

UNIVERSIDADE FEDERAL DE VIÇOSA  
CAMPUS DE RIO PARANAÍBA  
SISTEMAS DE INFORMAÇÃO

LUIZ PAULO DA SILVA

CRIAÇÃO DE UM JOGO PARA ENSINO DE  
GERENCIAMENTO DE RISCOS NO CONTEXTO DO  
SCRUM

RIO PARANAÍBA  
2018

LUIZ PAULO DA SILVA

CRIAÇÃO DE UM JOGO PARA ENSINO DE GERENCIAMENTO DE  
RISCOS NO CONTEXTO DO SCRUM

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal de Viçosa – Campus de Rio Paranaíba como requisito para a obtenção do título de bacharel em Sistemas de Informação

Orientador: Iris Fabiana de Barcelos Tronto

Coorientador: Pedro Moises de Souza

RIO PARANAÍBA

2018

*Primeiramente a Deus, por ser essencial em minha vida, pois sem ele eu não teria forças para essa jornada, aos meus pais, minha noiva e as pessoas que estiveram ao meu lado, que não mediram esforços para me apoiar e me ajudar na conclusão deste trabalho.*

# Agradecimentos

Agradeço em primeiro lugar, a Deus pilar principal da minha vida que me concedeu saúde, perseverança, por me fortalecer e sustentar em todos os momentos.

Aos meus pais, pelo amor, incentivo e apoio incondicional que me deram nesta trajetória.

A todos os professores que compartilharam seus conhecimentos durante a graduação, transmitindo-me valores e ensinamentos.

A todos os meus colegas que de alguma forma contribuíram para esta conquista, o meu muito obrigado.

*“A persistência é o caminho do êxito.”*  
*(Charles Chaplin)*

# Resumo

Nos últimos anos, a metodologia ágil tem se popularizado e ganhado a confiança da indústria para a gerência e desenvolvimento de *software*. Isso se deve a necessidade de adaptação rápida de projetos de médias e micro empresas. A metodologia ágil Scrum mantém uma abordagem iterativa incremental com rápida entrega e possui maior flexibilidade para solucionar pequenos problemas que a abordagem clássica. O gerenciamento de riscos, é considerado uma área fundamental do gerenciamento de projetos no desenvolvimento de *software*. Com a crescente atenção para a importância de ter um bom gerenciamento de riscos, este processo representa a forma pela qual os riscos e empecilhos são gerenciados para garantir os objetivos principais do desenvolvimento. Este trabalho propõe a criação de um jogo educacional para o ensino de gerenciamento de risco proposto de acordo com a filosofia Scrum. Esse jogo permite o ensino de metodologia Scrum, enfatizando as atividades que constituem o gerenciamento de risco, como a identificação, análise e tratamento de riscos de projetos, voltado a apoiar o ensino na identificação, análise e correção de riscos. Com o resultado deste trabalho a produção do jogo como ferramenta de apoio ao ensino das práticas de engenharia de *software* tanto no meio acadêmico como para a indústria de *software*.

**Palavras-chaves:** Metodologia ágil, Scrum, gerenciamento de risco.

# Abstract

In recent years, a methodology has been popularized for information on software management and software development. The Scrum methodology remains an incremental iterative approach with faster delivery and greater flexibility for more complex problems than a classical approach. This work proposes the creation of an educational game for risk management teaching proposed according to the Scrum philosophy. Risk management is a key software project management program. With increasing attention to the importance of a recovery process, this process is a risk-taking and empirical way to secure the major development goals. This game allows the teaching of Scrum methodology, emphasizing the activities that constitute risk management, such as the identification, analysis and treatment of project risks, aimed at supporting teaching in the identification, analysis and correction of risks. With the result of this work the production of the game as a tool to support the teaching of software engineering practices in both academia and the software industry.

**Key-words:** Agile Methodology, Scrum, risk management.

# **Lista de ilustrações**

Figura 1 – Visão Geral do Gerenciamento de risco - PMBOK . . . . .	19
Figura 2 – Visão Geral do Ciclo de vida Scrum . . . . .	27
Figura 3 – Scrum original modelado no SPEM . . . . .	28
Figura 4 – Scrum adaptado modelado no SPEM - Pré-game . . . . .	29
Figura 5 – Scrum adaptado modelado no SPEM - Fase game . . . . .	30
Figura 6 – Scrum adaptado no SPEM - Pós-game . . . . .	31
Figura 7 – Casos de uso tela inicial. . . . .	38
Figura 8 – Casos de uso tela professor . . . . .	39
Figura 9 – Caso de uso referente ao cenário. . . . .	39
Figura 10 – Desenvolvimento com Unity . . . . .	43
Figura 11 – Cenário inicial do jogo. . . . .	44
Figura 12 – Cenário do jogo. . . . .	44
Figura 13 – Cenário do jogo. . . . .	45
Figura 14 – Cenário do jogo. . . . .	46
Figura 15 – Menu principal do jogo . . . . .	47
Figura 16 – Menu de acesso para o professor . . . . .	48
Figura 17 – Menu de inserção de perguntas . . . . .	48
Figura 18 – Menu edição de perguntas . . . . .	49
Figura 19 – Menu interno . . . . .	49
Figura 20 – Tela do quiz . . . . .	50
Figura 21 – Mensagem de acerto da questão. . . . .	50
Figura 22 – Mensagem ao usuário . . . . .	51
Figura 23 – Pontuação do jogo. . . . .	51
Figura 24 – Tela de game over da fase. . . . .	52
Figura 25 – Pontuação do jogo . . . . .	53
Figura 26 – Mensagem de ajuda . . . . .	53

# **Lista de tabelas**

Tabela 1 – GDD . . . . .	40
Tabela 2 – GDD . . . . .	41

# **Lista de abreviaturas e siglas**

ABNT	Associação Brasileira de Normas Técnicas
CMMI	Capability Maturity Model Integration
CRP	<i>Campus</i> de Rio Paranaíba
DRU	Desenvolvimento para Reutilização
GDD	Game Design Document
GDE	Gerência de Decisões
GRI	Gerência de Riscos
MPS.BR	Melhoria de Processo de <i>software</i> Brasileiro
NBR	Norma Brasileira
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
RUP	Rational Unified Process

# Sumário

<b>1</b>	<b>Introdução</b>	<b>12</b>
1.1	Contexto e Justificativa	12
1.2	Objetivos	13
1.3	Organização do Documento	14
<b>2</b>	<b>Referencial Teórico</b>	<b>15</b>
2.1	Gerenciamento de Projetos	15
2.2	Gerenciamento de Risco	16
2.2.1	Gerenciamento de risco segundo método ágil Scrum	17
2.2.2	Gerenciamento de risco regundo PMI-PMBOK	18
2.2.3	Gerenciamento de risco segundo CMMI	20
2.2.4	Gerenciamento de Risco Segundo MPS.BR	21
2.3	Métodos Ágeis	22
2.4	Scrum	23
2.4.1	Time Scrum	24
2.4.2	Product Owner	25
2.4.3	Scrum Master	25
2.4.4	Backlog do Produto	25
2.4.5	Sprint	25
2.4.6	Definição dos Processos do Scrum	26
2.5	Desenvolvimento de Jogos	30
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>33</b>
<b>4</b>	<b>Medodologia</b>	<b>35</b>
4.1	Etapas do Desenvolvimento	35
4.2	Ferramenta e linguagem	36
4.2.1	Engine Unity	36
4.2.2	Recursos e Componentes	36
4.2.3	Linguagens de Programação	36
<b>5</b>	<b>Resultados</b>	<b>38</b>
5.1	Projeto	38
5.1.1	Requisitos	38
5.1.2	Game Design Document (GDD)	40
5.2	Implementação	42
5.2.1	Descrição do Jogo	42
5.2.2	Processo de Desenvolvimento	43
<b>6</b>	<b>Conclusão</b>	<b>54</b>



# 1 Introdução

## 1.1 Contexto e Justificativa

A maioria dos projetos de *software* podem sofrer com alterações de cronograma, qualidade ou de custo, muitas vezes se deve ao mau planejamento. O Gerenciamento de Projetos é a forma de aplicação do conhecimento, técnicas e ferramentas para atender os requisitos do escopo do projeto, custo e prazo([PALUDO; RAABE; BENITTI, 2013](#)).

As metodologias ágeis, já não são novidade, mas modificam a forma que o gerenciamento de projetos é aplicado. Com conceitos a fim de facilitar e estimular seu uso como o foco nas pessoas e não em processos, deixando de lado documentações exageradas para buscar mais agilidade e simplicidade no desenvolvimento([PALUDO; RAABE; BENITTI, 2013](#)).

As metodologias ágeis têm sido amplamente utilizadas por empresas de desenvolvimento de *software*. Porém, elas apresentam uma documentação muito limitada, além de às vezes não incluir algumas práticas importantes no contexto de desenvolvimento de *software*, como a de gerenciamento de riscos ([PALUDO; RAABE; BENITTI, 2013](#)).

O método ágil Scrum tem como objetivo, segundo [Schwaber e Sutherland \(2013\)](#) definir um processo para projeto e desenvolvimento de *software* orientado a objeto, que seja focado nas pessoas e que seja indicado para ambientes em que os requisitos surgem e mudam rapidamente. Baseando-se em princípios como: equipes pequenas; requisitos que são pouco estáveis ou desconhecidos; e iterações curtas, dividindo o desenvolvimento em intervalos de tempos.

O Gerenciamento de riscos em projetos tem como objetivo aumentar o impacto positivo e diminuir a probabilidade de riscos que possam impactar negativamente no projeto. É possível ver nas empresas uma crescente atenção para a importância de ter um bom gerenciamento de riscos, já que este processo representa a forma pela qual os empecilhos são gerenciados para garantir que os objetivos de prazo, custos, qualidade e escopo do projeto sejam alcançados com sucesso. O gerenciamento de riscos, é considerado uma área do gerenciamento de projetos fundamental no desenvolvimento de *software*. Possuindo atividades de análise e planejamento de risco que um projeto está sujeito a passar durante o desenvolvimento ([SOMMERVILLE; ARAKAKI; MELNIKOFF, 2008](#)).

Com um gerenciamento de riscos mais eficiente de maneira que possa tornar mais fácil o controle dos problemas e assegurar que não vão direcionar a um projeto inviável ([SOMMERVILLE; ARAKAKI; MELNIKOFF, 2008](#)). Devido a isso esta havendo a necessidade de incluir práticas de gerenciamento de riscos na metodologia Scrum, e ainda

com maior importância de se oferecer recursos para treinamento e ensino da metodologia Scrum com a inclusão de novas atividades.

Durante o planejamento e tomadas de decisões são oportunidades para reduzir as ameaças ao projeto. Os possíveis riscos identificados são devidamente encaminhado para algum membro da equipe envolvido no desenvolvimento através do Controle de riscos, sendo um processo contínuo no ciclo de vida do projeto (PMI, 2013). O uso de jogos educativos para ensino de engenharia de *software* tem sido debatido segundo [Navarro e Hoek \(2004\)](#), que os recomendam como um complemento para o aprendizado tradicional de processos de desenvolvimento de *software*. Utilizando a abordagem de um jogo em que apresente um ambiente de desenvolvimento, para permitir que o estudante possa aprender e adquirir habilidades. aplicando seus conhecimentos na prática.

O jogo foi desenvolvido utilizando Unity 3D. Em que simula um ambiente de uma empresa de desenvolvimento em 2D, em que o jogador deve percorrer todo o cenário para solucionar as questões espalhadas pelo cenário. Para isso são apresentados pequenas estrelas sinalizando para que seja aberto a pagina de questão, existe ainda um sinal para curiosidades e informações sobre o Scrum no decorrer do percurso. O jogo consiste em estimular e desenvolver o interesse do aluno por meio das situações vivenciadas no dia a dia do desenvolvimento de uma empresa, como o gerenciamento de requisitos, tomada de decisões, gerenciamento de risco, dentre outros. O jogo foi desenvolvido para apoio ao ensino do Scrum em um ambiente simulado adaptando as melhorias necessárias. Espera-se que o jogador consiga compreender e aplicar de maneira de fácil aprendizado entre o teórico ensinado em salas de aula e o prático, fazendo com que aumente o entendimento dos conceitos, capacitando o estudante para a aplicação do conhecimento abordado.

## 1.2 Objetivos

Em busca de maximizar os resultados do gerenciamento de risco e minimizar os futuros problemas, o jogo explora a capacidade, entendimento e aplicação de importantes conceitos durante o desenvolvimento de um projeto de *software*.

O principal objetivo é a utilização do jogo dentro de uma unidade de ensino, em cursos de nível técnico e superior, para o ensino do método ágil Scrum, sendo assim o público alvo são alunos da área de computação e tecnologia de informação. Também reconhecendo que o público alvo já esteja habituado com linguagens e características de métodos de desenvolvimento e possua conhecimento preliminar sobre jogos digitais.

Serão apresentados conceitos em relação ao gerenciamento de risco ao jogador de forma gradual e de acordo com a evolução do projeto, buscando solucionar e reduzir os problemas, com intuito de aumentar a eficiência e minimizar futuros prejuízos ao desenvolvimento. Serão realizados também:

- Realização da análise dos requisitos através de pesquisa sobre a aplicação dos métodos ágeis e a forma que é feito o gerenciamento de risco em suas aplicações;
- Criação e elaboração de questões de níveis fácil, intermediário e difícil de acordo com o Scrum e o gerenciamento de risco;
- Modelagem de um jogo para ensino do Scrum adaptando uma forma de gerenciamento de risco de forma mais eficaz;
- Construção e implementação do jogo modelado.

### 1.3 Organização do Documento

Este trabalho está dividido em 6 capítulos, organizado da seguinte forma: Capítulo 1 apresenta o contexto, justificativa e os objetivos deste trabalho. O capítulo 2 apresenta o Referencial Teórico com os conceitos essenciais que serão utilizados para a realização deste trabalho. O Capítulo 3 aborda os principais trabalhos relacionados ao tema de forma a contribuir para desenvolvimento deste trabalho e do jogo. O Capítulo 4 apresenta a metodologia que foi utilizada para a realização do trabalho proposto. O Capítulo 5 apresenta o resultados e apresentação das etapas de construção e desenvolvimento final do projeto. O Capítulo 6 é apresentado as considerações finais sobre este trabalho, e por fim são apresentadas as referências bibliográficas utilizadas neste trabalho.

## 2 Referencial Teórico

Buscando métodos e abordagens para facilitar e melhorar o gerenciamento de projetos, e com necessidade cada vez maior de introduzir inovações vem sendo testadas constantemente novas formas e adaptações de desenvolvimento de *software*. E neste contexto, onde esta crescendo a busca por novas formas e adaptações de técnicas, processos e ferramentas para o gerenciamento de projetos, que os métodos ágeis se destacam por suas abordagens e atividades simples e rápidas para gerência de projetos se comparadas com aos métodos tradicionais.

Neste capítulo serão abordados tópicos que serviram de embasamento para o estudo e pesquisa sobre de gerenciamento ágil e gerenciamento de risco, serão apresentados conceitos de gerenciamento de projetos conceituando principalmente o gerenciamento de risco, as características e principais atividades do método ágil Scrum. A gerência de risco segundo os principais métodos de qualidade apontando características e práticas relacionadas ao desenvolvimento, por fim, a utilização de jogos como meio de ensino.

### 2.1 Gerenciamento de Projetos

Um projeto é realizado para criar um produto, serviço ou resultado específico, caracterizado por tempo de desenvolvimento e resultado, serviço ou produto único e de elaboração progressiva ([PMI, 2013](#)). Projeto é um processo único, que consiste em um grupo de atividades controladas com datas para início e término, para alcance de um objetivo conforme especificações, incluindo limitações de tempo, custo e recursos NBR ISO 10006 ([ISO, 2000](#)).

O gerenciamento de Projetos é a aplicação de conhecimentos, habilidades e técnicas para a execução em projetos de forma eficiente. Com sua evolução e grande uso, houve o surgimento de organizações que oferecem certificações e guias de gerenciamento de projetos, como o PMI, onde foi criado o PMBOK ([PMI, 2013](#)).

Usado como um guia o PMBOK contém um conjunto de práticas e procedimentos divididos em dez áreas de conhecimento, que são elas: gerenciamento de integração, gerenciamento de escopo, gerenciamento de tempo, gerenciamento dos custos, gerenciamento da qualidade, gerenciamento das partes interessadas do projeto, gerenciamento dos recursos humanos, gerenciamento da comunicação, gerência de aquisições e gerência de riscos ([PMI, 2013](#)).

Criadas para servirem como guia no gerenciamento de projeto, e com exigências diferentes para os vários tipos de projetos, se faz necessário a adaptações para diferentes

projetos, para atender as necessidades específicas de cada um. Dentre esses novos métodos e adaptações eis que surgem os métodos de desenvolvimentos ágeis sendo a princípio voltadas a projetos pequenos, cujo objetivo é tornar o processo de gerenciamento de projetos simples, flexível e iterativo ([AMARAL et al., 2011](#)).

## 2.2 Gerenciamento de Risco

Geralmente os riscos em projetos são causados por uma condição que possam vir a acarretar efeitos negativos ou positivos ao projeto. Os riscos também são considerados como incertezas do projeto, uma vez que um risco seja considerado negativo, este então pode ser considerado um problema ([PMI, 2013](#)).

Com o ganho de cada vez mais importância em empresas e organizações o gerenciamento de riscos vem sendo utilizado em projetos para garantir o atendimento aos requisitos básicos do projeto como estipulação do escopo, prazo, qualidade do produto e no custo especificado ([BOYADJIAN, 2012](#)).

A gerência de risco só começou a receber mais atenção nos anos 80, quando Barry Boehm apresentou uma abordagem para gerenciar e identificar riscos ([BOEHM, 1991](#)). O modelo apresentado por Boehm está baseado no modelo espiral, também de sua autoria anos antes. O modelo espiral tem um ciclo de vida com a identificação dos objetivos relacionados e avaliação as alternativas identificadas na implementação dos objetivos. No processo onde são identificados os riscos do projeto, uma vez analisados e avaliados, será necessário definir formas de tratá-los. Usando várias técnicas e atividades como simulações e prototipagem são realizadas para avaliação por todos envolvidos no projeto ([PMI, 2013](#)).

O gerenciamento de riscos é um processo que tem como objetivo identificar, analisar e reagir aos riscos de um projeto, diminuindo ou até eliminando um impacto ao projeto, ele também se preocupa em aumentar a probabilidade de um evento positivo ([PMI, 2013](#)). O gerenciamento de riscos é uma atividade contínua, sendo uma das principais fases do desenvolvimento que atua durante todo o ciclo de vida do projeto de forma a antecipar, minimizar e eliminar os riscos do projeto.

O Gerenciamento de Riscos esta dentro um conjunto de atividades que devem ocorrer no inicio e durante a identificação, análise, planejamento, monitoramento e o controle do projeto desenvolvido ([SOMMERVILLE; ARAKAKI; MELNIKOFF, 2008](#)).

A prática de Gerenciamento de Riscos tem como vários benefícios ajudar no planejamento do projeto, na criação corretamente de cronogramas, estimativas de prazos e custos realistas com desenvolvimento da equipe, envolvendo toda a equipe na definição de pontos principais do projeto e integrar todas as áreas de conhecimento no Gerenciamento de Projetos ([SOMMERVILLE; ARAKAKI; MELNIKOFF, 2008](#)).

O monitoramento de riscos é uma atividade de verificação e avaliação continuas dos riscos já identificados, para decidir se esses riscos estão se tornando mais ou menos prováveis. Em cada revisão deve-se considerar e discutir cada um dos principais riscos separadamente, para decidir se os riscos são mais ou menos suscetíveis de surgirem no decorrer do desenvolvimento e suas consequências ([SOMMERVILLE; ARAKAKI; MELNIKOFF, 2008](#)).

Para um gerenciamento de riscos efetivo deve ser incluída a identificação antecipada de riscos, através da colaboração e envolvimento de toda equipe buscando maior entendimento dos problemas e necessidades do projeto, viabilizando e estabelecendo um planejamento de solução, de novas atividades dentro do processo e discussão dos riscos enfrentados ([CAMPOS, 2009](#)).

### 2.2.1 Gerenciamento de risco segundo método ágil Scrum

O Gerenciamento de Riscos teve o modelo espiral com um de seus precursores ([BOEHM, 1991](#)). Modelo que inspirou muitas abordagens e métodos como: o modelo CMMI e o Guia PMBOK ([PMI, 2013](#)).

O método ágil Scrum é uma das abordagens mais conhecidas para o gerenciamento de projetos e define um conjunto de práticas e atividades a serem aplicadas através de ciclos iterativos, o gerenciamento de riscos, também é tratado de forma incremental nos projetos que utilizam o Scrum.

Os processos do Scrum se dividem em iterações no *Sprint*, nele a identificação, análise, monitoramento e respostas aos eventuais riscos são realizados iterativamente e continuamente durante as reuniões de planejamento de cada iteração ([SCHWABER; SUTHERLAND, 2013](#)). O monitoramento dos riscos é feito durante a reavaliação nas reuniões, onde os riscos são analisados e reportados ao Scrum *Master* para que sejam eliminados para as próximas iterações.

Apesar de que o Gerenciamento de Riscos no Scrum não ser considerada tão eficaz quanto nas práticas de gerenciamento mais tradicionais ([MARÇAL et al., 2007](#)). Segundo [Rocha e Belchior \(2004\)](#) esta complexidade faz com que grande parte dos projetos de desenvolvimento de *software* exceda o prazo e o orçamento previstos, além de não atender às expectativas do cliente em termos de funcionalidades e qualidade. Diante deste cenário, um gerenciamento de risco eficaz tem se tornado de extrema para o sucesso desses projetos.

Destacando algumas práticas utilizadas no Gerenciamento de Riscos tradicionais os métodos ágeis não atendem de forma eficiente, exceto a atividade de identificação de riscos, e por não existir de maneira explícita, práticas em definir parâmetros ou categorias de risco dentre todos encontrados no processo de identificação ([MARÇAL et al., 2007](#)).

O Scrum tem uma estratégia para o Gerenciamento dos Riscos implicitamente

por toda a equipe de desenvolvimento em forma de *checklist* em suas reuniões diárias reportando assim a todos os membros da equipe os possíveis riscos encontrados, para que sejam solucionados o mais rápido possível pela própria equipe ou pelo Scrum Master (SCHWABER; SUTHERLAND, 2013).

Porém não há uma forma de avaliação daqueles riscos que foram apontados para que seja tratados de formas diferentes com mais ou menos prioridades dos demais encontrados, A Gerência de Projetos se preocupa com a distribuição e controle do esforço e dos recursos para o projeto, mas neste contexto, as organizações às vezes negligenciam o gerenciamento de risco e os futuros empecilhos que podem surgir no desenvolvimento do projetos (GUSMÃO, 2009).

Apesar de não fornecer uma documentação a respeito do risco encontrado o Scrum entende-se de maneira implícita que os prováveis riscos são eliminados rapidamente independente de sua prioridade.

### 2.2.2 Gerenciamento de risco regundo PMI-PMBOK

De Acordo com PMBOK PMI (2013) o risco em projetos é um evento ou condição incerta que, que se caso ocorra, causará um efeito positivo ou negativo no projeto. Se algum desses eventos incertos ocorrer, pode haver um impacto no escopo, custo, cronograma ou na qualidade final do projeto.

Práticas que podem contribuir para a ocorrência de riscos em projeto, são práticas imaturas de gerenciamento de projetos, projetos simultâneos ou dependência interna de algum membro da equipe.

Algumas organizações têm o risco como uma incerteza nos projetos e objetivos, elas estão dispostas a conviver com estes riscos, dependendo da atitude em relação aos riscos para serem mais benéficos futuramente. Esta atitude é influenciada por vários fatores, classificados em três tipos: Apetite de risco, que é o grau que a organização está disposta a aceitar o risco; Tolerância a riscos, volume de risco que a organização está disposta a suportar e Limite de riscos, que são as medidas ao impacto no qual a organização esta interessada ela aceitará o risco até aquele limite (PMI, 2013).

O Gerenciamento dos riscos do projeto deve incluir processos de planejamento, identificação, análise, planejamento de respostas e controle de riscos de um projeto (PMI, 2013).

Seguindo os passos do PMBOK PMI (2013) define seis processos para a gerência de risco, são eles: Planejamento da gerência dos riscos, que é a definição do processo que ira conduzir as atividades do projeto; Identificação dos riscos, processo que determina e documento se os riscos podem afetar o projeto e de quais formas; Análise qualitativa dos riscos, verificação da prioridade dos riscos para uma ação posterior e probabilidade de

ocorrência e seu impacto; Análise quantitativa dos riscos, verificação da quantidade real e o efeito dos riscos identificados no projeto; Respostas aos riscos, Planejamento de ações para reduzir ameaças aos objetivos do projeto; Controlar os riscos, Implementação de planos de respostas aos riscos encontrados, acompanhados, monitoração riscos restantes, identificação de novos riscos e avaliação da eficácia do processo de gerenciamento dos riscos durante todo o desenvolvimento do projeto PMI (2013), A Figura 1 apresenta uma visão geral dos processos.

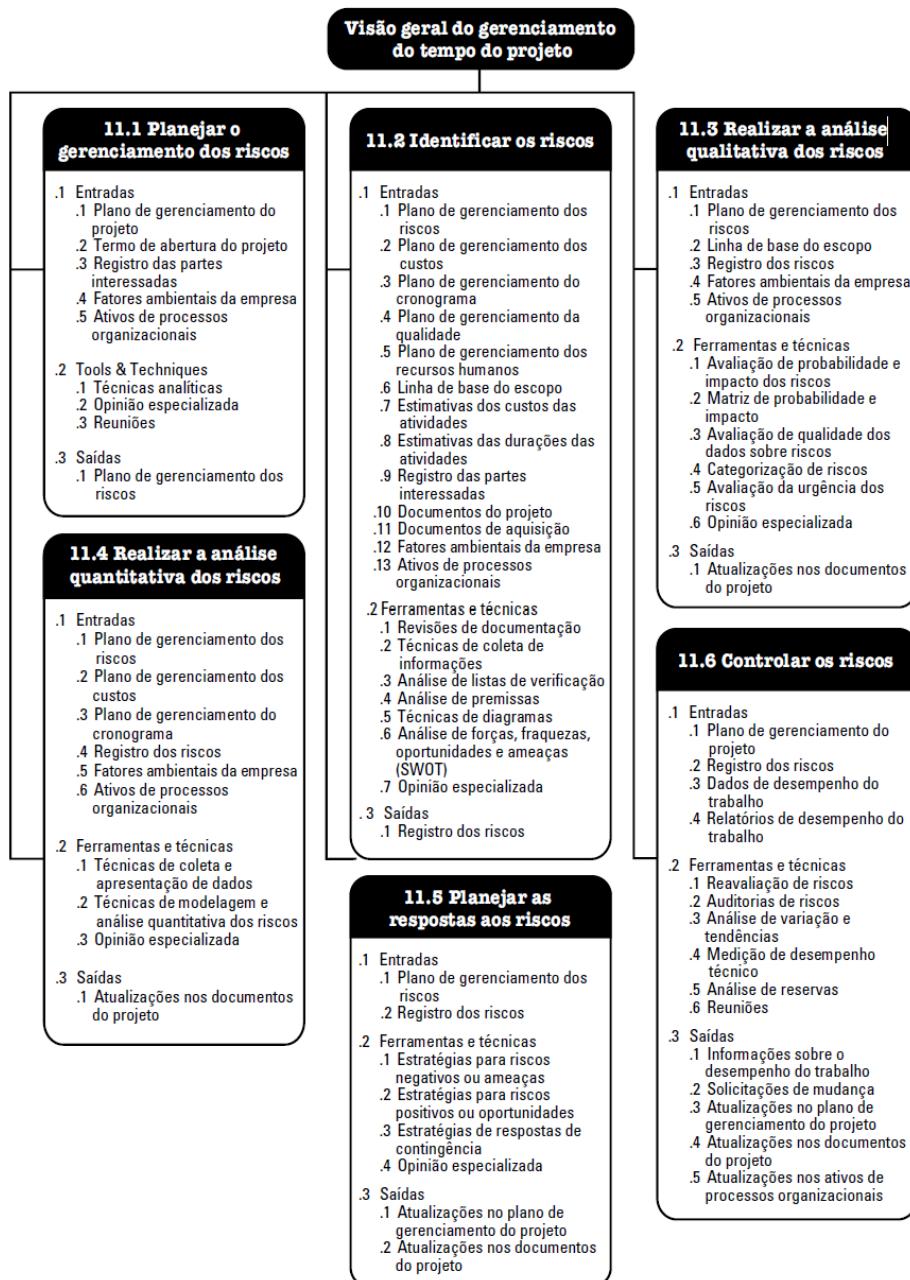


Figura 1 – Visão Geral do Gerenciamento de risco do guia PMBOK

Fonte: PMI 2013

### 2.2.3 Gerenciamento de risco segundo CMMI

Segundo o CMMI [Sei \(2006\)](#) o gerenciamento de risco é um meio de fornecer formas de identificar e prever problemas de forma que as próximas atividades para o tratamento possam ser planejadas e aplicadas ao longo do ciclo de vida do desenvolvimento do projeto.

Para a gerência de risco deve considerar fontes de riscos internas e externas para ampliar e antecipar a detecção de riscos. O gerenciamento de riscos é dividido em três partes: definição da estratégia de riscos, identificação e análise de riscos e tratamento de riscos ([SEI, 2006](#)).

O gerenciamento de riscos é realizado para estabelecer uma estratégia para identificação, análise e tratamento dos riscos, sendo tudo isso documentado em um plano de gestão de riscos, com intuito de ser utilizado para controlar o processo, incluindo a identificação das fontes em que se originam os riscos, também é utilizado parâmetros para categorizá-los sendo utilizados para analisar, avaliar e controlar os riscos ([SEI, 2006](#)).

O CMMI [Sei \(2006\)](#) destaca que a identificação de fontes dos riscos fornece uma base para serem examinados, de forma consistente em situações de mudanças ao longo do desenvolvimento, a fim de descobrir quais as possibilidades possam impactar no projeto. Eles são documentados incluindo as condições e as consequências de sua ocorrência. A identificação de riscos deve ser de forma organizada e abrangente para levantar prováveis riscos que possam comprometer o desenvolvimento do projeto.

Com um meio de organizá-los em meio a tantos problemas e empecilhos que podem ocorrer há o estabelecimento de categorias de riscos fornecendo um meio para assegurar uma análise e verificação mais detalhada sobre os riscos que podem ter consequências no desenvolvimento do projeto, a lista de riscos deve ser repassada continuamente para poder reexaminar as possíveis fontes de riscos e condições de mudança, para poder identificar fontes de riscos ainda presentes ([SEI, 2006](#)).

Após examinados e verificados os possíveis riscos, a análise e avaliação de cada risco agora é feito para determinar a probabilidade e sua consequência ao projeto. A categorização do risco, baseada nas categorias de riscos estabelecidas nos critérios da estratégia feita anteriormente, fornece as informações necessárias para o tratamento dos riscos ([SEI, 2006](#)).

Para controlar com mais eficiência os riscos, é preciso monitorar regularmente os riscos e os resultados gerados após todas as ações de tratamento. O CMMI [Sei \(2006\)](#) define que se aplicada sucessivamente a verificação e monitoramento dos riscos, no qual o status dos riscos deve ser reavaliado para definir ações para novos riscos.

## 2.2.4 Gerenciamento de Risco Segundo MPS.BR

A princípio o MPS.BR tem o gerenciamento de riscos como uma forma de identificar, analisar, tratar, monitorar e reduzir os riscos do projeto ([GERAL, 2012](#)).

O processo de gerenciamento de risco é aplicado a todos os riscos presentes no projeto sejam internos ou externos. Este processo engloba as atividades de identificação, controle dos riscos, em conjunto com as ações de mitigação e redução, a fim de garantir a redução de riscos e de seu impacto no desenvolvimento de projetos ([GERAL, 2012](#)).

Segundo o guia MPS.BR o planejamento e monitoração dos riscos são iniciados no nível G do processo de gerência de projetos, embora o foco da gerência de riscos no nível C acrescenta vários aspectos diferentes, como a necessidade de planos de intervenção e de monitoramento, buscando avaliar as situações dos riscos e do progresso das atividades desempenhadas do tratamento de risco ([GERAL, 2012](#)).

Para facilitar e garantir por completo a identificação de possíveis riscos deve-se determinar uma classificação de gravidade dos riscos observados, que podem envolver desde a tecnologia empregada, equipe de desenvolvimento, custo final e seu cronograma.

Para definição e classificação de riscos a própria experiência da equipe ou de algum especialista pode servir como parâmetro e definir a classificação de risco a ser utilizada na organização baseada em origens e categorias ([IMPLEMENTAÇÃO-PARTE; NÍVEL, 2007](#)). São geradas a partir da classificação as estimativas de sua probabilidade de impacto, que na maioria das vezes são feitas de maneira qualitativa, esses dados fornece um critério rígido para a priorização dos riscos verificados.

No nível G, não é necessário a definição de ações para redução dos efeitos dos riscos, o gerenciamento de risco só é aprofundado pelo MPS.BR a partir da evolução para o nível C do MR-MPS-SW, pois e a partir de então e implementado três novos processos que são eles: Gerência de Decisões (GDE), Desenvolvimento para Reutilização (DRU) e Gerência de Riscos (GRI) ([SERVIÇOS, 2012](#)).

De acordo com o MPS.BR existem 9 resultados esperados para o gerência de risco para o nível C.

- GRI 1 - O escopo da gerência de risco é determinado; Onde o gerenciamento de risco é aplicado dentro do desenvolvimento de projetos, mas também devem ser aplicado para os processos dentro da organização. Deve-se determinar visivelmente a dimensão da aplicação no processo de gerenciamento.

- GRI 2 - As origens e as categorias de riscos são determinadas e os parâmetros usados para analisar riscos, categorizá-los e controlar o esforço da gerência de riscos são definidos; Para simplificar e assegurar a plenitude da identificação de possíveis empecilhos, assim como para garantir uma maior isonomia de análise, deve definir a classificação e parâmetros para indicação da probabilidade e da rigidez dos riscos encontrados.

- GRI3 - As estratégias apropriadas para a gerência de riscos são definidas e implementadas; Um plano de gerência de riscos deve ser definida, comparando características desde o escopo aos métodos e ferramentas utilizados na identificação, análise e monitoração dos riscos.
- GRI4 - Os riscos do projeto são identificados e documentados, incluindo seu contexto, condições e possíveis consequências para o projeto e as partes interessadas; O processo de reconhecimento dos riscos pode fazer uso de vários métodos, como o uso de *checklists*, reuniões com gerente e equipe de desenvolvimento, análise de situações e através de material aprendido em projetos anteriores.
- GRI5 - Os riscos são priorizados, estimados e classificados de acordo com as categorias e os parâmetros definidos; Após apontar os riscos é será preciso organizá-los em categorias e definir sua prioridade. Determinando a probabilidade seus efeitos baseados nos parâmetros definidos anteriormente.
- GRI6 - Planos para a mitigação de riscos são desenvolvidos; A fim de aliviar os riscos, diminuindo a possibilidade de ocorrência ou enfraquecer os efeitos causados pelos riscos, deve-se estabelecer um plano para os riscos de maior prioridade.
- GRI7 - Os riscos são analisados e a prioridade de aplicação dos recursos para o monitoramento desses riscos é determinada; Garantir que os riscos serão tratados pela gerência de riscos sejam escolhidos após a análise que determinou a prioridade dos riscos para aplicação dos meios necessários para sua monitoração.
- GRI8 - Os riscos são avaliados e monitorados para determinar mudanças em sua situação e no progresso das atividades para seu tratamento; O plano de gerenciamento de risco deve ser seguido, garantindo que os riscos sejam monitorados e avaliados durante todo o desenvolvimento e que os planos de contingência definidos em GRI 6 sejam executados de maneira eficiente.
- GRI9 - Ações apropriadas são executadas para corrigir ou evitar o impacto do risco, baseadas na sua prioridade, probabilidade, consequência ou outros parâmetros definidos; Durante o desenvolvimento deve-se monitorar e verificar a tem obrigação da execução de ações de imprevistos para os riscos, de acordo com o plano traçado.

## 2.3 Métodos Ágeis

Segundo [AMARAL et al. \(2011\)](#), o gerenciamento ágil é uma abordagem alternativa para o desenvolvimento de projetos, que visa tornar o processo mais simples, flexível e iterativo, com objetivo a obtenção de melhores resultados no desempenho, menor esforço de gerenciamento e maiores níveis de inovação e agregação de valor a equipe e ao cliente.

A metodologia ágil pode ser definida como métodos de desenvolvimento itera-

tivo incremental, em que são desenvolvidos pequenas partes para serem incrementados no decorrer das etapas do desenvolvimento. Esses incrementos, normalmente, resultam em pequenas novas versões para serem realizados alguns testes e verificações em curtos espaços de tempo, com objetivo de demonstrar a evolução dos requisitos desse produto. Dessa forma, é possível minimizar as necessidades da documentação através do uso da comunicação informal ([SOMMERSVILLE; ARAKAKI; MELNIKOFF, 2008](#)).

As metodologias ágeis se diferenciam das metodologias tradicionais, principalmente porque geram menos documentos. Métodos ágeis são mais adaptativos do que previdentes, acolhem a mudança a qualquer momento, e adaptar a própria metodologia para serem bem sucedidos. Baseia-se principalmente nas pessoas do que nos processos, e asseguram que nenhum processo pode superar as habilidades de uma equipe. Desta forma, o papel do processo é dar suporte à equipe de desenvolvimento do projeto ([FOWLER, 2001](#)).

As metodologias ágeis têm sido muito utilizadas em vários tipos de desenvolvimento de projeto, como o desenvolvimento de pequenos e médios projetos que não necessitem de uma grande alocação de recursos, e o desenvolvimento para uma organização ou público em geral desde que havendo uma opinião constante para guiar a evolução do projeto ([SOMMERSVILLE; ARAKAKI; MELNIKOFF, 2008](#)).

A agilidade não é apenas uma resposta às alterações que ocorrem no decorrer de um projeto ([PRESSMAN, 2006](#)). Há também incluso a forma de manifesto ágil, encorajando toda equipe, facilitando a comunicação, tornando o cliente uma parte da equipe de desenvolvimento para o projeto possa ser muito mais flexível, estável e com menos mudanças no decorrer do desenvolvimento.

As metodologias ágeis têm como princípio, de que o processo de desenvolvimento de *software* é empírico, baseado na visibilidade das características e resultados do processo e em disponibilizar maneiras de monitorar e corrigir o processo e o projeto. Dessa forma, processos empíricos têm a possibilidade de aceitar falhas como uma consequência da produção e tentam torná-las passíveis de correção o mais rápido ([PRESSMAN, 2006](#)).

Dentre os métodos ágeis existentes foi escolhido o Scrum para ser implementado poís tem se destacado por se concentrar em práticas e atividades de gerenciamento de projetos, reunindo ações de monitoramento e *feedback* que visam identificar e corrigir deficiências e impedimentos no processo de desenvolvimento.

## 2.4 Scrum

Scrum é um processo de desenvolvimento iterativo e incremental, surgiu nos anos 80, porém sua disseminação só ocorreu no início dos anos 90 ([TAKEUCHI E NONAKA, 1986](#)). Através de estudos comparando o desenvolvimento ágil ao processo de desenvolvimento clássico aplicado na época ([PRESSMAN, 2006](#)). Sendo possível sua utilização em

diversos tipos de projetos e ser utilizado em conjunto ou adaptado com outros métodos ([OLIVEIRA; GUIMARÃES; FONSECA, 2007](#)).

O conceito de agilidade em desenvolvimento de projetos vem através da criação de times multidisciplinares, que deveriam trabalhar em conjunto, prezando pela comunicação entre seus membros e o foco em uma meta compartilhada ([TAKEUCHI, 1986](#)).

O Scrum é um método para a gerência de projetos de desenvolvimento em que trabalha com curtas iterações no desenvolvimento, que ao final das iterações é adicionado ao produto como correção ou como nova funcionalidade. O Scrum envolve e relaciona valores e práticas do gerenciamento de projetos, sem definir práticas específicas para o desenvolvimento, testes ou de requisitos ([SCHWABER, 2004](#)).

Apresentado como uma abordagem empírica que aplica algumas idéias da teoria de controle de processos para o desenvolvimento de *software*, introduzindo idéias mais flexíveis e adaptativas. O foco da metodologia é encontrar uma forma de trabalho dos membros da equipe para produzir o *software* de forma flexível e em um ambiente em constante mudança ([SCHWABER, 2004](#)).

Segundo [Schwaber e Sutherland \(2013\)](#) o Scrum possui três pilares: A visibilidade, que garante que os requisitos ou riscos do processo que afetam os resultados devem estar visíveis e serem entendidos por todos que gerenciam os resultados; A inspeção e monitoramento, em que todos os possíveis aspectos do processo devem ser inspecionados e monitorados para que variações e empecilhos futuros não sejam aceitos no processo e que possam ser detectadas. A adaptação, que de forma que se inspeciona o processo a determinar, a partir do monitoramento, que um ou mais aspectos do processo deverá ajustar o processo atual o mais rápido possível, de forma a minimizar desvios posteriores.

De acordo com [Schwaber \(2004\)](#) o método se baseia em princípios como: equipes pequenas, requisitos poucos estáveis ou até desconhecidos e iterações incrementais curtas. Dividindo o desenvolvimento em intervalos de tempos no máximo quatro semanas, também chamadas de *Sprints*. Ele apenas estabelece um conjuntos de regras e práticas que devem ser adotadas para o sucesso do projeto ([PRESSMAN, 2006](#)).

#### 2.4.1 Time Scrum

O Time Scrum é toda equipe de desenvolvimento envolvida do produto, em que todos trabalham junto no projeto a fim de completar o conjunto de trabalho que se comprometeram a fazer em determinado projeto. O Time Scrum geralmente é composto pelo *Product Owner*, o time de desenvolvimento e o *Scrum Master*. O time Scrum por ser extremamente multifuncional escolhe qual é a melhor forma para completar seu trabalho, não sendo necessário ser dirigidos por gerentes de processos ou outro personagem. Este modelo de equipe foi projetado para serem mais flexíveis e criativos, buscando a maior produtividade.

vidade e entrega de produtos em curto espaço de tempo (SCHWABER; SUTHERLAND, 2013).

#### 2.4.2 Product Owner

O *Product Owner* é a pessoa responsável pelo gerenciamento e em definir quais itens deverá ser incluído o *Backlog* do produto, definição e articulação de metas e missões, aumentar e valorizar o produto e do time de desenvolvimento, responsável pelo entendimento de todo o time de desenvolvimento deixando de forma transparente todos os requisitos do *Backlog* do produto (SCHWABER; SUTHERLAND, 2013).

#### 2.4.3 Scrum Master

O Scrum *Master* geralmente pode ser qualquer pessoa da equipe, um gerente de projeto ou algum responsável técnico. O Scrum *Master* considerado o líder de toda a equipe, é responsável por garantir que toda a equipe siga as práticas do Scrum, ajudar que estão com dificuldades no entendimento do projeto, responsável em ajudar e maximizar iterações desenvolvidas pelo time Scrum, compreender a longo prazo o planejamento de produto, treinar e liderar a equipe de desenvolvimento na criação e agregação de valor ao produto, remover quaisquer empecilhos que sejam levantados pela equipe durante as reuniões (SCHWABER; SUTHERLAND, 2013).

#### 2.4.4 Backlog do Produto

O *Backlog* do Produto é uma lista de todos os requisitos, características, funções, melhorias e correções necessárias para o produto, os primeiros *Backlog* desenvolvidos são apenas para serem conhecidos e entendidos pela equipe. O *Backlog* é um artefato dinâmico com mudanças constante. Com toda equipe buscando identificar formas necessárias para serem o mais útil possível, visto que o *Backlog* do projeto nunca estará completo podendo sofrer alterações desde mudanças nos requisitos do negócio, custos, e em alterações do mercado (SCHWABER; SUTHERLAND, 2013) .

#### 2.4.5 Sprint

No Scrum, os projetos são divididos em ciclos, variando de duas a quatro semanas chamados de *Sprint*. O *Sprint* é um conjunto de atividades que deve ser executado em determinado período. O Scrum por seu um método iterativo a cada ciclo é gerado uma versão incrementada e funcional do produto (SCHWABER; SUTHERLAND, 2013).

As Sprints são compostas por: reunião de planejamento inicial, reuniões diárias para informar ao Scrum *Master* e toda a equipe do andamento do desenvolvimento e

empecilhos encontrados no projeto, revisão do *Backlog* do produto e ao final uma retrospectiva da *Sprint* (SCHWABER; SUTHERLAND, 2013).

#### 2.4.6 Definição dos Processos do Scrum

Para dar inicio ao processo Scrum, primeiramente é preciso definir a equipe que irá compor o Time Scrum, esta equipe não deve ter muitos membros para que não possa haver desorganização e falta de comunicação entre os membros, caso tenha muitos membros pode se separar em mais equipes, ficando então cada equipe responsável por uma área específica do trabalho.

Com a escolha do Scrum *Master* que será a pessoa que irá conduzir o time responsável por medir o progresso do desenvolvimento, e toma as principais decisões envolvendo o projeto. O Scrum *Master* fica encarregado de sanar todas as duvidas e excluir possíveis desentendimentos da equipe (FERREIRA et al., 2017) (SCHWABER; SUTHERLAND, 2013).

Para os materiais que serem disponibilizados para a equipe, existem quadros de reuniões, marcadores, *post-it*, impressões do *Backlog* do produto e cartões de referência para cada para cada membro a fim de estimular e ter acesso fácil a informação do desenvolvimento (CARVALHO; ABRANTES; CAMEIRA, 2011).

Durante cada *Sprint*, na Figura 2 apresenta o ciclo de vida Scrum destacando o inicio do *Sprint*, *Backlog* do produto e onde são realizadas reuniões diárias para permitir que o time de desenvolvimento possa se comunicar e trocar conhecimento entre cada membro do time (FERREIRA et al., 2017). Nas reuniões quinzenais ou semanais, a equipe se dedica a revisões de etapas anteriores e do todo o projeto, a fim de ajustar, prever, monitorar e aperfeiçoar os processos do projeto (SBROCCO; MACEDO, 2012).

Tendo como ponto de partida a *Sprints Planning* em que na primeira parte da reunião são feitos os requisitos registrados no *Backlog* do produto onde serão desenvolvidos ao longo do *Sprint*. Na segunda parte, são discutidas e detalhadas as atividades envolvidas na execução de cada item selecionado dos requisitos, sendo que após a segunda etapa é dado inicio à fase de execução das atividades do *Sprint*.

No desenvolvimento de *software*, o *Sprint* inclui fases tradicionais do desenvolvimento de *software* como: requisitos, análise, projeto e entrega (CARVALHO; ABRANTES; CAMEIRA, 2011)

Na identificação do *Backlog*, é feito uma lista de todo o trabalho conhecido a fazer e separar em partes para incrementar que não devem ultrapassar 30 dias e definir prioridades para todos os elementos listados. Com o *Backlog* criado cada grupo de desenvolvimento deve cumprir o *Sprint* de acordo com cada etapa prevista (SCHWABER; SUTHERLAND, 2013).

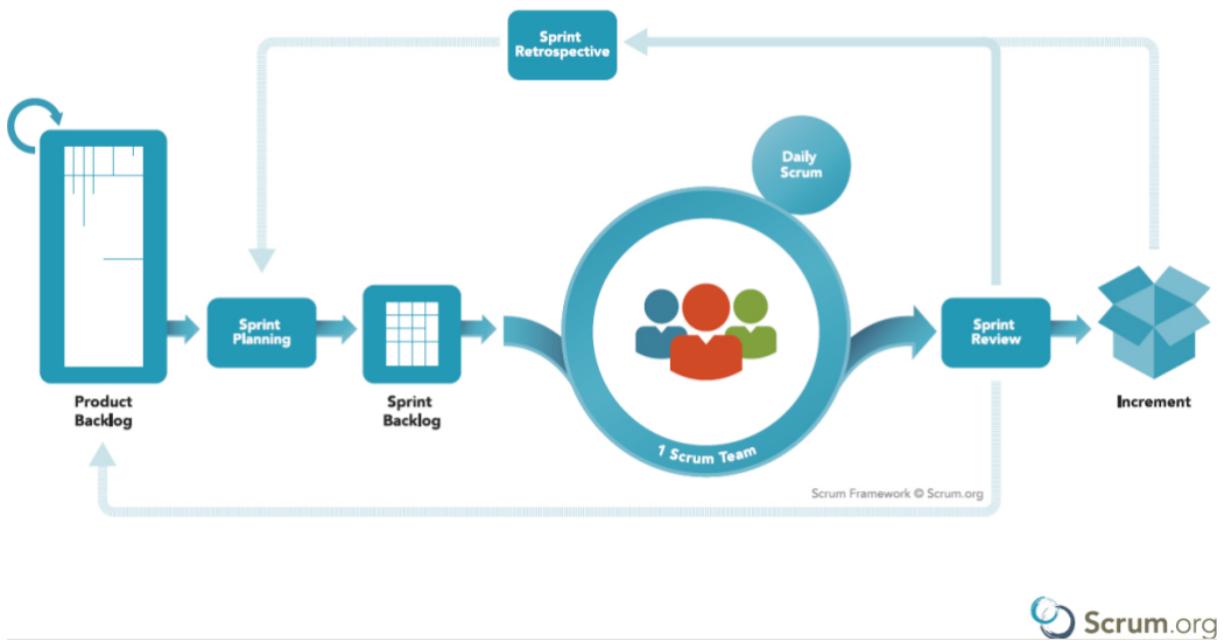


Figura 2 – Visão Geral do Ciclo de vida Scrum

Fonte: <[www.scrum.org](http://www.scrum.org)>

Toda a equipe que trabalhou na produção do *Sprint* deve demonstrar algo ao fim do *Sprint*, apresentar os resultados que a equipe tiveram ao completar o seu planejamento através do *Sprint*, por meio de um *Sprint Backlog* num quadro de reuniões ([SCHWABER; SUTHERLAND, 2013](#)).

Martins (2016) e com base em práticas de Gerenciamento de Riscos abordadas pelo SCRUM e no planejamento de Gerência de Riscos pelo PMBOK, para adaptação Scrum original, conforme apresentado na Figura 3 utilizando o SPEM, para a modelagem e criação um modelo que visa incluir algumas práticas de gerenciamento de riscos consideradas essenciais, com o objetivo de ajudar no desenvolvimento de projetos, melhorar a visualização, identificação e a compreensão das práticas de gerenciamento de riscos que ficam implícitas no método Scrum, com a utilização da gerência de riscos ainda está aquém das expectativas..

De acordo com o PMBOK ([SEI, 2006](#)) é importante identificar os riscos já no grupo de Iniciação, quando se faz uma definição preliminar do escopo e dos riscos.

Com o objetivo de rever o planejamento e de planejar a nova *Sprint*. O retorno à fase do Pré-game pode ocorrer quando forem identificadas as mudanças muito significativas no escopo como mostra a Figura 4.

Na fase do Pré-Game os critérios para o gerenciamento de riscos devem ser es-

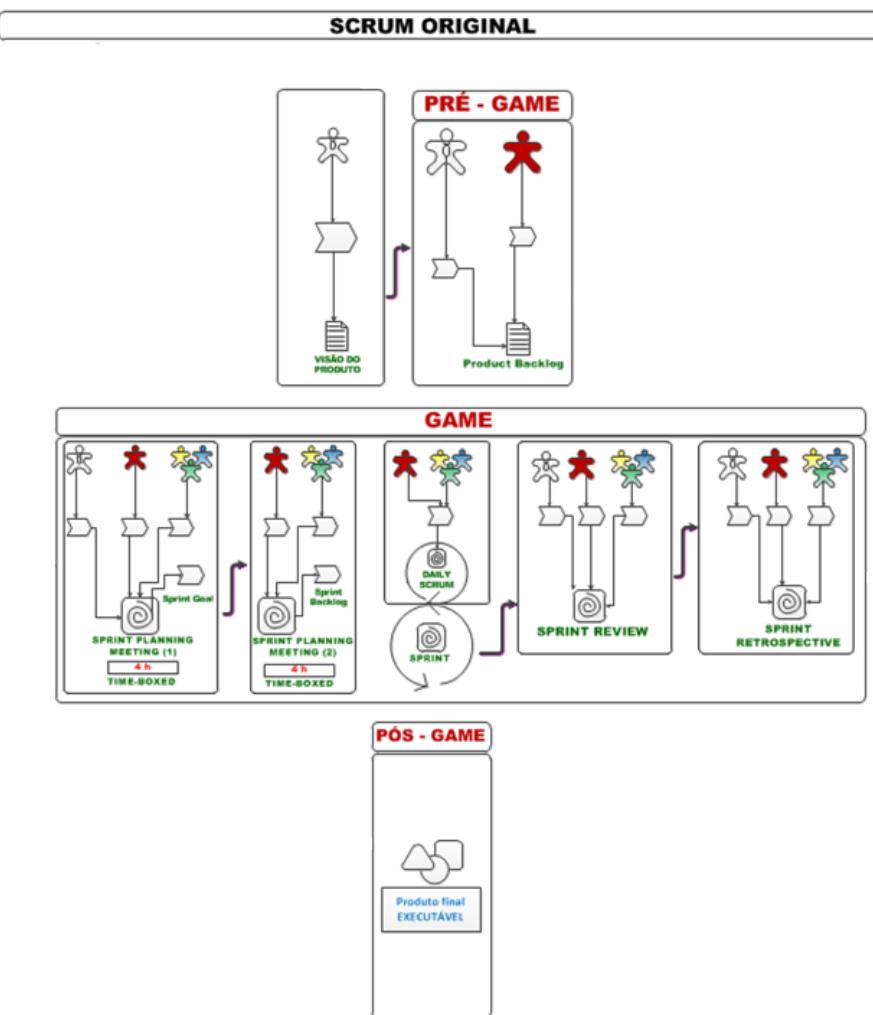


Figura 3 – Scrum original modelado no SPEM

Fonte: ([MARTINS, 2016](#))

tabelecidos. Com a criação de uma lista de riscos de acordo com o grau em que o risco possa o vir a prejudicar no desenvolvimento do projeto, afim de que todos possam adotar o mesmo critério e aplicá-los de forma padronizada. Assim, o *Product Owner* deve identificar os riscos e os analisar quantitativamente e qualitativamente. A partir dos resultados desta análise, devem-se planejar respostas para tais riscos, além de identificar ativos, ameaças e controles existentes. O Scrum *Master* fica responsável por identificar os riscos relacionados aos recursos humano e ao espaço físico ([MARTINS, 2007](#)) ([MARTINS, 2016](#)) ([GUIMARÃES; ROSA, 2016](#)).

Na fase de planejamento define-se o plano do projeto e um plano dos riscos. Essa abordagem foi adaptada e completada por este modelo na Fase do Game, nesta fase é feita uma identificação e análise dos riscos. São gerados os artefatos: lista de riscos obtidos

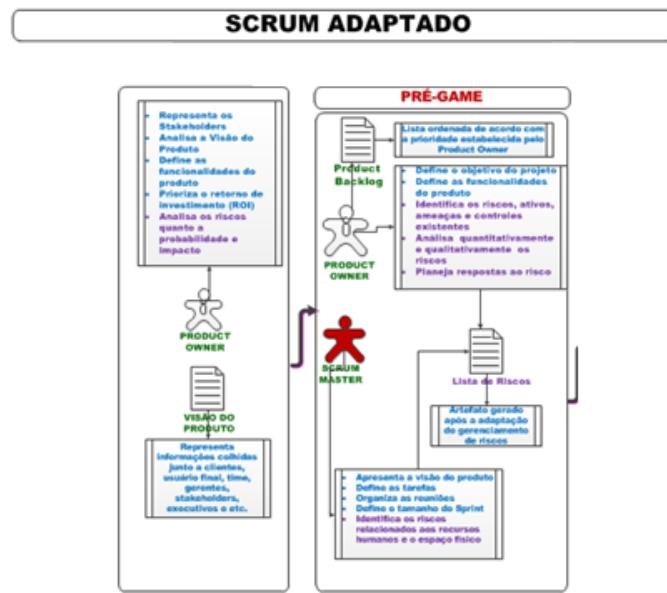


Figura 4 – Scrum adaptado modelado no SPEM na fase Pré-Game

Fonte: ([MARTINS, 2016](#))

nas reuniões de planejamento de *Sprint*, inicialmente como resultado da primeira reunião do *Sprint* é obtida a lista de riscos, essa lista é atualizada a cada iteração e gerada a lista de impedimentos como mostrado na Figura 5 ([MARTINS, 2016](#)).

Na reunião de planejamento, chamada de *Sprint Planning Meeting* são acrescentadas as atividades de identificação com o foco nos itens do *Product Backlog* com maior valor de negócio. A identificação pode ser feita através de um *checklist*, visando facilitar tarefas, contendo fontes e categorias de riscos. Ainda durante a *Sprint Planning Meeting* a análise de riscos é realizada pelo *Scrum Master* compreendendo etapas para determinar o impacto (baixo, médio e alto), sua probabilidade (baixa, média e alta) e o fator de exposição ao risco. A análise de impacto e probabilidade é utilizada para estabelecer a prioridade dos riscos e se criar uma lista de riscos de acordo com sua prioridade ([MARTINS, 2016](#)).

Na Figura 6 é ressaltado a importância na identificação e análise dos riscos que ocorrem a cada *Sprint*, ou seja, o gerenciamento de riscos é iterativo e o encerramento é caracterizado pelo Pós-Game gerando um artefato histórico de projetos esse histórico contempla o documento de riscos atualizados ([MARTINS, 2016](#)).

Na fase de Pós-Game deve-se incluir a base de dados de históricos de projetos com os dados atualizados do projeto e nesse contexto os dados da Gestão de Riscos, para que os mesmos possam ser considerados na estimativa de novas *Sprints* e de outros projetos. Ao final de todo o ciclo revisado e avaliado, e com um produto funcional ([MARTINS,](#)

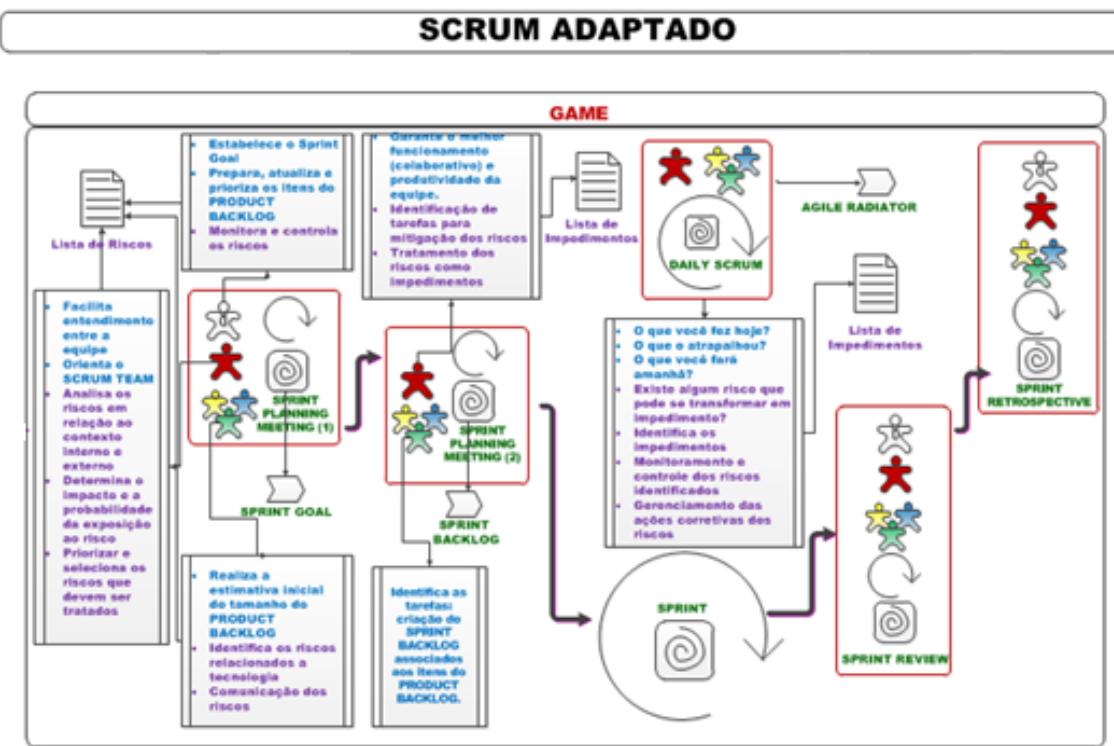


Figura 5 – Scrum adaptado modelado no SPEM na fase Game

Fonte: Martins 2016

2016) (GUIMARÃES; ROSA, 2016).

## 2.5 Desenvolvimento de Jogos

Jogos educacionais são mecanismos que retratam a realidade e forma interativa que levam os participantes a praticarem tarefas de alta capacidade cognitiva possibilitando o desenvolvimento de habilidades sem qualquer risco e quase sempre sem custo, através do uso de programas e *software* computadorizados (TEIXEIRA; JESUS; FERNANDES, 2007).

Jogos educacionais são criados para qualquer atividade desde que tenha um formato adequado para a aprendizagem ou que envolve algum meio de que se aprenda com o seu uso, envolvendo o competidor em uma disputa que seja guiada por regras e restrições, estimulando o raciocínio em busca de uma solução para o problema enfrentado (PRENSKY, 2010).

Os jogos podem ser ferramentas para o ensino muito eficientes, pois além de divertidos e interessantes eles são motivadores, fazendo que sejam ampliadas a criativi-

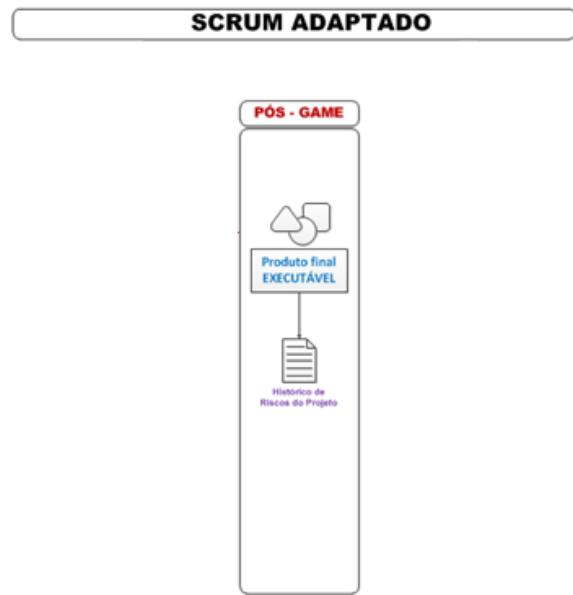


Figura 6 – Scrum adaptado modelado no SPEM na fase Pós-Game

Fonte: ([MARTINS, 2016](#))

dade e vontade de competir das pessoas facilitando o aprendizado ([OLIVEIRA, 2014](#)) ([PRENSKY, 2010](#)).

Como principal benefício da apresentação de um jogo para a aprendizagem tem o desenvolvimento de capacidades intelectuais, em conjunto com a percepção da realidade das disciplinas teóricas em sala de aula ([PRENSKY, 2010](#)).

De acordo [Prensky \(2010\)](#) para que os jogos possam ser descritos como jogos educativos, eles devem possuir algumas características essenciais:

- Regras: é o que distinguem os jogos educativos dos outros tipos, pois elas determinam limites e forçam os competidores a seguir por caminhos específicos para alcançar os propósitos e garantir que os próximos jogadores também passem pelo mesmo caminho.
- Metas ou Objetivos: São os alvos principais do jogo, em que o jogador está focado em seguir todo o caminho para a realização de eventos para alcançar os objetivos propostos e bater as metas;
- Resultados e *Feedback*: é tendo em visto seus resultados em cada meta e objetivos conquistados que o jogo tem parâmetros para medir se o jogador obteve êxito em todo caminho percorrido, pois se chegou até o fim, passando por todos desafios ele pode ter adquirido algum tipo de conhecimento;
- Conflito/Competição/Desafio/Oposição: São questões que o jogador deve resolver durante todo o jogo, em busca de ultrapassar barreiras e desafios para alcançar os

objetivos;

- Interação: Um jogo para retratar um ambiente real, busca um interação com o jogador, fornecendo *Feedback* em certos momentos;
- Representação ou história: Elemento principal, e imersão do jogador no mundo do jogo, com criação de problemas e ambiente próprio.

Jogos têm sido empregados para o ensino em diversas áreas, permitindo a visualização e o teste de conceitos e métodos presentes nas aulas teóricas. Desse modo, o uso de jogos é uma forma de que os alunos coloquem em prática tudo o que aprenderam em sala de aula, podendo melhorar seu desempenho, possibilitando a vivência de experiências em um ambiente preparado e seguro para seu desenvolvimento.

### 3 Trabalhos Relacionados

Vários projetos têm sido conduzidos com a finalidade de contribuir para a adaptação do Scrum em empresas de desenvolvimento e também na criação de jogos envolvendo a engenharia de software como foco.

[Oliveira, Guimarães e Fonseca \(2007\)](#) apresentam uma forma em que foi empregada a adaptação do Scrum, juntamente com métodos de qualidade na empresa Powerlogic Consultoria e Sistemas para melhoramento de processos e, consequentemente alcançar um nível superior de maturidade do MPS.BR. Em busca de alinhar a agilidade do gerenciamento ágil Scrum com as práticas de qualidade do nível F do MPS.BR, demonstrando que a adaptação dos métodos foi à solução para o formalismo excessivo do MPS.BR em comparação ao desenvolvimento Scrum, sugerindo melhorias e comprometimento de toda a equipe.

[Martins \(2016\)](#) com base em práticas de Gerenciamento de Riscos abordadas pelo Scrum e a criação um modelo que visa incluir algumas práticas de gerenciamento de riscos consideradas essenciais para o desenvolvimento de projetos, é abordado a adaptação do Scrum, como forma de solucionar problemas e dificuldades enfrentadas através da aplicação do Scrum. Apresentando uma adaptação do Scrum como tentativo de ajudar no desenvolvimento, melhorar a visualização, identificação e a compreensão das práticas de gerenciamento de riscos que ficam implícitas no método Scrum, fornecendo ainda um modelo para uma futura análise e criação de um jogo a ser utilizado por estudantes em busca de experiência teórica dos métodos aprendidos em sala de aula..

[Prikladnicki, Rosa e Kieling \(2007\)](#) propõe o uso de jogos para apoiar o ensino e a aprendizagem de gerência e desenvolvimento de projetos de *software* a aqueles que não possuem experiência ou querem praticar através de uma plataforma especialmente desenvolvida para este fim. Ainda foi demonstrada a experiência que duas universidades tiveram com o uso de jogos no ensino, em comparação ao ensino das aulas teóricas das universidades.

[Paludo, Raabe e Benitti \(2013\)](#) levanta o conhecimento que pode ser adquirido através da aplicação de métodos ágeis, gerando novas habilidades, tomada de decisões, entendimento e o impacto ao desenvolvimento de projetos. Aborda o preceito do uso de jogos no ensino e na aprendizagem, estando fortemente associadas às tentativas de aumentar o conhecimento aprendido em sala de aula. Apresentam o desenvolvimento de um jogo educativo voltado para área de gerenciamento de riscos em projetos de *software*, visando apoiar o ensino através de simulações de atividades gerenciamento de projetos, em busca de maior aproveitamento do conhecimento adquirido em aulas teóricas podendo

ser aplicadas na prática em simulações envolvendo riscos de projeto.

De acordo com algumas características previamente definidas como principais requisitos e que pudessem auxiliar na criação do jogo proposto e estando condizente com os modelos e normas de Gerenciamento de Risco apresentados pelo PMBOK ou MPS.BR. Diante dos resultados encontrados nos trabalhos estudados com pontos relevantes da pesquisa e estudo, fortalecendo a proposta e desenvolvimento de um jogo para apoiar o ensino de gerenciamento de riscos em projetos, como forma de auxiliar o estudante a aplicar seus conhecimentos fora da sala de aula.

## 4 Medodologia

O desenvolvimento do jogo foi dividido em etapas, que são: Pesquisa, Concepção e Produção. Também foi feita a formulação de questões a serem abordadas e de como o jogo poderia apresentar de forma intuitiva ao jogador, bem como um estudo de qual ferramenta de implementação e qual linguagem de programação seria utilizada, da plataforma e dos recursos para o desenvolvimento (*frameworks*, modelos, interface, tipo de jogo, dentre outras). Tendo em vista que se trata de um projeto com tempo limitado e apenas um integrante no processo de desenvolvimento.

As próximas seções descrevem as etapas do desenvolvimento, a ferramenta e a linguagem utilizada.

### 4.1 Etapas do Desenvolvimento

- Na primeira etapa foi realizada uma pesquisa exploratória sobre gerenciamento de projeto, desenvolvimento ágil e abordagens de melhoria de processos, uma vez que estes conceitos são a base para o desenvolvimento deste trabalho. Esta pesquisa exploratória foi realizada através da leitura de artigos científicos, monografias de conclusão de curso e livros.

A partir dessa pesquisa foram incluídas no modelo Scrum adaptado por [Martins \(2016\)](#), algumas questões sobre as práticas do gerenciamento de riscos consideradas importantes por modelos e normas de qualidade de processo de *software*.

- Na segunda etapa a de Concepção onde foram produzido o GDD e levantamento de requisitos do jogo, que contém todas as informações do jogo desenvolvido como nome, modos, gênero de jogo, história, prática de jogo e áudio ([PERUCIA et al., 2005](#)). Nessa etapa foi onde também, foi criada previamente toda a parte dos cenários e parte artística utilizados no jogo antes da implementação.

- A terceira etapa é constituída pela Produção. Essa etapa consiste na produção do jogo em si, em que foram criados o código-fonte do jogo, animações dos elementos na tela, mapas, terrenos, *Sprites*, e colocados efeitos de som, além disso foram feitos os testes de inserção e edição de perguntas, respostas e jogabilidade. Para a criação e codificação, foi utilizado a ferramenta gratuita de desenvolvimento Unity, há uma loja, *Asset Store*, que disponibiliza boas ferramentas para a criação de jogos em duas ou três dimensões. Nesta loja existem modelos de personagens, texturas e materiais, ferramentas de pintura de paisagem, ferramentas de efeitos sonoros, ferramentas de desenvolvimento em duas dimensões, entre muitos outros recursos. Possui também bom suporte para documentação

e possui diversos tutoriais que auxiliam no desenvolvimento, uma interface intuitiva e de fácil visualização e um fórum de desenvolvedores ([SCHNEIDER, 2016](#)).

## 4.2 Ferramenta e linguagem

Nesta seção é apresentada uma breve descrição das principais ferramentas, linguagem e recursos do Unity.

### 4.2.1 Engine Unity

Unity é um motor de jogos (*engine*) multi-plataforma, criado pela Unity Technologies para auxiliar na criação de jogos e aplicativos para diferentes plataformas. Aceita varias linguagens de programação que são elas: javascript, C *Sharp* e boo script. O Unity é destacado por vários tipos de desenvolvedores, desde grandes e pequenas empresas a profissionais e estudantes.

### 4.2.2 Recursos e Componentes

A Unity é uma ótima *engine* para criação de jogos em duas ou três dimensões, possuindo biblioteca de física, detector de colisão e ferramentas para animação com o auxílio de plugins e *frameworks* específicos.

Os desenvolvedores também podem criar componentes e recursos, que possam auxiliam em seus projetos. Esses recurso são disponibilizados na *Asset Store*. Dessa forma, com os recursos específicos incorporados, cada desenvolvedor pode fazer seu projeto de forma única e se adaptar o Unity para atender as suas necessidades de desenvolvimento. O Unity suporta ainda arquivos e formatos de objetos 3D de vários programas, podem ser adicionados ao projeto, e manejados pela interface gráfica do Unity. Porém estes recursos em três dimensões não foram utilizados neste projeto.

### 4.2.3 Linguagens de Programação

A programação no Unity é realizado por meios de scripts que podem ser codificado em três diferentes linguagens que são: JavaScript, C *Sharp* e Boo. Essas três linguagens de script apresentam diferentes formas de acessar a mesma funcionalidade básica do jogo ([PASSOS et al., 2009](#)).

Para este projeto foi utilizada a linguagem c *Sharp* por existir diversos exemplos e documentações na comunidade de desenvolvedores Unity, e também é mais simples e de baixo nível sendo uma linguagem de programação orientada a objetos.

Possui muitas similaridades com a linguagem C++, na qual ela foi baseada juntamente com linguagens Java e C, adaptando os melhores recursos de cada e acrescentando

novas capacidades. ela fornece suporte para várias funcionalidades que auxiliam no desenvolvimento de aplicações como esta desenvolvida neste trabalho, tais como: Suporte a áudio, imagens, animações e o processamento de banco de dados utilizando MySQL.

# 5 Resultados

Este capítulo apresenta o processo desenvolvimento e implementação do jogo criado neste trabalho. Primeiramente, são discutidas algumas etapas que antecederam o desenvolvimento do jogo como: Elaboração dos requisitos, análise dos requisitos funcionais e não funcionais, viabilidade de desenvolvimento do projeto, definição de público alvo, definição de metas, dentre outros. Por fim, é demonstrado todo o processo de desenvolvimento, como: telas, assets, mapa, texturas, jogabilidade, modos do jogo, cenários, pontuação e roteiro de eventos.

## 5.1 Projeto

### 5.1.1 Requisitos

O jogo aborda o seguinte conteúdo: conceitos básicos do Scrum, artefatos do Scrum, papéis no Scrum, gerenciamento de risco do Scrum adaptado com base no MPS-BR e no guia PMBOK.

O diagrama de casos de uso foi utilizado para descrever os atores e suas possíveis interações com o sistema. Na Figura 7 é apresentada a interação inicial do usuário com o “Menu Principal” do jogo, onde permite que o usuário escolha entre iniciar um novo jogo, ou o professor efetuar login para inserção das perguntas, como mostra a Figura 8, e também que o usuário saia do jogo.

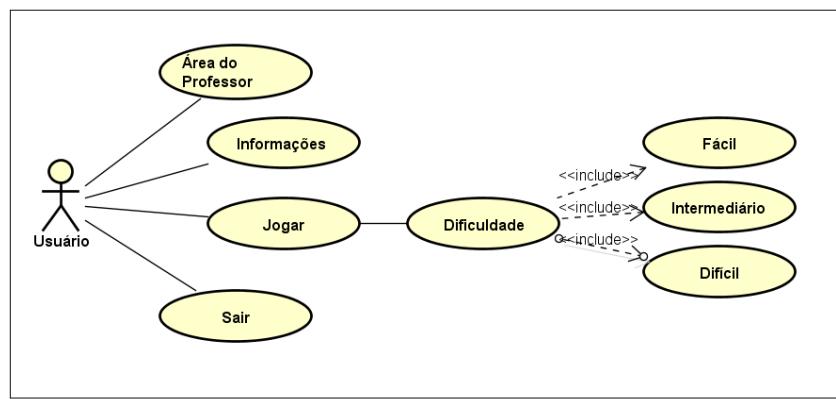


Figura 7 – Casos de uso à tela inicial do jogo.

Fonte: Próprio Autor

A Figura 9 apresenta a interação que o usuário tem dentro do cenário do jogo, onde ele deve percorrer o cenário até encontrar o NPC para realizar a missão.

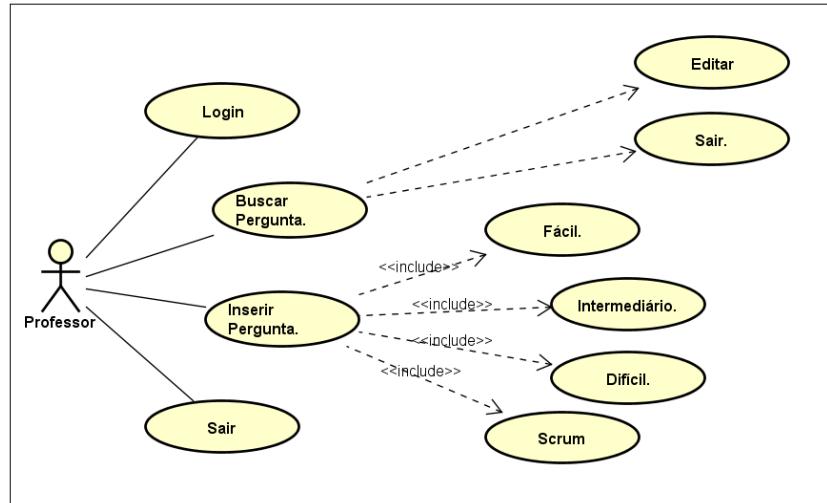


Figura 8 – Casos de uso à tela professor.

Fonte: Próprio Autor

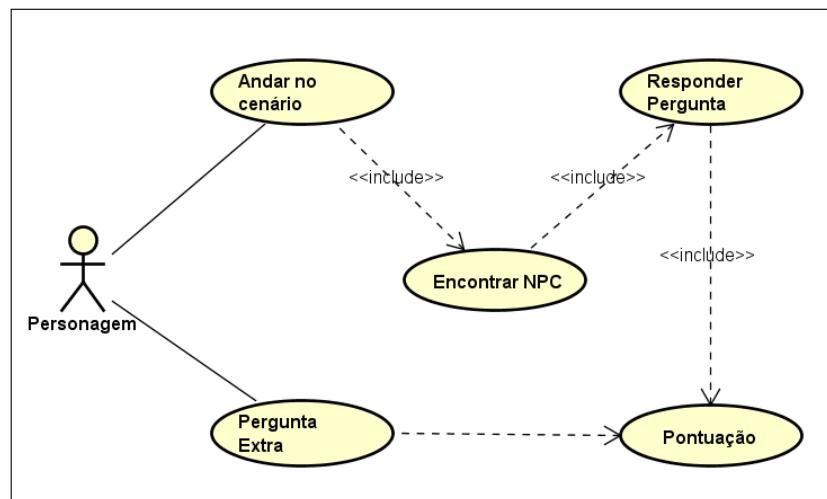


Figura 9 – Casos de uso referente ao cenário inicial do jogo.

Fonte: Próprio Autor

O jogo se divide em 8 etapas. Podendo ser utilizado em um encontro em sala de aula ou como tarefa de trabalho em casa, com a obtenção um de uma pontuação mínima. O jogo digital de ensino é em português, com cenários em duas dimensões, o jogo é disponibilizado gratuitamente, possibilita a execução *offline* por único jogador, compatível com a plataforma Windows. O jogo deve possuir boa temática para prender a atenção dos jogadores.

### 5.1.2 Game Design Document (GDD)

O *Game Design Document* é um documento gerado durante as etapas de desenvolvimento do *game* sendo o bastante para que o projeto seja executado de forma satisfatória (PERUCIA et al., 2005) (SCHUYTEMA, 2008). A maioria das metodologias para desenvolvimento de jogos surgiu de adaptações de metodologias genéricas de desenvolvimento de *software*, como *Extreme Programming* e o *Rational Unified Process* (RUP) (BARROS, 2007).

A primeira etapa foi a de Concepção. Nessa etapa, foi produzido o GDD, que contém todas as informações do jogo como título, gênero de jogo, enredo, mecânica de jogo, telas e áudio. O GDD foi criado tendo como base o modelo disponível em Schuytema (2008). Este documento (Tabela 1) visa demonstrar aspectos técnicos, discricionário do jogo, enredo, mecânica de jogo, objetivo, jogabilidade e ferramentas de desenvolvimento (Tabela 2). Destacando estes pontos é possível dar sequência ao desenvolvimento do jogo (SCHUYTEMA, 2008).

Tabela 1 – GDD

Atividade	Descrição
Descrição Simples	O jogo consiste em responder algumas questões sobre o Scrum, e principalmente sobre o gerenciamento de risco durante o processo de produção e desenvolvimento.
Descrição Longa	O jogo é sobre um membro da equipe de desenvolvimento de uma empresa, que busca participar ativamente dos projetos da empresa. O jogador se passa por este membro da equipe e cabe a ele passar por varias áreas de desenvolvimento buscando soluções para responder aos questionamentos feitos espalhados pela empresa. Existe também áreas em que é possível tirar algumas duvidas, receber alguma ajuda ou saber alguma curiosidade sobre o Scrum e o gerenciamento de projetos. O jogador deve sair pela empresa em busca de onde encontre novos símbolos de questões para poder responder e assim poder passar para o novo dia de planejamento que são divididos em 8 dias de desenvolvimento do projeto, A cada questão é subdividida em níveis de dificuldade para os jogadores iniciantes ou avançados. As questões e caixas de diálogos seja a curiosidades ou textos aparecem de forma aleatória para que não seja um ciclo vicioso entre as respostas dos jogadores. A câmera permanece sem movimento durante toda a partida, fazendo com que o jogador caminhe sempre em todo mapa, de forma que todo o cenário fique a mostra.

Fonte: Próprio Autor

Tabela 2 – GDD

<b>Atividade</b>	<b>Descrição</b>
Tipo/Gênero de Jogo	Quiz, RPG
Cenário	O jogo se passa em uma empresa de desenvolvimento e criação de <i>software</i> , onde o jogador se passar por um membro da equipe da que esta trabalhando em um projeto, que precisa responder alguns questionamentos para prosseguir durante os dias do desenvolvimento do projeto.
Sistema de jogo	O sistema de jogo comporta os seguintes elementos: Jogador: caminha por todo cenário em busca de questões ou informações sobre o plano de desenvolvimento daquele dia. Símbolos de questões (colisão): já existe na tela um ponto de referência para cada questão disponíveis no cenário do projeto; Símbolo de curiosidade (colisão): ativa o campo de informações e curiosidades para o jogador sobre o Scrum; Escada: após responder todas as questões disponíveis no primeiro andar ou cenário abre a possibilidade de subir para novos andares; Efeito de som: Música de fundo para cada cenário do jogo.
Jogabilidade	A jogabilidade consiste apenas no direcionamento do personagem no cenário do jogo pelas setas do teclado, em busca de questões e informações espalhados pelo cenário. A mecânica é simples, o jogador pode ir para qualquer parte da tela. O jogador não poderá aplicar mais nenhum tipo de efeitos como zoom ou arrastar a tela para visualizar outras partes do cenário, pois a câmera é fixa em todo o cenário do jogador.
Telas	O jogo deve conter as seguintes telas: Menu principal: a opção de iniciar o jogo, a opção de acessar o menu de administrador ou professor, informações sobre o jogo, sair e uma música de fundo; Menu de opções/professor: tela que contém a opção de editar, incluir, ou editar alguma questão para inclusão no banco de dados principal; Tela inicial do jogo <i>Play</i> : tela onde o jogo inicia; Tela de questão dentro do jogo: tela dentro do jogo, onde depois de encontrado o símbolo de pergunta o jogador possa ler e responder a questões propostas; Tela de pontuação: tela em que mostra o resultado, emitindo uma mensagem de parabéns caso a pontuação obtida seja maior que o mínimo previsto para passar para a nova etapa. Tela de <i>game over</i> : tela mostrada quando o jogador não consegue atingir a pontuação mínima. Deve conter uma mensagem para encorajamento e opção para jogar novamente.
Requisitos de Áudio	Música de fundo: Tema musical que permaneça ativo durante toda a partida;

Fonte: Próprio Autor

## 5.2 Implementação

### 5.2.1 Descrição do Jogo

O jogo desenvolvido segue o estilo um jogo de estratégia Quiz e RPG, onde o objetivo é tentar responder todas as perguntas do cenário de forma a conseguir uma pontuação mínima para passar para o próxima fase de desenvolvimento. Para isso, o jogador deve procurar atravessando todo o mapa a procura no cenário dos símbolos de perguntas que são elementos responsáveis por abrir o campo de perguntas e respostas para que o jogador possa responder.

Sempre que o jogador responder uma resposta certa o jogador recebe pontos, a partir da pontuação será verificado se ele conseguiu o mínimo exigido em cada fase. Cabe ao jogador criar uma estratégia para de estudos e procura de fontes para a melhoria de seus estudos, para passar por vários níveis do jogo e manter sua pontuação o melhor possível.

Para poder jogar, o usuário deve ter apenas o jogo em sua maquina para começar a jogar. Cada jogador tem atributos e particularidades desde o inicio do jogo. Através deles, é possível escolher o nível do jogo para iniciar, e ao passar por todas as fases de um nível do jogo, pode comparar seu desempenho com outros níveis disponíveis. Os principais atributos são:

- Pontos: Toda vez que o jogador responder uma pergunta correta ele será contemplado com pontos, que serão acumulados. Assim, será gerada uma nota final de todos seus pontos obtidos em cada fase do desenvolvimento, onde será designado se o jogador passou ou não de fase.
- Pontuação insuficiente: É a pontuação mínima exigida em cada fase, pois se o jogador não obtiver o mínimo suficiente ele não poderá prosseguir, tendo assim a possibilidade de repetir a fase ou sair do jogo.
- Nível: Atributo escolhido ao inicio do jogo. O nível das perguntas em dificuldade aumenta de acordo com um dado nível. O jogador responder todas as perguntas de um nível com uma pontuação satisfatória, ele poderá avançar para um nível superior para o aprendizado mais aprofundado.
- Fases: No jogo cada fase é descrita como uma dia no processo de desenvolvimento. A cada fase deve-se responder as perguntas sobre o gerenciamento de projetos Scrum, e o gerenciamento de risco do Scrum adaptado ao PMBOK e MPS-BR.
- Interação: Para jogar serão utilizados o mouse e o teclado como forma de controles. Através do teclado, o jogador pode percorrer todo o cenário, escolher onde e qual objeto buscar entre todos disponíveis apresentadas na tela. Os principais comandos do teclado são para movimentação de acordo com as funções padrão do Unity, e também

para inserção de novas perguntas com *login* de professor. Com o mouse é possível escolher a alternativa para a resposta do quiz, escolher itens da interface como resumo, retornar ao menu principal e prosseguir para a próxima fase.

### 5.2.2 Processo de Desenvolvimento

Por se tratar de um jogo casual, simples e de fácil entendimento para o público alvo, a estrutura do jogo também é simplificada. Após um estudo sobre as funcionalidades do jogo descritas na seção 5.1, houve um planejamento sobre quais as principais iterações, e que incrementos poderia ter.

O jogo é implementado utilizando uma câmera que não se move, os objetos do cenário são posicionados de forma a gerar um ambiente ilustrado. Em um jogo de duas dimensões toda arte é construída utilizando sprites e *sprite sheets*, que são várias imagens que, se mostradas em sequência, formam a animação desejada. Cenários e animações construídas com sprites requerem um grande trabalho artístico.

Neste jogo definiu-se uma jogabilidade simples e clara. Por isso buscou-se criar um cenário onde todas as plataformas estão alinhadas com o plano onde o jogador se move. Para tornar mais intuitivo o jogador deve interagir com as plataformas e se movimentar entre elas diretamente. Essas decisões contribuem para a uniformização da jogabilidade. O cenário foi construído utilizando a própria ferramenta de criação da Unity (Figura 10). Modelos e animações, podem ser encontrados à venda na internet ou na própria loja da Unity.

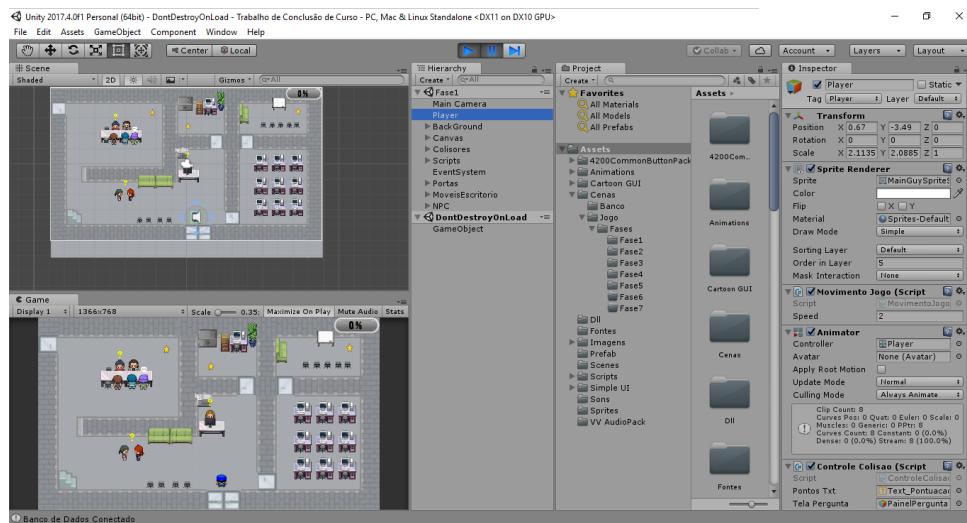


Figura 10 – Desenvolvimento do jogo com Unity

Fonte: Próprio Autor

Cenários

O Unity disponibiliza objetos para criação de cenários e para modelar os cenário. Um dos desafios deste projeto foi a criação dos cenários, pois o desenvolvimento em duas dimensões requer cenários e animações construídas com sprites com grande trabalho artístico conforme mostrado nas Figura 11 e 12.



Figura 11 – Cenário inicial do jogo.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor



Figura 12 – Cenário do jogo.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor

## Modelagem dos Objetos

Vários objetos foram criados diretamente no Unity para interagirem ou não com o personagem do jogo, entre eles destacam-se os símbolos de interrogação, estrelas para chamar a atenção dos jogadores.

## Texturas

Texturas de boa qualidade nos modelos desenvolvidos proporciona um maior realismo para chamar atenção e dar uma maior aproximação com a realidade com o jogador. Após a escolha e posicionamento de todos os objetos que estão no cenário foi feito o mapeamento manual dos objetos, por ser mais preciso nas superfícies, e de fácil implementação como nos pisos e paredes longas usando repetições de texturas. Após a criação e formação das texturas que seriam utilizadas no projeto, foram criadas de tamanho padrão para que não seja preciso o Unity converter as imagens de tamanhos diferentes (Figura 13).



Figura 13 – Cenário do jogo.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor

## Terreno

Foram modelados apenas um tipo de terreno para o projeto, mas como uma parte do cenário, existe várias paredes e delimitando salas e áreas de convivência no meio do terreno, podendo ainda sofrer variações de acordo com o ambiente em que jogo se encontra pois cada fase é dividido em até três andares. Essas variações não são no terreno do jogo em si, pois este é sempre constante, mas sim, nos elementos do cenário que podem facilitar ou dificultar determinada ação ou caminho até os objetivos de encontrar as questões.

## Assets

Para este projeto foram importados vários *assets* gratuitos, da propria loja da Unity e também alguns que incluidos por padrões na instalação do Unity. Na criação de um novo projeto tem como selecionar quais destes pacotes serão importados, mas posteriormente podem ser adicionados manualmente no projeto. Foram utilizados os *assets* padrões de terreno, controle de personagem, partículas, entre outros.

## Roteiro de eventos

Foram definidos todos os passos que o jogador precisa efetuar para conseguir avançar na partida. Em vários momentos o jogador precisará encontrar determinados itens ou ações para abrir a janela de questões, de informações ou subir para o andar superior. O jogo com este roteiro tem a vantagem de ter poucas falhas, mas caso queira iniciar uma nova partida, praticamente o roteiro será o mesmo que o anterior, mais as várias questões abordadas no cenários mudam aleatoriamente.

## Níveis de dificuldade do jogo

Uma análise rápida foi feita em relação à dificuldade. Qual é o nível de dificuldade do jogo, se tornou um meio de pesquisa para separar o que o deve ser levado a fundo em cada um dos níveis escolhidos pelo jogador no inicio do jogo. A dificuldade varia em uma escala gradativa, podendo ser: fácil, intermediário ou difícil conforme mostrado na Figura 14. Vale ressaltar que a dificuldades são estáticas, com banco de dados separados em tabelas diferentes umas das outras. Além disso, todo inicio do jogo é preciso escolher a dificuldade para entrar no jogo.



Figura 14 – Cenário do jogo.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor

## Personagem

Com a criação de apenas um personagem simulando o próprio jogador. Sendo controlado pelo jogador, ele é completamente visível para que possa acompanhar todas suas ações e movimentos. As movimentações do personagem estão inclusas nas *assets* padrões do Unity, onde foram feitas pequenas alterações em suas propriedades, como aceleração dos passos, virar o corpo do personagem enquanto caminha e todos sendo controlados pelo teclado.

## Músicas e Efeitos Sonoros

Para este projeto foi criada uma pasta Sons dentro do diretório principal do projeto, onde estão todos os sons usados no jogo. O Unity suporta varios formatos de arquivos de som e foram usados no formato mp3. Todos os sons que foram utilizados, foram adquiridos no site pela [www.freesound.org](http://www.freesound.org), especialmente é para arquivos de áudio que não possuem copyright.

## Transições de telas e opções

Tela inicial, onde todos tem acesso, primeiramente com a opção de *Play* para iniciar um novo jogo, informações para ler as informações disponíveis a respeito do jogo (Figura 15).



Figura 15 – Menu principal do jogo.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor

A Figura 16 mostra a tela da área do professor, onde tem a possibilidade de editar, inserir ou apagar algumas das questões abordadas no jogo. Na Figura 17 apresenta os campos para inserir uma nova pergunta do quiz ao banco de dados do jogo. Na Figura 18

mostra onde podemos alterar perguntas ja armazenadas no banco de dados.



Figura 16 – Menu de acesso restrito ao professor.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor



Figura 17 – Menu de inserção de pergunta ao banco de dados.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor



Figura 18 – Menu edição de perguntas do banco de dados.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor

Dentro do jogo é possível abrir um menu de botões apertando a tecla Esc. A tela de menu interna do jogo é carregada automaticamente na execução do jogo e pode ser chamada a qualquer momento no jogo, conforme mostrado na Figura 19. Existe o botão de resume, menu principal do jogo e o botão sair que é o responsável por sair jogo. Pode ser chamada a qualquer momento do jogo, por isso não é uma transição de tela, possui apenas as opções de sair e reiniciar a partida.



Figura 19 – Menu interno durante a execução do jogo.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor

A tela do quiz é aberta assim que o personagem colidir com alguns símbolos de interrogação espalhados pelo cenário. Ao colidir com o objeto será aberto uma tela onde será apresentado, diretamente uma pergunta com quatro alternativas sendo que apenas uma é a correta. A tela não tem tempo, para que o jogador possa ter tempo de ler e pesquisar na resposta correta, conforme mostra a Figura 20.

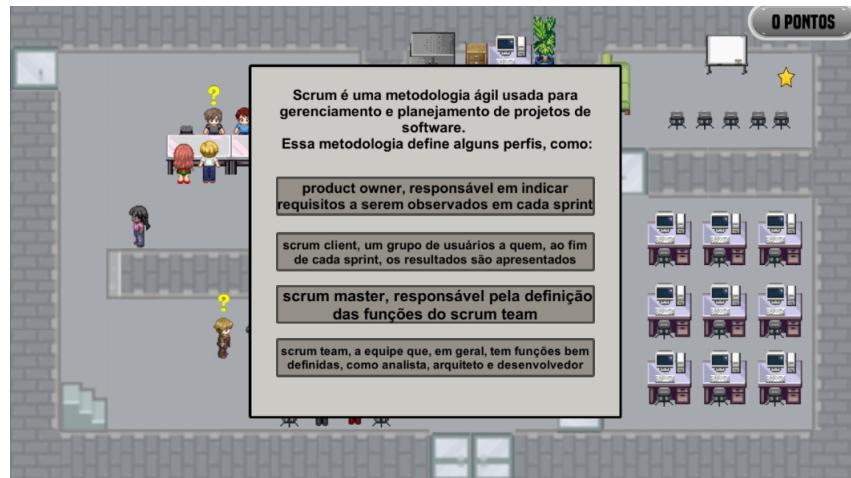


Figura 20 – Tela do quiz aberta durante o jogo.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor

A tela de acerto ou erro do quiz é executado quando o jogador após responder a pergunta selecionando a alternativa correta, como mostrado a Figura 21, ele recebe a pontuação daquela questão, ao selecionar a alternativa errada. Na Figura 22 é mostrado a resposta certa, para que o jogador saiba a resposta e não recebe nenhuma pontuação.



Figura 21 – Mensagem de acerto da questão.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor



Figura 22 – Mensagem informa que o jogador errou a questão e mostra a resposta certa.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor

### Pontuação

A pontuação é determinada pelos pontos obtidos pelo acerto das respostas em cada uma das questões respondidas durante o jogo conforme mostra a Figura 23. No caso, que o jogador acerte todas as questões e obtiver 100 por cento de aproveitamento. Para determinar a pontuação final da fase, o cálculo é feito de maneira simples, em que cada pergunta respondida corretamente o jogador tem um acréscimo de pontos e obtendo um número mínimo de sessenta por cento de aproveitamento ele passa para a próxima fase.



Figura 23 – Pontuação do jogo.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor

A tela de conclusão é executado quando o jogador conclui com sucesso todas as tarefas dentro do jogo, e consegue o mínimo de pontuação exigida, abre três opções: prosseguir para nova fase, menu principal ou repetir a mesma fase.

O menu de *game over* é executado quando o jogador não conclui todas as tarefas e não consegue o mínimo de pontuação exigida, sendo assim ele não consegue prosseguir para a próxima fase, podendo apenas ir ao menu principal ou repetir a mesma fase (Figura 24).



Figura 24 – Tela de game over da fase.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor

### Colisões

Foi usado o sistema padrão do próprio Unity para as colisões do personagem. Define um raio de colisão do corpo do personagem para o local onde está programado uma colisão. Usando alguns gatilhos do Unity, estes são um dos itens mais importantes no desenvolvimento. Definindo quando determinada ação deve ser feita, uma porta a ser liberada como mostra a Figura 25, um personagem passar ou ter a liberdade para uma determinada área, um item ou objeto aparecer ou desaparecer.

Durante o jogo existe objetos que abrem uma tela com mensagens sobre o tema abordado como mensagens de ajuda (Figura 26). De acordo com cada movimento do personagem dentro do jogo, sempre são apresentadas algumas mensagens explicando ou alertando sobre eventos de como ajudar ou curiosidades, desta forma o mesmo não ficará perdido dentro do ambiente.



Figura 25 – Pontuação do jogo.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor

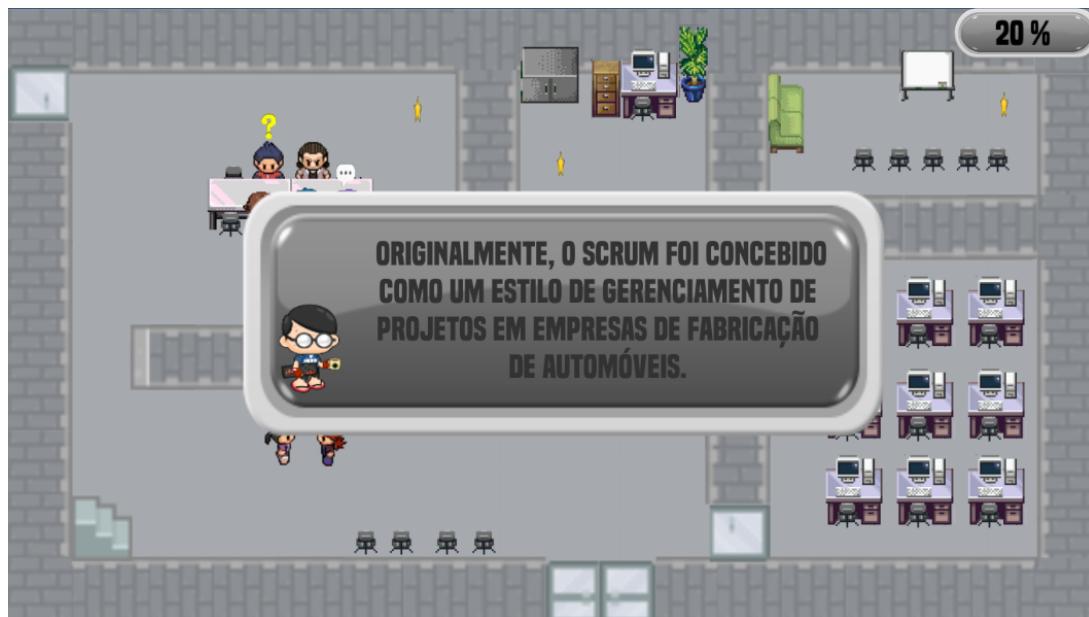


Figura 26 – Tela com mensagem de ajuda.

Fonte: Imagem do jogo desenvolvido pelo próprio Autor

## 6 Conclusão

Neste trabalho foi apresentado o desenvolvimento de um jogo para ensino do processo de desenvolvimento ágil Scrum. Sabendo-se que os jogos têm sido considerados como uma excelente ferramenta no processo de ensino aprendizagem, neste contexto foi abordando principalmente o gerenciamento de risco, adaptando melhorias de processos segundo o guia PMBOK e MPS-BR. Com base na pesquisa e análise da fundamentação teórica realizada onde foi possível identificar meios de melhoria no processo de desenvolvimento de projetos ([OLIVEIRA; GUIMARÃES; FONSECA, 2007](#)).

Cada elemento do jogo foi cuidadosamente analisado, em busca de desenvolver um jogo de fácil entendimento e rápida adaptação. Após a revisão teórica, o desenvolvimento dos níveis de dificuldade do jogo, demandou mais planejamento e pesquisa, em busca de questões que abordassem de forma gradual o passar do desenvolvimento. Os elementos projetados e o terreno criado se adequaram bem ao intuito do jogo que é: simular o ambiente de trabalho de uma empresa de desenvolvimento, resultando em uma experiência consistente e agradável. Levando em consideração o perfil dos usuários que vão fazer uso do jogo, em que exige um conhecimento sólido em diversas áreas da engenharia de *software* e do aprendizado em sala de aula.

Com o desenvolvimento desde trabalho onde foi apresentado um meio para que estudantes de nível técnico e superior possam utilizar esta ferramenta como uma nova opção de ensino e aprendizagem e que possa impactar de maneira positiva nos estudos e aplicação de gerenciamento de projetos e gerenciamento de risco em qualquer projetos futuros a serem desenvolvimentos pelos estudantes.

Como todo *software*, o jogo precisa estar em constante ajuste e adaptações para novas tendências. Para trabalhos futuros a proposta seria uma forma de maximizar o atendimento do jogo, não apenas ao princípio do Scrum e o gerenciamento de risco, mas para todo o processo de desenvolvimento, passando por todas etapas do processo. Para isso, deve ser criado mais scripts, cenas adequadas para cada momento específico do jogo. Também poderia melhorar a experiência com o usuário, com a criação de inimigos para que possam perseguir o personagem criando assim um ritmo mais dinâmico ao jogo.

## Referências

- AMARAL, D. C. et al. Gerenciamento ágil de projetos: aplicação em produtos inovadores. **São Paulo:** Saraiva, 2011.
- BARROS, R. d. Análise de metodologias de desenvolvimento de software aplicadas ao desenvolvimento de jogos eletrônicos. **Trabalho de Graduação–Departamento de Informática–Universidade Federal de Pernambuco, Recife,** 2007.
- BOEHM, B. W. Software risk management: principles and practices. **IEEE software,** IEEE, v. 8, n. 1, p. 32–41, 1991.
- BOYADJIAN, J. C. **Gestão de Projetos Brasil - Conceitos e técnicas.** [S.l.]: IETEC - Instituto de Educação Tecnológica, 2012.
- CAMPOS, C. S. d. **GeRDDoS: uma proposta de um modelo para identificação e análise de riscos em ambiente de desenvolvimento distribuído de software.** Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul, 2009.
- CARVALHO, C. E. C. d.; ABRANTES, C. T. d.; CAMEIRA, R. F. Métodos ágeis de desenvolvimento de software: um caso prático de aplicação do scrum. 2011.
- FERREIRA, D. et al. Scrum-um modelo ágil para gestão de projetos de software. **Acesso em,** v. 28, 2017.
- FOWLER, M. The new methodology. **Wuhan University Journal of Natural Sciences,** Springer, v. 6, n. 1-2, p. 12–24, 2001.
- GERAL, M. G. Mps. br-melhoria de processo do software brasileiro. 2012.
- GUIMARÃES, C. A.; ROSA, D. D. S. Gerenciamento ágil de projetos de desenvolvimento de softwares na secretaria de estado da casa civil. 2016.
- GUSMÃO, C. Martins Gomes de. Um modelo de processo de gestão de riscos para ambientes de múltiplos projetos de desenvolvimento de software. Universidade Federal de Pernambuco, 2009.
- IMPLEMENTAÇÃO-PARTE, G. de; NÍVEL, C. Mps. br-melhoria de processo do software brasileiro. 2007.
- ISO, N. 10006, gestão da qualidade—diretrizes para a qualidade no gerenciamento de projetos. **Associação Brasileira de Normas Técnicas ABNT. Rio de Janeiro,** 2000.
- MARÇAL, A. S. C. et al. Estendendo o scrum segundo as áreas de processo de gerenciamento de projetos do cmmi. **CLEI Electronic Journal,** 2007.
- MARTINS, J. C. C. **Técnicas para gerenciamento de projetos de software.** [S.l.]: Brasport, 2007.
- MARTINS, L. M. d. A. Adaptação do scrum e modelagem de um jogo segundo a área de processo de gerenciamento de riscos. 2016.

- NAVARRO, E. O.; HOEK, A. van der. Simse: An interactive simulation game for software engineering education. In: **CATE**. [S.l.: s.n.], 2004. p. 12–17.
- OLIVEIRA, A. C. G. de; GUIMARÃES, F. A.; FONSECA, I. de A. Utilizando metodologias ágeis para atingir mps. br nível f na powerlogic. 2007.
- OLIVEIRA, C. D. C. d. Ensino e gestão de riscos em projetos de software usando jogos sérios, lógica fuzzy e agentes inteligentes. 2014.
- PALUDO, L.; RAABE, A. L. A.; BENITTI, F. B. V. Rskmanager—um jogo para apoiar o ensino de gerência de riscos em projetos de software. **RENOTE**, v. 11, n. 3, 2013.
- PASSOS, E. B. et al. Tutorial: Desenvolvimento de jogos com unity 3d. In: **VIII Brazilian Symposium on Games and Digital Entertainment**. [S.l.: s.n.], 2009. p. 1–30.
- PERUCIA, A. S. et al. Desenvolvimento de jogos eletrônicos. **São Paulo**: Novatec, 2005.
- PMI, P. G. Um guia do conhecimento em gerenciamento de projetos. **Pennsylvania: Project Management Institute**, 2013.
- PRENSKY, M. Não me atrapalhe, mãe—eu estou aprendendo. **São Paulo**: Phorte, p. 320, 2010.
- PRESSMAN, R. S. **Engenharia de Software**. Tradução: Rosângela Deloso Penteado. [S.l.]: São Paulo: McGraw-Hill, 2006.
- PRIKLADNICKI, R.; ROSA, R.; KIELING, E. Ensino de gerência de projetos de software com o planager. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2007. v. 1, n. 1, p. 11–20.
- ROCHA, P. C.; BELCHIOR, A. D. Mapeamento do gerenciamento de riscos no pmbok, cmmi-sw e rup. 01 2004.
- SBROCCO, J.; MACEDO, P. Metodologias ágeis-engenharia de software sob medida. **São Paulo**: Érica, 2012.
- SCHNEIDER, M. **SCRUM'ed: Um jogo de RPG para ensinar Scrum**. 2015. Tese (Doutorado) — Tese de Graduação em Sistemas de Informação, Universidade Federal de Santa Catarina (UFSC). Disponível em:< [http://www.gqs.ufsc.br/wpcontent/uploads/2011/11/TCCfinal\\_SCRUMed.pdf](http://www.gqs.ufsc.br/wpcontent/uploads/2011/11/TCCfinal_SCRUMed.pdf), 2016.
- SCHUYTEMA, P. **Design de games: uma abordagem prática**. [S.l.]: Cengage Learning, 2008.
- SCHWABER, K. **Agile project management with Scrum**. [S.l.]: Microsoft press, 2004.
- SCHWABER, K.; SUTHERLAND, J. Um guia definitivo para o scrum: As regras do jogo. **Julho de**, 2013.
- SEI, S. Cmmi® for development (cmmidev), v1. 2, cmu/sei-2006-tr-008. **Software Engineering Institute**, 2006.

- SERVIÇOS, G. G. M. de. Mps-melhoria de processo de software e serviços. 2012.
- SOMMERVILLE, I.; ARAKAKI, R.; MELNIKOFF, S. S. S. **Engenharia de software**. [S.l.]: Pearson Prentice Hall, 2008.
- TAKEUCHI, H. &ikujiro nonaka. the new product development game. **Harvard Business Review**, 1986.
- TEIXEIRA, J. S. F.; JESUS, V. S. Eveline de; FERNANDES, C. T. Representação de jogos educacionais a partir do modelo de objetos de aprendizagem. In: **Anais do Workshop de Informática na Escola**. [S.l.: s.n.], 2007. v. 1, n. 1.