

Relatório de Execução de Processo

Breno de Melo Gomes - 201503330
David Matheus Santos Sousa - 201503336
Erick Vinícius Vieira e Souza - 201515568
Iago Bruno Oliveira Miranda Almeida - 201703756
Paula Isabela Cardoso Resende - 201503358
Rhenan Konrad - 201503363

11 de novembro de 2017, Goiânia -GO

Sumário

- 1- Definição, documentação e elaboração dos diagramas (BPM) do processo
- 2 - Iniciação e Análise de Requisitos
- 3 - Projeto
- 4 - Implementação
- 5 - Teste e Homologação
- 6 - Implantação

1- Definição, documentação e elaboração dos diagramas (BPM) do processo

A documentação e os diagramas referentes ao processo podem ser encontrados no repositório oficial do projeto, localizado em: [processo-sw-2017-2](#).

2 - Iniciação e Análise de Requisitos

O processo de Análise de Requisitos se dá de maneira manual, sendo desempenhado pela figura do analista de requisitos. Durante a execução do processo tem-se como artefato o **Backlog do Produto**, que possui listados todos os requisitos, devidamente catalogados em formato de *stories* e em categorias que envolvem: ID, nome, importância, estimativa, demonstrar e notas.

Além da construção do backlog, são definidos os **Stakeholders**, ou seja, todos os envolvidos do processo com descrição de participação, delimitando características do Product Owner, cliente e equipe de desenvolvimento. Dando continuidade na execução do projeto, definem-se então o **Cronograma de Iterações** e suas datas, assim como breve descrição algumas das principais atividades da sprint em questão.

Nesta fase também é definido **Diagrama de Casos de Uso e Descrição de Casos de Uso**, seguido da descrição das *stories*. Esses artefatos são todos gerados a partir de trabalho em conjunto entre analista e cliente, juntamente com product owner, sendo que o principal objetivo é delimitar as funcionalidades que serão desenvolvidas nas próximas etapas a serem descritas.

O documento de saída da Análise de Requisitos é utilizado como entrada para o processo de Projeto (Design).

3 - Projeto

Nesta etapa foi definido que o modelo arquitetural do projeto é baseado no modelo cliente-servidor (onde, no servidor, a linguagem adotada é Java e *frameworks* em geral de baixo custo enquanto a camada do cliente dependerá de recursos do Google Chrome) e *Model-View-Controller* para diminuir o acoplamento ou seja será dividido em três camadas:

- **Persistência:** onde há abstrações das estruturas de dados tornando flexíveis mudanças futuras
- **Controle:** onde há chamadas de operações que atendem os requisitos funcionais.
- **Visualização:** onde as chamadas de execução são feitas e a interface com o usuário é construída.

A classe *Persistence Model* armazena dados do serviço e **usuário** é uma especialização da classe de persistência que armazena os dados de usuários que podem ou passageiro ou motorista (cada um com informações pertinentes ao seu papel). **Itinerário** é outra especialização que armazena dados das corridas feitas e **Veículo** armazena dados do veículo conduzido pelo motorista.

4 - Implementação

O processo de implementação se dá de maneira manual, sendo desempenhado pela figura do **desenvolvedor**. Durante a execução do processo, o **desenvolvedor** seleciona um dos requisitos novos disponíveis para construção, realiza a criação de uma nova **Branch**, onde serão depositados os códigos produzidos acerca dessa nova funcionalidade. Ao final da construção, o desenvolvedor notifica o término da atividade, através do Trello, e realiza o commit do código produzido.

5 - Teste e Homologação

O processo de teste e homologação se dá de maneira mista, contendo partes automatizadas e partes manuais. Durante a parte de construção dos testes, quem desempenha essa atividade será o **desenvolvedor**, construindo casos de testes para o requisito que passou pela fase de implementação e já possui código em sua **Branch**. Ao concluir a construção dos casos de testes, o desenvolvedor realiza o commit do código produzido e notifica a equipe através da ferramenta **Trello**. A **homologação** é realizada pelo **Scrum Master** da equipe, verificando os testes e código produzido, se há conformidade com os requisitos, tendo aceitação, o teste automatizado é executado e em caso positivo, a **Branch** é feito o merge da **Branch** referente a funcionalidade com a Branch de desenvolvimento. Após a conclusão, a equipe é notificada sobre o sucesso ou rejeição do código produzido, através do **Trello**.

6 - Implantação

O processo de implantação será realizado ao final de cada sprint, após a homologação da mesma, e deverá ser concluído em até um dia útil. O nome da versão será definido seguindo a fórmula **x.yy**, sendo x uma versão final e y um draft da versão final. Com o processo totalmente finalizado, o cliente é informado por e-mail que uma nova versão da aplicação está disponível para download. Tendo em vista a entrega incremental, deverá ser entregue primeiro a parte da aplicação referente ao uso do motorista e em seguida, a aplicação do usuário.