**Especificação do Projeto**

**BPM Game Engine**

**Versão 1.0**

Fabio Takeshi Ishikawa

Janeiro de 2020

Contents

[**1. Introdução** 3](#_Toc32509987)

[1.1 O que é BPM Game Engine 3](#_Toc32509988)

[1.2 Motivação do Projeto 3](#_Toc32509989)

[1.3 Objetivo deste documento 3](#_Toc32509990)

[1.4 Público-alvo deste documento 3](#_Toc32509991)

[**2. Descrição Geral de BPM Game Engine** 3](#_Toc32509992)

[2.1 Funcionalidades essenciais 3](#_Toc32509993)

[2.2 Os Usuários 3](#_Toc32509994)

[2.3 Plataformas e Ferramentas 3](#_Toc32509995)

[2.4 Documentações 3](#_Toc32509996)

[2.5 Dependências 3](#_Toc32509997)

[**3. Processos de Software** 3](#_Toc32509998)

[3.1 Processo: Incremental Adaptado 3](#_Toc32509999)

[3.2 Descrição das Etapas do Projeto 3](#_Toc32510000)

[3.3 Descrição da análise e planejamento de cada iteração 4](#_Toc32510001)

[3.4 Descrição da execução da iteração 4](#_Toc32510002)

[**4. Engenharia de Requisitos** 4](#_Toc32510003)

[4.1 Como descrever requisitos de usuário e requisitos de sistema 4](#_Toc32510004)

[4.2 Como definir as prioridades e riscos dos requisitos 4](#_Toc32510005)

[4.3 Processo de elicitação e análise de requisitos 4](#_Toc32510006)

[**5. Modelagem Conceitual (é necessário?)** 4](#_Toc32510007)

[5.1 Cenários ou Diagramas UML? 4](#_Toc32510008)

[**6. Projeto de Software** 4](#_Toc32510009)

[6.1 Visão Geral de todas as ferramentas 4](#_Toc32510010)

[6.2 Arquitetura Game Engine (Core) 4](#_Toc32510011)

[6.3 Parser BPMN (Camunda) 4](#_Toc32510012)

[6.4 Método PYP – Play Your Process 4](#_Toc32510013)

[**7. Teste de Software** 5](#_Toc32510014)

[7.1 Plano de Testes 5](#_Toc32510015)

[7.2 Testes de Unidade Automatizadas 5](#_Toc32510016)

[7.3 Testes de Integração Automatizada 5](#_Toc32510017)

[7.4 Testes de Aceitação 5](#_Toc32510018)

[7.5 Documentação de Testes 5](#_Toc32510019)

[**8. Gerência de Configuração de Software** 5](#_Toc32510020)

[8.1 Slack – Comunicação entre a equipe 5](#_Toc32510021)

[8.2 Trello – Gestão de tarefas (Kanban) 5](#_Toc32510022)

[8.3 Github – Controle de Versão 5](#_Toc32510023)

[8.3.1 Versionamento Semântico 5](#_Toc32510024)

[8.3.2 Workflow: Pull Request Workflow 5](#_Toc32510025)

[8.3.3 Workflow: No Switch Yard 5](#_Toc32510026)

[8.3.4 Guia de contribuição open source 5](#_Toc32510027)

[8.4 DevOps com Github, TravisCI e Codacy 5](#_Toc32510028)

[**9. Garantia de Qualidade** 5](#_Toc32510029)

# **1. Introdução**

## 1.1 O que é BPM Game Engine

## 1.2 Motivação do Projeto

## 1.3 Objetivo deste documento

## 1.4 Público-alvo deste documento

# **2. Descrição Geral de BPM Game Engine**

## 2.1 Funcionalidades essenciais

## 2.2 Os Usuários

## 2.3 Plataformas e Ferramentas

## 2.4 Documentações

## 2.5 Dependências

# **3. Processos de Software**

## 3.1 Processo: Incremental Adaptado

## 3.2 Descrição das Etapas do Projeto

* Análise do projeto (geral)
* Análise dos requisitos de software (geral)
* Plano de Desenvolvimento
* Plano de Teste
* Desenvolvimento
* Teste de Unidade
* Teste de Integração
* Teste de Aceitação
* Final da iteração – análise da iteração realizada e atualizações de todos os documentos

**Nota:** Os itens com fundo azul representam as etapas que utilização ferramentas automatizadas relacionada a DevOps (Github, Travis.CI, ReaderTheDocs, etc).

## 3.3 Descrição da análise e planejamento de cada iteração

## 3.4 Descrição da execução da iteração

# **4. Engenharia de Requisitos**

## 4.1 Como descrever requisitos de usuário e requisitos de sistema

## 4.2 Como definir as prioridades e riscos dos requisitos

## 4.3 Processo de elicitação e análise de requisitos

* Descoberta e compreensão dos requisitos
* Classificação e organização dos requisitos
* Priorização e negociação dos requisitos
* Documentação dos requisitos

# **5. Modelagem Conceitual (é necessário?)**

## 5.1 Cenários ou Diagramas UML?

# **6. Projeto de Software**

## 6.1 Visão Geral de todas as ferramentas

## 6.2 Arquitetura Game Engine (Core)

## 6.3 Parser BPMN (Camunda)

## 6.4 Método PYP – Play Your Process

# **7. Teste de Software**

## 7.1 Plano de Testes

## 7.2 Testes de Unidade Automatizadas

## 7.3 Testes de Integração Automatizada

## 7.4 Testes de Aceitação

## 7.5 Documentação de Testes

# **8. Gerência de Configuração de Software**

## 8.1 Slack – Comunicação entre a equipe

## 8.2 Trello – Gestão de tarefas (Kanban)

## 8.3 Github – Controle de Versão

## 8.3.1 Versionamento Semântico

## 8.3.2 Workflow: Pull Request Workflow

## 8.3.3 Workflow: No Switch Yard

## 8.3.4 Guia de contribuição open source

## 8.4 DevOps com Github, TravisCI e Codacy

# **9. Garantia de Qualidade**