Discordo	Nem discordo, nem concordo	Concordo	Concordo Fortemente
O jogo contribuiu para a minha aprendizagem na disciplina.	☒	☒	☒	☒	☒
O jogo foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.	☒	☒	☒	☒	☒
O jogo contribuiu para lembrar os conceitos do papel de ScrumMaster	☒	☒	☒	☒	☒
O jogo contribuiu para lembrar o conceito do papel de Product Owner	☒	☒	☒	☒	☒
O jogo contribuiu para compreensão do conceito de sprint	☒	☒	☒	☒	☒
O jogo contribuiu para compreensão do conceito da reunião diária	☒	☒	☒	☒	☒
O jogo contribuiu para compreensão do conceito da Review	☒	☒	☒	☒	☒
O jogo contribuiu para compreensão do conceito da Retrospectiva	☒	☒	☒	☒	☒

O jogo contribuiu para compreensão do conceito da Reunião Diária	×	×	×	×	×

Cite 3 pontos fortes do jogo:

Dê 3 sugestões para a melhoria do jogo:

Comentários adicionais:

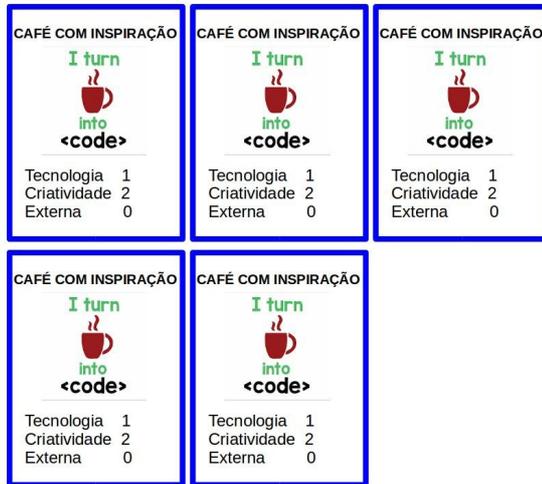
Muito obrigado pela sua contribuição!

APÊNDICE B – ELEMENTOS DO SCRUMBOARD

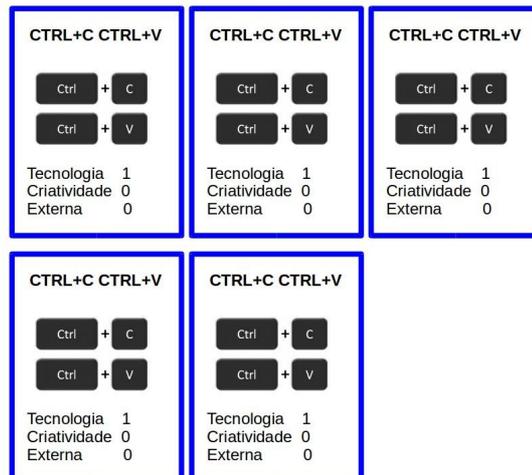
Cartas de Ação Ócio Criativo – Impressão em formato A4 Paisagem



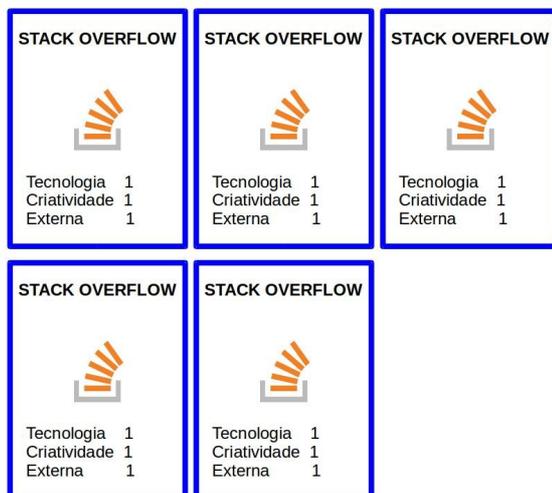
Cartas de Ação Café com Inspiração – Impressão em formato A4 Paisagem



Cartas de Ação Ctrl+C Ctrl+V – Impressão em formato A4 Paisagem



Cartas de Ação StackOverFlow – Impressão em formato A4 Paisagem



Fundo para cartas de ajuda – Impressão em modelo A4 Paisagem



Cartas de Melhoria – Impressão em modelo A4 Paisagem
 Fundo de Cartas de Idéias– Impressão em modelo A4 Paisagem



Cartas de Idéias– Impressão em modelo A4

Alterar o Visual			-	Novo relatório			-
Análise	Desenv	Teste	●	Análise	Desenv	Teste	●
0	2	1		0	3	2	
Visual Responsivo			●●	Integração API			●●●
Análise	Desenv	Teste		Análise	Desenv	Teste	
0	5	1		0	8	5	
Hotsite			●●	Importação de dados			●●●
Análise	Desenv	Teste		Análise	Desenv	Teste	
0	3	1		0	8	2	

Fundo de Cartas de Idéias– Impressão em modelo A4



Cartas de Projeto 1– Impressão em modelo A4

Cartas de Projeto 1– Impressão em modelo A4

<table border="1"> <tr> <td>#1</td> <td>CRUD Título</td> <td>-</td> </tr> <tr> <td>Análise</td> <td>Desenv</td> <td>Teste</td> </tr> <tr> <td>0</td> <td>2</td> <td>1</td> </tr> </table>	#1	CRUD Título	-	Análise	Desenv	Teste	0	2	1	<table border="1"> <tr> <td>#5</td> <td>Cadastro de arco de histórias</td> <td>2</td> </tr> <tr> <td>Análise</td> <td>Desenv</td> <td>Teste</td> </tr> <tr> <td>0</td> <td>3</td> <td>2</td> </tr> </table>	#5	Cadastro de arco de histórias	2	Análise	Desenv	Teste	0	3	2
#1	CRUD Título	-																	
Análise	Desenv	Teste																	
0	2	1																	
#5	Cadastro de arco de histórias	2																	
Análise	Desenv	Teste																	
0	3	2																	
<table border="1"> <tr> <td>#2</td> <td>CRUD Edições</td> <td>1</td> </tr> <tr> <td>Análise</td> <td>Desenv</td> <td>Teste</td> </tr> <tr> <td>0</td> <td>2</td> <td>1</td> </tr> </table>	#2	CRUD Edições	1	Análise	Desenv	Teste	0	2	1	<table border="1"> <tr> <td>#6</td> <td>Cadastro de Edições Campo Precisa de conserto</td> <td>2</td> </tr> <tr> <td>Análise</td> <td>Desenv</td> <td>Teste</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> </table>	#6	Cadastro de Edições Campo Precisa de conserto	2	Análise	Desenv	Teste	0	1	1
#2	CRUD Edições	1																	
Análise	Desenv	Teste																	
0	2	1																	
#6	Cadastro de Edições Campo Precisa de conserto	2																	
Análise	Desenv	Teste																	
0	1	1																	
<table border="1"> <tr> <td>#3</td> <td>Relatório de HQ's por edição</td> <td>2</td> </tr> <tr> <td>Análise</td> <td>Desenv</td> <td>Teste</td> </tr> <tr> <td>0</td> <td>5</td> <td>3</td> </tr> </table>	#3	Relatório de HQ's por edição	2	Análise	Desenv	Teste	0	5	3	<table border="1"> <tr> <td>#7</td> <td>Relatório de edições faltantes</td> <td>2</td> </tr> <tr> <td>Análise</td> <td>Desenv</td> <td>Teste</td> </tr> <tr> <td>0</td> <td>13</td> <td>5</td> </tr> </table>	#7	Relatório de edições faltantes	2	Análise	Desenv	Teste	0	13	5
#3	Relatório de HQ's por edição	2																	
Análise	Desenv	Teste																	
0	5	3																	
#7	Relatório de edições faltantes	2																	
Análise	Desenv	Teste																	
0	13	5																	
<table border="1"> <tr> <td>#4</td> <td>Ordenação das listagens por colunas</td> <td>1</td> </tr> <tr> <td>Análise</td> <td>Desenv</td> <td>Teste</td> </tr> <tr> <td>0</td> <td>8</td> <td>3</td> </tr> </table>	#4	Ordenação das listagens por colunas	1	Análise	Desenv	Teste	0	8	3	<table border="1"> <tr> <td>#8</td> <td>CRUD Usuário</td> <td>-</td> </tr> <tr> <td>Análise</td> <td>Desenv</td> <td>Teste</td> </tr> <tr> <td>0</td> <td>5</td> <td>1</td> </tr> </table>	#8	CRUD Usuário	-	Análise	Desenv	Teste	0	5	1
#4	Ordenação das listagens por colunas	1																	
Análise	Desenv	Teste																	
0	8	3																	
#8	CRUD Usuário	-																	
Análise	Desenv	Teste																	
0	5	1																	

Fundo de Cartas de Projeto 1– Impressão em modelo A4

#1			-
Cadastrar Produtos			●
Análise	Desenv	Teste	
0	5	3	

#5			1
Ordenar por preço			●
Análise	Desenv	Teste	
0	3	2	

#2			-
Cadastrar Categorias			●
Análise	Desenv	Teste	
0	2	1	

#6			3 1
Cupom de desconto			●●
Análise	Desenv	Teste	
0	5	5	

#3			2 1
Listar Por Categorias			●●
Análise	Desenv	Teste	
0	5	1	

#7			3 1
Pesquisa por produto			●●
Análise	Desenv	Teste	
0	8	3	

#4			1
Ordenar por nome			●● ●
Análise	Desenv	Teste	
0	3	1	

#8			9 10
Tela de Checkout			●● ●
Análise	Desenv	Teste	
0	8	5	

Cartas de Projeto E-Commerce – Impressão em modelo A4

#1			-
Cadastrar Produtos			●
Análise	Desenv	Teste	
0	5	3	

#5			1
Ordenar por preço			●
Análise	Desenv	Teste	
0	3	2	

#2			-
Cadastrar Categorias			●
Análise	Desenv	Teste	
0	2	1	

#6			3 1
Cupom de desconto			●●
Análise	Desenv	Teste	
0	5	5	

#3			2 1
Listar Por Categorias			●●
Análise	Desenv	Teste	
0	5	1	

#7			3 1
Pesquisa por produto			●●
Análise	Desenv	Teste	
0	8	3	

#4			1
Ordenar por nome			●● ●
Análise	Desenv	Teste	
0	3	1	

#8			9 10
Tela de Checkout			●● ●
Análise	Desenv	Teste	
0	8	5	

Cartas de Projeto E-Commerce – Impressão em modelo A4

#9	Gateway de pagamento	-	●●
Análise	Desenv	Teste	
0	5	5	

#13	Vitrine exibe preço com desconto	6	●●
Análise	Desenv	Teste	
0	8	3	

#10	Cadastro de Cliente	-	●
Análise	Desenv	Teste	
0	5	2	

#14	Login de cliente	10	●
Análise	Desenv	Teste	
0	3	1	

#11	Cadastro de novo Endereço	10	●
Análise	Desenv	Teste	
0	3	1	

#15	Login de administrador	-	●●
Análise	Desenv	Teste	
0	3	1	

#12	Cadastro de descontos	3	●●
Análise	Desenv	Teste	
0	5	1	

#16	Banner para promoção	10	●●
Análise	Desenv	Teste	
0	5	2	

Cartas de Projeto E-Commerce – Impressão em modelo A4

#17	Relatório de vendas	8	●●
Análise	Desenv	Teste	
0	13	5	

#21	Avise-me quando chegar	22	●
Análise	Desenv	Teste	
0	5	2	

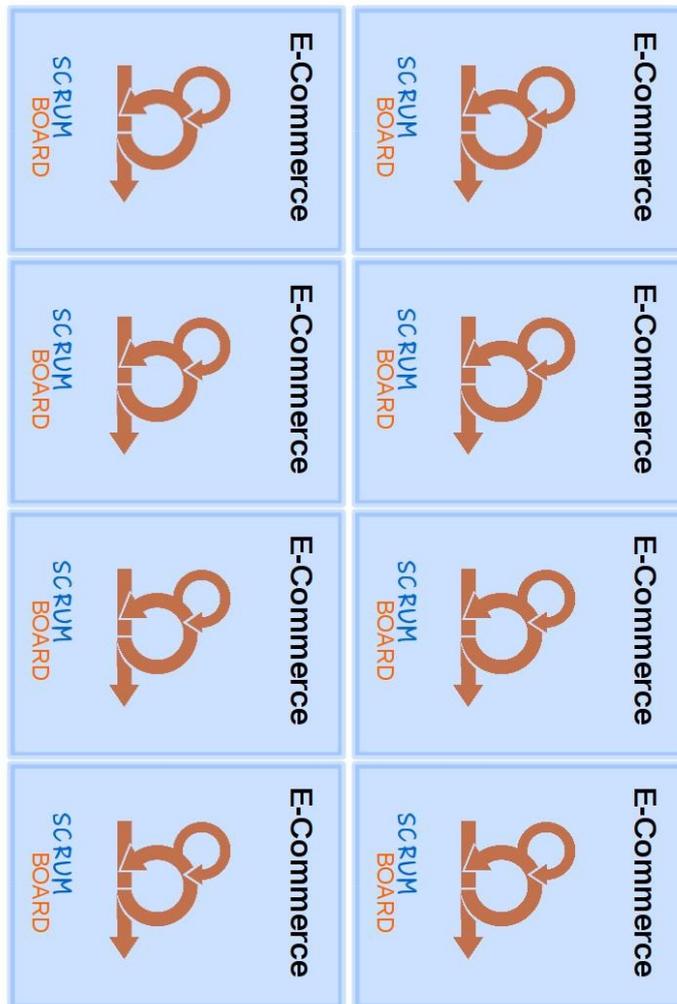
#18	Relatório de estoque	1	●
Análise	Desenv	Teste	
0	8	5	

#22	Exibir Produto em detalhe	3	●
Análise	Desenv	Teste	
0	5	2	

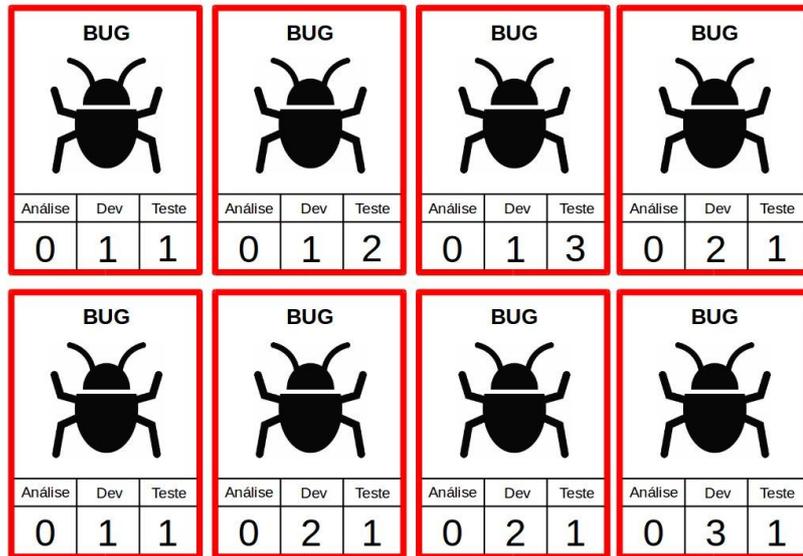
#19	Cadastro de newsletter	10	●
Análise	Desenv	Teste	
0	5	1	

#20	Relatório de newsletter	19	●●
Análise	Desenv	Teste	
0	3	1	

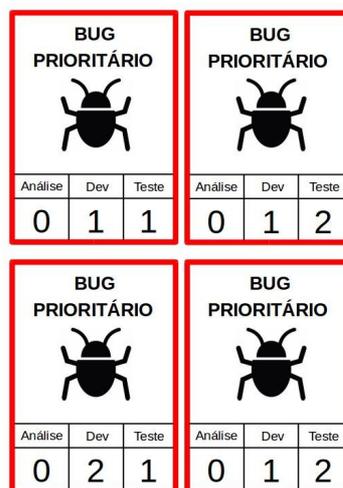
Fundo de Cartas de Projeto E-Commerce – Impressão em modelo A4



Cartas de Treta – Impressão em modelo A4 Paisagem



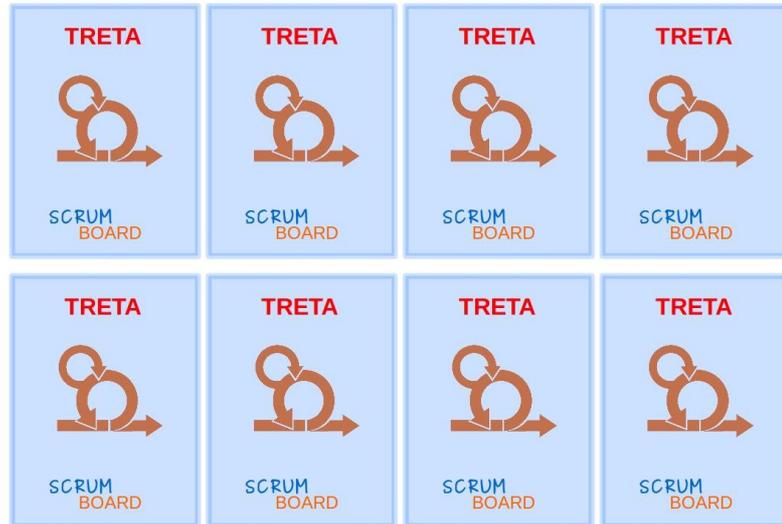
Cartas de Treta de Bug Prioritário – Impressão em modelo A4 Paisagem



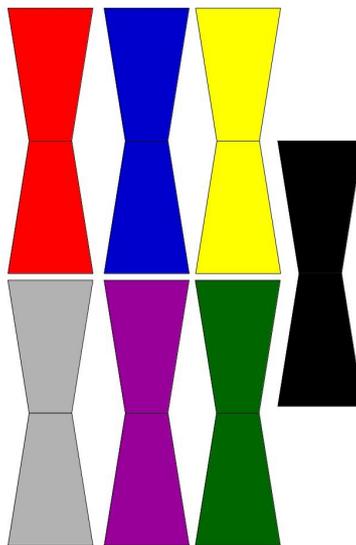
Cartas de Treta de Impedimentos! - Impressão em A4 Paisagem
 Cartas de Treta de Impedimentos! - Impressão em A4 Paisagem



Cartas de Treta de Ufa! - Impressão em A4 Paisagem
Cartas de Fundo de Treta – Impressão em A4 Paisagem



Peões – Impressão em modelo A4



Cartões de trabalhadores – Impressão em modelo A4

	Programador Sênior		Programador Pleno		Programador Júnior
Análise	Desenv	Teste	Análise	Desenv	Teste
+2	+3	+1	+0	+1	+0
Análise	Desenv	Teste	Análise	Desenv	Teste
+0	+1	+0	-1	+0	-2

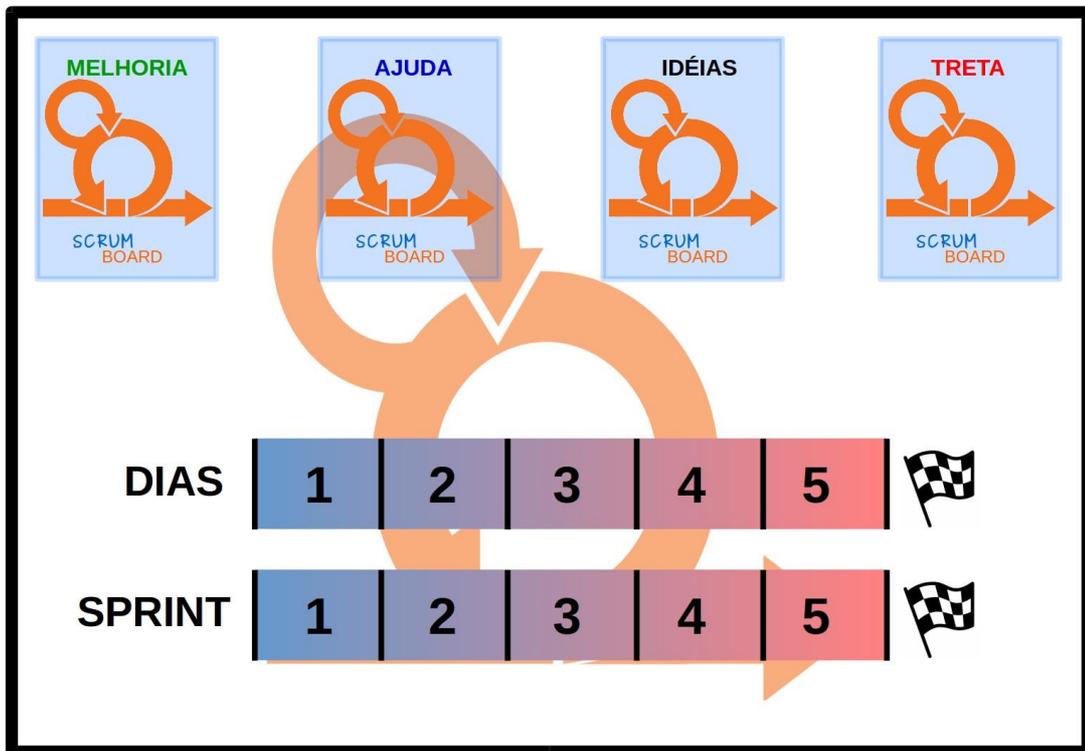
	Programador Pleno		Programador Júnior
Análise	Desenv	Teste	Análise
+0	+1	+0	-1
Análise	Desenv	Teste	Análise
+0	+1	+0	-1

Tabuleiro Kanban – Impressão em modelo A3 Paisagem

A FAZER	ANÁLISE	DESENV	
---------	---------	--------	--

TESTE	PRONTO
-------	--------

Tabuleiro – Impressão em modelo A3 Paisagem



APÊNDICE F – Declaração de finalização de trabalho de curso



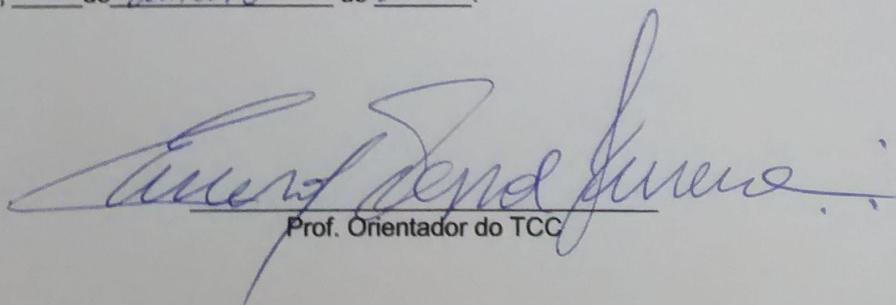
INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS

DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO

Declaro que o(a) estudante Frederico Dübich,
matrícula n° 151005538-0 do Curso Superior de
Gestão de Tecnologia da Informação, defendeu o trabalho intitulado
SCRUM board: Um jogo para o auxílio do ensino do SCRUM,
o qual está apto a fazer parte do banco de dados da Biblioteca Hercílio Luz do Instituto Federal de
Santa Catarina, Câmpus Florianópolis.

Florianópolis, 2 de Janeiro de 2018.


Prof. Orientador do TCC