



PROCESSO DE SOFTWARE DA PRODABEL



Este trabalho está licenciado sob a Licença Atribuição-NãoComercial-Compartilhalgual 3.0 Brasil Creative Commons. Para visualizar uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nc-sa/3.0/br/> ou mande uma carta para Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Qualquer parte desta publicação pode ser reproduzida, desde que citada a fonte, de acordo com as orientações da licença Creative Commons (CC BY-NC-SA 4.0)

Este documento encontra-se disponível no Portal do PSP [PSPPBH].

ACRÔNIMOS

AD - *Active Directory*

CD - Entrega Contínua (*Continuous Delivery*)

CI - Integração Contínua (*Continuous Integration*)

DAU - Diretoria de Atenção ao Usuário

DevOps - Desenvolvimento e Operações

DSI - Diretoria de Sistemas de Informação

EIC - Esteira de Integração Contínua

EPP - Escritório de Projetos

EVT - Estudo de Viabilidade Técnica

ISO - *International Organization for Standardization* (Organização Internacional para padronização)

LDAP - *Lightweight Directory Access Protocol*

LGPD - Lei Geral de Proteção de Dados

HML - Homologação

MLFDC - Modelo Lógico Funcional do Datacenter

MVP - *Minimum Viable Product* (Produto Viável Mínimo)

PBH - Prefeitura de Belo Horizonte

PO - *Product Owner* (Dono do Produto)

Prodabel - Empresa de Informática e Informação da Prefeitura de Belo Horizonte

PSP - Processo de Software da PBH

QA - *Quality Assurance* (Garantia de Qualidade)

RDM - Requisição de Mudança

SAS - Superintendência de Arquitetura de Sistemas

SDM - *Service Desk Manager*

SIS - Sistemas da Prefeitura de Belo Horizonte

SMI - Superintendência de Monitoração e Segurança da Informação

SOI - Superintendência de Operações Datacenter

SSS - Superintendência de Sustentação de Sistemas

TDD - *Test Driven Development*

TI - Tecnologia da Informação

TIC - Tecnologia da Informação e Comunicação

XP - *eXtreme Programming* (Programação Extrema)

SUMÁRIO

ACRÔNIMOS.....	2
SUMÁRIO	3
1. Apresentação	1
1.1. O Processo de Software da PBH - PSP	1
1.2. O relacionamento do PSP com o <i>Scrum</i>	2
2. Escopo do documento	4
2.1. Público alvo e serviços abordados.....	4
2.1. Limitações.....	5
3. Grupos de atividades dos projetos	6
4. Modelo de Referência	9
4. Papéis envolvidos	11
5. Detalhamento das atividades	12
5.1. Atividades de planejamento	12
5.1.1. Análise Inicial da Demanda	13
5.1.2. Planejamento do Projeto	15
5.2. Atividades de construção do <i>Release</i> (ciclo de desenvolvimento)	18
5.2.1. Atividades de Planejamento da <i>Release</i>	19
5.2.1.1. Planejar <i>Release</i>	20
5.2.1.2. <i>Sprint</i> 0	22
5.2.2. Atividades de Construção das <i>Sprints</i>	25
5.2.2.1. Planejar a <i>Sprint</i>	26
5.2.2.2. Executar <i>Sprint</i>	29
5.2.2.3. Refinamento do <i>Backlog</i>	37
5.2.2.4. Reunião Diária	39
5.2.2.5. Revisão da <i>Sprint</i>	41
5.2.2.6. Reunião de Retrospectiva	43

5.3. Atividades de Transição do Projeto.....	44
5.3.1. Validar / Homologar a <i>Release</i>	45
5.3.2. Preparar e Realizar Treinamentos.....	46
5.3.3. Comunicar a entrada de um sistema em Produção ou em <i>Staging</i>	47
5.3.4. Comunicar a realização de uma mudança no sistema.....	48
5.3.5. Realizar o repasse para a Sustentação	49
5.3.5. Encerrar o Projeto	50
5.4. Atividades de Gestão de Ambientes de TI	52
5.4.1. Publicar <i>software</i> com a utilização da EIC	53
5.4.1. Publicar Bancos de Dados	55
5.5. Atividades de Acompanhamento de Projeto.....	56
5.5.1. Acompanhar Projeto.....	58
5.5.2. Operação Assistida	59
6. Referências Bibliográficas.....	61
7. Revisões do documento	64
ANEXO I - CONCEITOS E FUNDAMENTOS SOBRE O MODELO DE REFERÊNCIA	65
ANEXO II - PAPÉIS DA EQUIPE NO MODELO DE REFERÊNCIA	70

1. APRESENTAÇÃO

1.1. O Processo de Software da PBH - PSP

A Empresa de Informática e Informação do Município de Belo Horizonte (Prodabel) é uma sociedade de economia mista municipal responsável pela gestão da informática pública da capital mineira. Ela tem como missão promover, integrar e gerenciar soluções de TI e está presente nas mais diversas áreas de gestão da Prefeitura como a saúde, educação, turismo, finanças, transporte e inclusão digital. Além disso, ela também é responsável pela definição, manutenção e evolução dos métodos, processos e ferramentas de desenvolvimento de *software* que atendam às necessidades da PBH e de seus órgãos. Isso ocorre através do Processo de *Software* da PBH - PSP que é coordenado pela Superintendência de Arquitetura de Sistemas - SAS-PB da Diretoria de Sistemas de Informação da Prodabel - DSI.

O Processo de *Software* da PBH - PSP é um Modelo de Referência de desenvolvimento de *software* baseado principalmente no Guia de Projetos de Software com Práticas de Métodos Ágeis para o SISP [SISP15], no *Scrum* [SCSU13] e na Esteira de Integração Contínua - EIC [EICV2], que é o principal método de publicação de *software* da Prodabel. Ele está organizado em ciclos de atividades de desenvolvimento iterativos e incrementais, que são:

- Atividades de planejamento e análise inicial da demanda;
- Atividades de construção da *Release*;
- Atividades de transição;
- Atividades de gestão de ambientes de TI.

As definições tecnológicas de ambiente e infraestrutura, ferramentas, padrões e processos, e linguagens e frameworks que são utilizadas nesses ciclos estão documentadas no Catálogo de Tecnologia [CTGTEC].

1.2. O relacionamento do PSP com o *Scrum*

Devido a efetividade e ampla adoção de metodologias ágeis, tanto no meio privado quanto no público [SISP15], o PSP adotou as práticas do *Scrum* tendo como referência o seu guia oficial desenvolvido pelos próprios criadores [SCSU13], e os princípios do Manifesto Ágil [AM01], que valorizam:

- ***Indivíduos e interações*** mais que processos e ferramentas.
- ***Software em funcionamento*** mais que documentação abrangente.
- ***Colaboração com o cliente*** mais que negociação de contratos.
- ***Responder a mudanças*** mais que seguir um plano.

Nesses princípios há uma maior valorização dos itens à esquerda sobre os itens da direita.

A metodologia *Scrum* não é um processo para o desenvolvimento de *software* durante a etapa de implementação. Ao contrário disso, ele é um *framework* dentro do qual você pode empregar vários processos ou técnicas que se concentra em descrever como os membros da equipe devem trabalhar para produzir um sistema flexível em um ambiente de mudanças constantes. Portanto, ele é um *framework* para desenvolver, entregar e manter produtos de qualquer tamanho e complexidade.

O *Scrum* define papéis, eventos, artefatos e as regras que os unem e os mantém integrados. Ele é baseado no empirismo e possui três pilares básicos: transparência, inspeção e adaptação.

Os princípios elencados na enumeração abaixo devem ser utilizados nas atividades do PSP, para que seja possível ter efetividade na adoção dos métodos ágeis:

- Buscar simplicidade na medição do desenvolvimento de *software*;
- Simplificar o ônus de gestão e controle de mudanças de forma a minimizar os impactos sobre a agilidade do processo de desenvolvimento;
- Aceitar/Receber incrementos de *softwares* prontos e homologados;

- Focar os esforços na qualidade e na entrega de sistemas;
- Manter a simplicidade em todas as fases do desenvolvimento: Elimine tudo aquilo que for desnecessário, para não onerar e sobrecarregar o processo, as pessoas e o produto;
- O valor agregado é a percepção pelo Dono do Produto (*Product Owner*) - PO de que um objetivo específico do negócio foi satisfeito. Certifique-se que os acordos firmados e os produtos entregues agregam valor ao negócio;
- Estabeleça um foco nos itens de solução, nas necessidades e nos problemas, para que a energia de trabalho se concentre apenas nos objetivos e acordos firmados;
- A entrega de valor é efetivada, através do atendimento das necessidades e expectativas do cliente.

Neste documento é apresentado o PSP, que é o Modelo de Referência para a construção de projetos de *software* para Prodabel/PBH. Todos os *templates* de artefatos, materiais relacionados assim como referências de interesse em projetos ágeis encontram-se no Portal do PSP em [PSPPBH].

2. ESCOPO DO DOCUMENTO

2.1. Público alvo e serviços abordados

Este documento serve como referência para equipes terceirizadas e equipes internas da PBH/Prodabel no que diz respeito ao desenvolvimento e sustentação de seus sistemas.

O seu foco é a apresentação do Modelo de Referência do Processo de *Software* da PBH - PSP para construção de projetos de *software* que pode ser adaptado para outras atividades como a realização de sustentação de sistemas de informação já existentes, tais como a realização de pequenas correções ou evoluções de curto prazo que não demandam um projeto.

Entende-se como Modelo de Referência um documento orientador e não prescritivo, de forma a manter a autonomia dos projetos e times, que podem ser adaptados conforme suas características e particularidades, fomentando a melhoria contínua da aplicação de métodos ágeis.

O Modelo de Referência abordado contemplará os seguintes serviços:

- **Serviços de Apoio ao Gerenciamento de Projetos de *Software***

Envolvem tarefas de acompanhamento e controle de projetos. Esse serviço descreve atividades típicas do Escritório de Projetos (EPP).

- **Serviços de Desenvolvimento e Sustentação de *Software***

Envolvem atividades típicas de engenharia de *software* como elicitação de requisitos, implementação, testes e implantação de sistemas de informação; Atividades do modelo de execução de serviços de desenvolvimentos de *software*; Atividades e práticas que auxiliam no desenvolvimento de projetos de *software*.

- **Serviços de Apoio à Avaliação de Qualidade**

Envolvem tarefas de avaliação de qualidade de *software* e seus artefatos, tais como, realizar testes, avaliar arquitetura, modelos de dados, desempenho, documentação, critérios de aceitação, entre outros.

- **Serviços de Infraestrutura de TI**

Envolvem atividades de configuração, disponibilização e administração de infraestrutura física e lógica, necessária à entrega contínua de sistemas de informação, bem como ao armazenamento de dados, monitoramento, entre outros.

2.1. Limitações

Existem algumas limitações impostas pela concepção deste documento. Abaixo estão relacionados os temas que não foram tratados, resolvidos ou exauridos:

- Transição entre metodologias ou paradigmas de desenvolvimento de *software*;
- Métricas de *software*. Os serviços de métricas e análise de pontos de função são executados conforme Manual de Prática de Contagem do IFPUG (CPM) e Roteiro de Métricas do SISP, ou conforme descrito especificamente em cada contrato com fornecedores externos. Alguns itens que não estão especificados nestes roteiros e já foram motivos de estudo pela PBH estão tratados no Detalhamento de Cenários de Métricas da PBH, disponível no Portal do PSP [PSPPBH];
- Implantação de governança corporativa e de tecnologia da informação;
- Não mitiga riscos da contratação de empresa de desenvolvimento de *software*;
- Não contempla todos os aspectos do gerenciamento de um projeto, tais como aquisições, seleção e aprovação de projetos de *software*;
- O PSP não aprofunda nas Atividades de Gestão da Ordem de Serviço, como, Abertura de Ordem de Serviço, Monitoração de Cláusulas Contratuais, Métricas de *Software* e Faturamento. Estas atividades devem seguir as orientações do Fiscal do Contrato e do Escritório de Projetos - EPP. Entretanto, este documento sugere algumas ideias e subsídios para gestão de riscos, elaboração de termo de referência, disseminação da cultura ágil na Prodabel/PBH, entre outros.

3. GRUPOS DE ATIVIDADES DOS PROJETOS

A execução de projetos de *software* tratadas no âmbito da PBH baseiam-se essencialmente nos seguintes grupos de atividades:

- **Análise Inicial da Demanda**

Fase em que são levantados e descritos junto ao cliente os objetivos de negócio e o escopo do produto. Também é feita a definição da sua arquitetura básica e estimativa de tamanho e prazo para a sua construção.

A demanda por um projeto de software, em geral, nasce na Diretoria de Atenção ao Usuário (DAU). Após isso, a área de Soluções de Negócio da Diretoria de Sistemas (DSI), juntamente com o PO e com apoio da Equipe Técnica da Prodabel, elaboram o Documento de Análise de Demanda, que contempla aspectos relacionados a problemas, objetivos de negócio, necessidades, processos de negócio, expectativas, entre outros e, por fim, registra uma proposta de solução, a qual envolve elementos tecnológicos, tais como interoperabilidade com outros sistemas, descrição das características-chaves do produto, descrição dos requisitos de ambientes, modelo de arquitetura, descrição dos requisitos do produto, entre outros, além de fazer uma estimativa do tamanho e dos prazos envolvidos para o desenvolvimento do produto.

- **Planejamento da Demanda**

Nesta fase é feito o entendimento da demanda e do seu escopo pelos executores e a distribuição em diferentes Releases, formando um *roadmap* do produto.

- **Acompanhamento do Projeto**

Contempla atividades de acompanhamento da execução do projeto, de forma que possíveis problemas possam ser identificados no momento adequado e que possam ser tomadas ações preventivas e corretivas, quando necessário, para controlar a

execução do projeto. Estas atividades contemplam identificação e comunicação de desvios e impedimentos e apoio à equipe do projeto para garantir entregas definidas.

- **Atividades do Ciclo de Desenvolvimento**

Contempla as atividades relacionadas a construção e garantia de qualidade do *software* como a especificação, implementação e validação dos objetivos de negócio e características-chaves do produto contidas nas *Sprints* de cada *Release*. É formada por atividades como: codificação, testes, homologação e validação, entre outros.

- **Atividades de Transição**

Contempla atividades que garantam a implantação de cada *Release* em produção, avaliação dos resultados obtidos e das condições de entrega, suporte do produto e operação assistida.

- **Atividades de Gestão de Ambientes de TI**

Contempla atividades para viabilizar a criação de ambientes de TI, a entrega contínua facilitando a sua disponibilidade e implantação em cada incremento do produto de *software*.

A *Figura 1* resume o fluxo de execução dos projetos de *software* da PBH que envolve os grupos de atividades descritos.

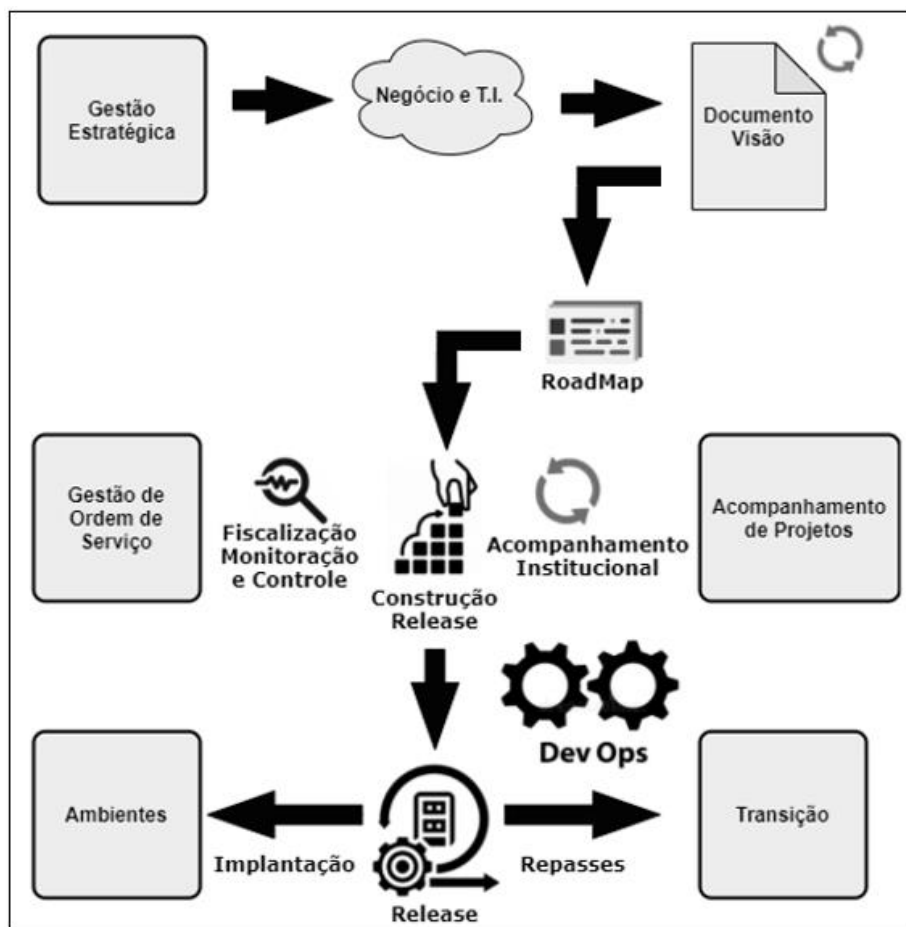


Figura 1. Fluxo de execução dos Projetos de Software da PBH.

4. MODELO DE REFERÊNCIA

O Modelo de Referência do Processo de *Software* da PBH, proposto neste documento e representado na *Figura 2*, se baseia no desenvolvimento de *software* em ciclos de entregas incrementais e iterativos, por meio de *Sprints* e *Releases*. Esses ciclos envolvem a execução dos grupos de atividades mencionados na seção anterior que são:

- Atividades de Planejamento (análise inicial da demanda);
- Atividades de Construção do Release;
- Atividades de Transição;
- Atividades de Gestão de Ambientes de TI;
- Atividades de Acompanhamento do Projeto.

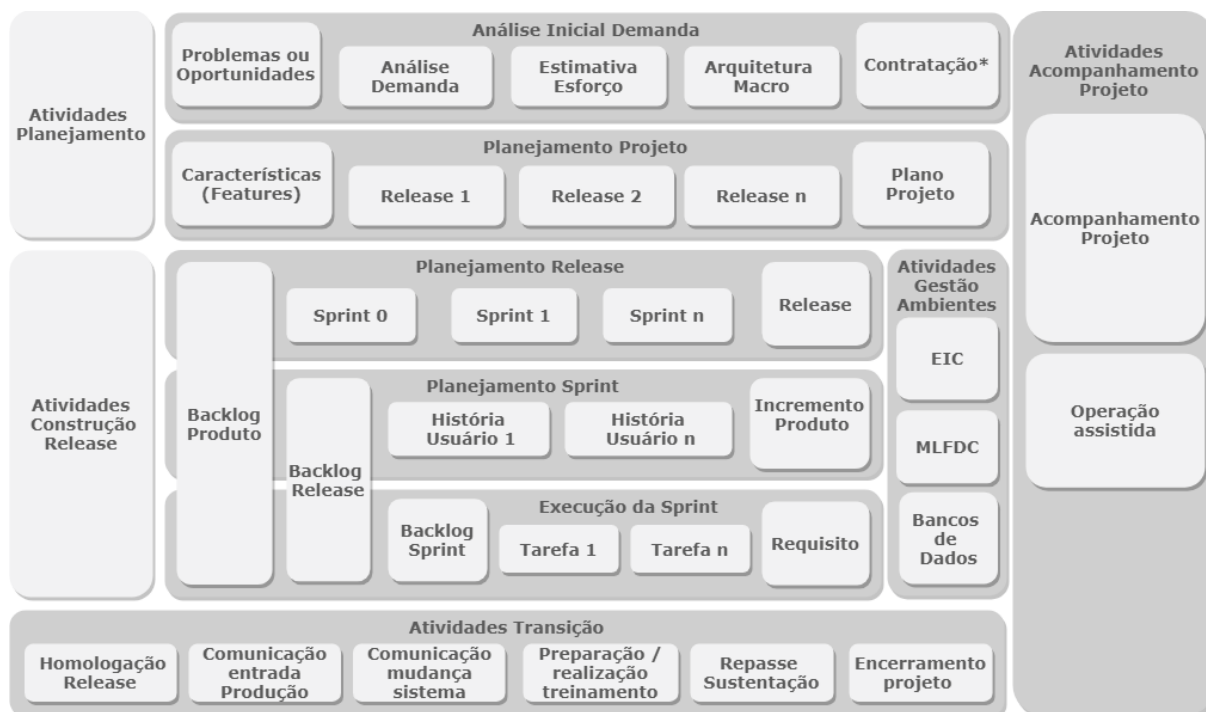


Figura 2. Modelo de Referência para Construção de Projetos de Software

Os principais benefícios da adoção deste modelo são:

- Incorporação de práticas ágeis;
- Redução dos riscos associados ao projeto através do *feedback* contínuo fornecido pelos mecanismos do *Scrum* (Reuniões diárias, *Reviews*, etc);
- Maior adaptação a mudanças durante a construção de sistemas;
- Incorporação de processos automatizados de publicação e de verificação da qualidade dos produtos (*DevOps*);
- Simplificação dos processos e papéis exercidos;
- Maior agilidade nas alterações realizadas nos projetos e sistemas;
- Encurtamento das entregas com a percepção de valor mais rapidamente pelo PO;
- Replanejamento constante.

Nos próximos tópicos serão abordados os papéis envolvidos e os detalhes que permeiam cada grupo de atividades de construção do projeto do Modelo de Referência.

4. PAPÉIS ENVOLVIDOS

O *Quadro 1* descreve os principais papéis envolvidos nas atividades descritas no Modelo de Referência. Alguns papéis foram extraídos diretamente da metodologia de desenvolvimento ágil *Scrum* [SCSU13].

Papel	Responsável pelo Papel
Gerente de Projeto	Prodabel
Gerente de Relacionamento	Prodabel
Analista de Métricas	Prodabel
Equipe Técnica da Prodabel (Analistas das áreas de Arquitetura de Sistemas, Infraestrutura, Segurança, Geoprocessamento, etc)	Prodabel
PO	Cliente
<i>Scrum Master</i>	Fornecedor ¹
<i>Time Scrum</i>	Fornecedor ¹
Time de Desenvolvimento	Fornecedor ¹
¹ Em projetos desenvolvidos por equipes internas o responsável pelo papel será a própria Prodabel.	

Quadro 1. Principais papéis envolvidos no Modelo de Referência.

Os utilizadores deste guia podem definir outros papéis além dos descritos no *Quadro 1* conforme necessário. No *ANEXO II - PAPÉIS DA EQUIPE NO MODELO DE REFERÊNCIA* são descritos os papéis e suas responsabilidades de forma detalhada, assim como inclui outros papéis também envolvidos no Modelo de Referência.

5. DETALHAMENTO DAS ATIVIDADES

Esta seção descreve, de forma detalhada, as atividades realizadas durante a execução do ciclo de desenvolvimento do Modelo de Referência adotado por este documento. Serão descritos os Grupos de Atividades de Planejamento (análise inicial da demanda), Construção da *Release*, Atividades de Transição, Atividades de Gestão de Ambientes de TI e Atividades de Acompanhamento de Projeto. Cada atividade é descrita utilizando os seguintes atributos: nome da atividade, objetivo, descrição, responsável, artefatos de entrada e saída.

5.1. Atividades de planejamento

O fluxo de valor do produto de *software* inicia-se no grupo de atividades de planejamento com a análise inicial da demanda conforme detalhamento na *Figura 3*. As atividades desse grupo consistem na identificação e compreensão das necessidades, problemas, objetivos de negócio, características-chaves do produto (*features*) e oportunidades. Também está incluído nesse grupo a elaboração do *Roadmap de Releases*.

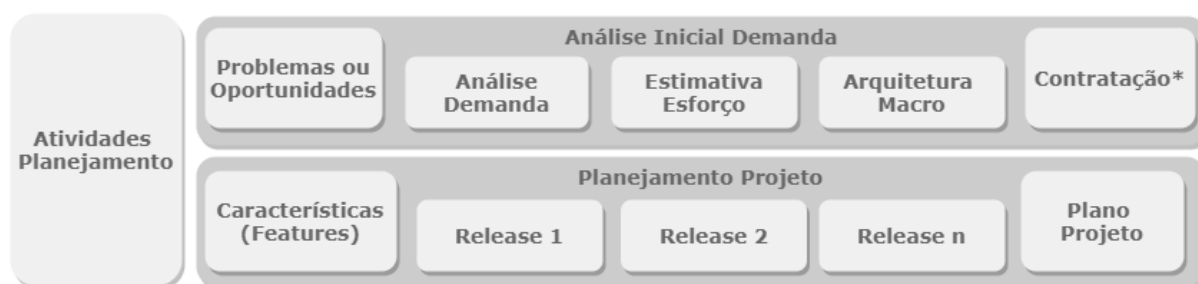


Figura 3. Detalhamento das atividades de Planejamento.

O documento de Análise Inicial da Demanda, elaborado pela Equipe Técnica da Prodabel de Análise de Negócio, reúne todo o conhecimento adquirido do projeto por meio de técnicas de elicitación de requisitos de *software* e análise de negócio. Além disso, ele contempla a discussão prévia da arquitetura inicial do *software*.

Após a definição da forma de construção do projeto (contratação de fornecedor externo ou equipe interna) e da realização da Análise Inicial da Demanda, o projeto é inserido na carteira

do Escritório de Projetos e inicia-se o planejamento de cada entrega (*Release*) o qual consiste no agrupamento de objetivos de negócio e características-chaves do produto por meio de critérios de priorização e de importância, ou seja, é planejado o *Roadmap*.

Cada *Release* é o lançamento de uma determinada versão de *software* incrementada com novas funcionalidades, as quais satisfazem determinados objetivos de negócio e características-chaves do produto de *software*.

A área requisitante do projeto de *software*, representada pelo PO e clientes, deve atuar principalmente neste grupo de atividades apoiada pela Equipe Técnica da Prodabel. Ela deve fornecer a compreensão do negócio, das necessidades, dos objetivos de negócio e dos requisitos do produto de *software*.

Portanto, o grupo de atividades de planejamento irá definir as atividades para a construção dos artefatos que contém os objetivos de negócio e as características-chaves do produto de *software* produzidos pela área requisitante.

5.1.1. Análise Inicial da Demanda

Atividade: Análise Inicial da Demanda

Objetivo: Na análise inicial da demanda é construído o documento de Análise Inicial da Demanda onde são descritas as necessidades, expectativas, objetivos específicos de negócio e proposta de solução para o projeto, além de prospectar uma estimativa do esforço do projeto.

Descrição: Nesta atividade, entende-se que existe um problema a ser resolvido ou uma oportunidade a ser aproveitada. Para atender a essa demanda, um sistema será construído. O documento de Análise da Demanda deve buscar responder às seguintes questões:

- Qual problema, oportunidade, benefícios e necessidades que este produto/projeto resolve ou aproveita?
- Quais são os objetivos específicos de negócio do produto (Objetivos)?
- Como clientes e usuários poderão atingir os objetivos de negócio (Impacto)?
- Quais são as características-chaves (ou macro funções) do produto final: quais funcionalidades de negócio, performance, segurança, escalabilidade, precisam ser entregues (*Features* - Entregáveis)?

- Quais são os principais riscos, premissas e restrições do projeto?
- Qual é a expectativa de custos, prazos e estimativa de esforço para o desenvolvimento do produto?
- Qual a arquitetura macro do projeto?
- Quais são os diferenciais em relação à solução atual ou outra existente?
- Os objetivos específicos do negócio devem estar sempre no foco do projeto (agregação de valor para o cliente), pois são a chave para tomar boas decisões sobre custo, escopo e prazos, tanto no início e depois, quando as mudanças acontecem.
- Quais processos serão aplicáveis à Metodologia de Adequação à Lei Geral de Proteção de Dados - LGPD [MALGPD20]?

O resultado é o Documento de Análise de Demanda, que servirá como entrada para a próxima etapa de planejamento do projeto.

Outras atividades também são realizadas até que seja possível incluir o projeto na Carteira de Projetos, atividades sequenciadas podem ser vistas na *Figura 4* abaixo:



Figura 4. Representação do fluxo de atividades envolvidas na produção do Documento de Visão.

Caso necessário, o Gerente de Relacionamento deve fazer a solicitação de análise da demanda com estimativa de esforço para a Área de Negócio da Prodabel.

Notas:

- A análise da demanda deve ser feita utilizando uma linguagem de fácil entendimento, visto que este será um elo de comunicação com a equipe do projeto;
- O PO deve realizar reuniões com a equipe de análise de negócio até definir a visão do produto. Algumas técnicas que contribuem para a elaboração da visão do produto são:
 - Mapeamento do Processo de Negócio;
 - Reunião de entendimento, *Brainstorming*;
 - Veja mais informações sobre as técnicas citadas acima no *ANEXO I - CONCEITOS E FUNDAMENTOS SOBRE O MODELO DE REFERÊNCIA*.

Após inclusão do projeto na carteira de projetos no Sistema Portal da Gestão [PGESTPBH], o sistema também deve ser cadastrado no SIS [SISPBH].

Responsável:

- PO;
- Gerente de Relacionamento;
- Equipe Técnica da Prodabel (Análise de Negócio).

Entradas:

- Demanda do Cliente.

Saída:

- Ata de Reunião;
- Documento de Análise Inicial de Demanda feito pela Equipe Técnica da Prodabel (Área de Negócio);
- Planilha de Estimativa de esforço feito pela Equipe Técnica da Prodabel (Área de Negócio);
- Documento de Arquitetura Macro;
- Proposição de Projeto/Ação preenchido no Portal da Gestão e demanda inserida na respectiva Carteira;
- Sistema cadastrado no SIS [SISPBH]
- Sistema cadastrado no Portal da Gestão [PGESTPBH].

5.1.2. Planejamento do Projeto

Atividade: Planejamento do Projeto

Objetivo: Construir o planejamento macro de entregas ou plano cronológico de liberação dos *Releases* (versões do produto), tendo em vista o MVP (Produto Mínimo Viável).

Descrição: Deve-se dividir objetivos de negócio ou macro funções do produto em partes entregáveis, tendo em vista o MVP, por ordem de prioridade. Conforme o Modelo de Referência

deste documento, as partes são as *Releases* que, por sua vez, são construídas a partir das características-chaves do produto priorizadas e ordenadas.

As características-chaves do produto são extraídas do Documento de Análise Inicial de Demanda que devem estar alinhadas com as necessidades, as expectativas do cliente e, por fim, aos objetivos específicos de negócio.

Orientações para a elaboração do plano de entregas dos diversos *Releases*:

- Estabeleça um *ranking* dos objetivos e das características-chaves do produto a partir do que mais agrega valor para o que agregue menos;
- Agrupe os itens (objetivos e características-chaves) por ordem de prioridade, na quantidade compatível com a capacidade de produção do projeto e no tempo disponível para o desenvolvimento. Exemplos de escala de prioridade: Imprescindível, importante, desejável;
- Estabeleça uma cronologia de entregas ou a periodicidade;
- Organize reuniões em que os envolvidos participem ativamente da construção do *Roadmap*;
- Divulgue o *Roadmap* para todos os envolvidos.

Ao final de cada *Release*, o *Roadmap* poderá ser revisado e atualizado.

Por fim, o plano de entrega das versões é o registro cronológico e pré-ordenado da entrega de cada parte da solução, devidamente aprovado pelo PO e clientes. A *Figura 5* representa a esquematização das entregas de versões de produtos que são compostas por *Releases* construídas ao longo de um conjunto de *Sprints*.

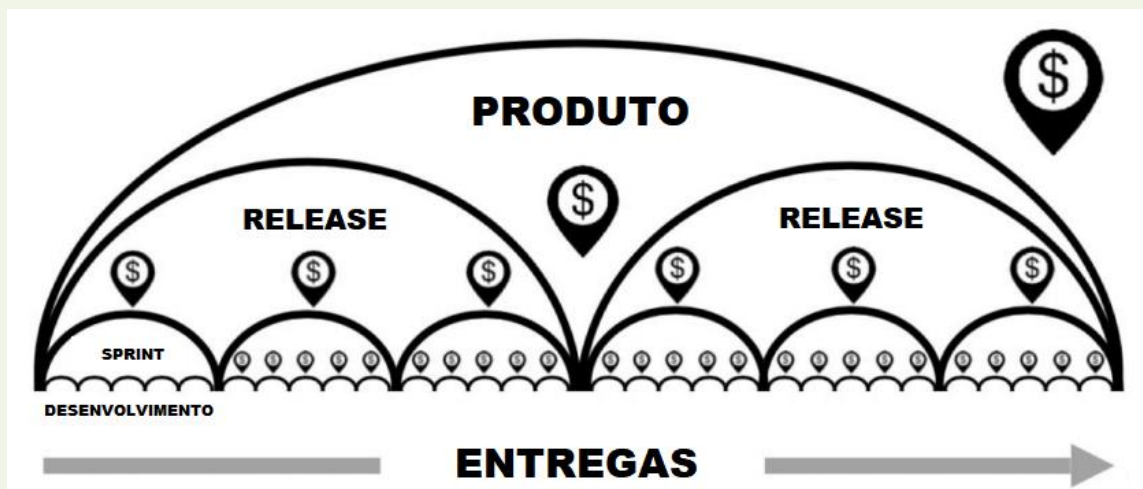


Figura 5. Esquemática das entregas de versões do produto.

Nesta fase pode ser construído o Plano de Projeto pelo Gerente do Projeto. O Plano de Projeto não é pré-requisito para iniciar a Execução do Projeto, ou seja, enquanto o mesmo está sendo elaborado e validado, as reuniões para detalhamento dos requisitos podem ser executadas em paralelo.

No Planejamento do Projeto deve constar os marcos de entregáveis previstos no Acordo de Serviço, sendo que após a sua validação e homologação pelo PO e Prodabel.

Notas:

- Definir a arquitetura inicial do *software* em conjunto com o Time de Desenvolvimento e Equipe Técnica da Prodabel (Analista de Infraestrutura de TI), caso necessário.
- Agregação de valor ao negócio: a fórmula do sucesso para os projetos ágeis (vide os princípios do manifesto ágil [AM01]) é priorizar pelo valor de negócio, buscando sempre a simplicidade (na especificação, implementação e execução).

Não é recomendável executar *Sprints* apenas técnicas (resolver débitos técnicos, por exemplo). Pode-se até mesclar revisões técnicas, mas sempre incluindo entregas que agreguem valor ao cliente.

Responsável:

- PO;
- Gerente de Projeto;

- Equipe Técnica da Prodabel.

Entradas:

- Documento de Análise Inicial da Demanda;
- Documento de Arquitetura Macro.

Saídas:

- *Roadmap* do Produto (Obrigatório);
- Plano de Projeto.

5.2. Atividades de construção do *Release* (ciclo de desenvolvimento)

No grupo de atividades de construção da *Release*, que envolve o ciclo de desenvolvimento do produto (atividades listadas na *Figura 6*), será produzida uma ou mais *Releases* do produto, que são conjuntos de *Sprints*. As atividades das *Sprints* são executadas pelo Time de Desenvolvimento com o apoio da Equipe Técnica da Prodabel, quando necessário.



Figura 6. Detalhamento das atividades de construção da *Release*.

Nos tópicos abaixo serão descritos em detalhes as atividades envolvidas na construção do projeto de acordo com o Modelo de Referência do PSP. A *Figura 7* exhibe o resumo das

práticas do *Scrum* que também são adotadas durante as atividades de construção da *Release* no PSP.



Figura 7. Práticas do Scrum por papéis desempenhados [SCSU13].

5.2.1. Atividades de Planejamento da *Release*

Planejar a *Release* é determinar o escopo da entrega de uma versão do produto. Posteriormente, a definição desse escopo se transformará em atividades menores que podem ser chamadas de itens de *Backlog*, que serão priorizados, estimados e separados em *Sprints* na atividade de definição do plano da *Release*.

A *Release* é liberada na forma de uma versão funcional e pronta para homologação, podendo ser disponibilizada no ambiente de produção de acordo com a necessidade do Cliente. Recomenda-se que uma *Release* tenha a duração de 3 *Sprints* e que não seja feito *Releases* de somente uma *Sprint*.

O início de cada planejamento da *Release* é marcado pela elaboração do *Backlog* do Produto, técnica extraída do *Scrum* [SCSU13]. O *Backlog* desdobra objetivos de negócio e características-chave do produto, estabelecidas na Análise Inicial da Demanda, em requisitos

de *software* (funcionais e não funcionais) e tarefas técnicas importantes para a construção dos produtos de *software* que são descritos através de Histórias de Usuário.

As Histórias de Usuário podem pertencer a um ou mais *Backlogs* de *Sprint*. Elas são orientadas por critérios de priorização de seleção de requisitos e cronograma do projeto. Nessa fase também deve ser definido o conceito de pronto e qual será a duração das *Sprints*.

5.2.1.1. Planejar *Release*

Atividade: Planejar o *Release*.

Objetivo: Construir e disponibilizar o *Backlog* do Produto, que é a lista priorizada dos itens necessários para o desenvolvimento e entrega do produto de *software* e definir o Plano do *Release* com a meta a ser alcançada em função dos objetivos de negócio e características-chaves do produto.

Descrição:

No Planejamento do *Release* é realizada a composição do *Backlog* do Produto representando tudo que é necessário para desenvolver um produto de valor agregado ao negócio. Ele é composto por uma lista de todos os requisitos (funcionais e não funcionais), funções, tecnologias, melhorias e correções de defeitos que serão efetuadas no produto para versões futuras. Além disso, nesta atividade é importante estabelecer uma meta para o *Release* em função dos objetivos de negócio e do prazo de acordo com o que foi definido no *Roadmap* do Produto.

Nesta etapa também é necessário identificar os itens do *Backlog* que serão aplicáveis à Metodologia de Adequação à LGPD. Esses itens deverão conter explicitamente os requisitos necessários para o atendimento das exigências relacionadas a metodologia LGPD, conforme informações disponíveis em [MALGPD20].

***Backlog* do Produto:**

Os itens do *Backlog* do Produto são definidos pelo PO e conta com a participação do Time de Desenvolvimento para identificar novos itens e auxiliar no registro e especificação dos requisitos do *Backlog*.

O PO revisa o *Backlog* do Produto e prioriza os seus itens. Cada novo item é inserido, priorizado ou removido do *Backlog* pelo PO, mediante os objetivos de negócio.

A lista de itens do *Backlog* do Produto será utilizada pelo Time de Desenvolvimento para implementar o produto através de cada *Sprint*.

O *Backlog* do Produto é dinâmico no sentido de que ele está constantemente sendo priorizado e alterado conforme necessidade do negócio buscando agregar valor ao cliente (alcançar os objetivos de negócio). Esse refinamento é conhecido por *Backlog Grooming* e gera as Histórias de Usuários preparadas para serem construídas nas próximas *Sprints*.

Durante um *Grooming Session*, artefatos de planejamento poderão ser refinados e atualizados de acordo com eventuais mudanças de negócio no projeto.

O *Backlog* do Produto deve ser organizado/gerenciado através do quadro *Kanban* através da ferramenta indicada pela Prodabel para este fim [GITPBH].

Plano da Release:

No plano da *Release* é realizada a estimativa de tamanho e esforço, utilizando técnicas de estimativa de práticas ágeis, para implementar os itens do *Backlog* do produto selecionados para a *Release*. Com a estimativa de tamanho e esforço definida, é possível estimar a duração e a quantidade de *Sprints* da *Release*.

Por fim, as informações do Plano da *Release* podem ser inseridas em um Gráfico de *Burndown*, para o seu devido acompanhamento durante o projeto.

Responsável:

- PO;
- Gerente do Projeto.

Entradas:

- Documento de Análise Inicial da Demanda;
- Plano de Projeto.

Saída:

- Plano da *Release*;
- *Backlog* do Produto.

5.2.1.2. *Sprint 0*

Atividade: *Sprint 0*

Objetivo: Realizar as atividades iniciais de configuração e preparação do projeto necessárias para o início da construção do produto.

Descrição:

Para o início das atividades de construção pode ser realizado um conjunto de atividades de preparação e configuração do projeto no primeiro *Release* que também é conhecido como *Sprint 0*. As principais atividades dessa *Sprint* estão relacionadas abaixo:

- Cadastro do Sistema no SIS;
- Cadastro dos itens de configuração do sistema no *Service Desk Manager* - SDM da Prodabel (o cadastro é realizado automaticamente após a validação do cadastro do sistema pelo time de gerenciamento de cadastro do SIS da Prodabel);
- Elaboração do Modelo de Arquitetura inicial do sistema em conformidade com as tecnologias recomendadas no Catálogo de Tecnologia. Após isso, é necessário incluir o Modelo de Arquitetura inicial no cadastro do sistema no SIS e solicitar a validação do pelo time de Arquitetura de Sistemas da Prodabel¹;
- Elaboração do Plano de Teste inicial;
- Criação de usuários e acessos do time de desenvolvimento no AD (área de solicitação: DAU) e LDAP (área de solicitação: SOI) com as permissões de acesso na VPN (área de solicitação: SMI), nas ferramentas de *DevOps* e de Integração Contínua, e outras ferramentas necessárias para apoio no desenvolvimento do projeto que estejam indicadas no Catálogo de Tecnologia [CTGTEC];
- Criação dos usuários de banco de dados do sistema e do *Liquibase* para os ambientes de QA, Homologação, *Staging* (ambiente opcional) e Produção. Para mais detalhes sobre os ambientes ver a seção 5.4 Atividades de Gestão de Ambientes de TI;
- Cadastro dos segredos utilizados pelo sistema no *Vault* (ex: segredos de banco de dados);

- Criação do esboço da imagem base do *container* com a *stack* tecnológica do sistema e realização da estimativa inicial de utilização de recursos computacionais para solicitação da criação dos ambientes iniciais;
- Solicitação de criação e configuração dos repositórios e ambientes do sistema na Esteira de Integração Contínua – EIC. Para a criação dessa solicitação é necessário que o Modelo de Arquitetura inicial já tenha sido aprovado pela Prodabel¹. Consultar a seção de Primeiros Passos na documentação da EIC em [EICV2]² e a seção 5.4 Atividade de Gestão de Ambientes de TI deste documento;
- Instalação das ferramentas necessárias para automatização de Testes Funcionais e de Desempenho no ambiente de desenvolvimento e configuração das ferramentas de análise estática de código e execução de testes unitários no *pipeline* de construção do sistema na EIC;
- Demais tarefas necessárias para o início da construção do produto.

Observações

¹O pedido de criação dos ambientes deve ocorrer imediatamente após a aprovação do Modelo de Arquitetura para evitar problemas de concorrência com a fila de atendimento do time EIC.

² Em situações específicas alguns projetos poderão ser criados através do MLFDC (Modelo Lógico Funcional do Datacenter) por meio de um Plano de Implantação, conforme previsto na seção 5.4 Atividades de Gestão de TI. Essa opção deve ser avaliada em conjunto com o time de Arquitetura de Sistemas da Prodabel.

Modelo de Arquitetura Inicial:

No Modelo de Arquitetura Inicial do Sistema é necessário definir as características técnicas de arquitetura. Recomenda-se que a arquitetura inicial já esteja definida após a conclusão da *Sprint* 0. Ela deve estar aderente ao Catálogo de Tecnologia da Prodabel/PBH [CTGTEC] vigente no momento da concepção do projeto e deve ser submetido para aprovação da SAS (Superintendência de Arquitetura de Sistemas).

Nesse Modelo também será necessário incluir os requisitos não funcionais de capacidade e desempenho do projeto, tais como, expectativa de número de usuários suportados e de acessos concorrentes, tempo de resposta máximo aceitável, volume de processamento por unidade de tempo, métricas de desempenho do ambiente operacional (CPU, memória e disco) e outros que

puderem ser estabelecidos inicialmente. Esses requisitos são necessários para o planejamento inicial dos Testes de Desempenho. Mais detalhes do processo de teste de desempenho na seção de artefatos do Portal do PSP [PSPPBH].

Plano de Teste inicial:

O Plano de Teste inicial deve definir a estratégia que será adotada para a execução dos testes. Essa estratégia é constituída por um conjunto de definições que irão orientar a execução do plano ao longo do ciclo de desenvolvimento e inclui a determinação dos níveis de testes que serão aplicados (Testes Unitários, Testes Funcionais Manuais, Testes Funcionais Automatizados, Testes de Integração de APIs, etc). Mais informações sobre estratégia de testes podem ser obtidas no item Estratégias e Abordagens de Teste em [FDTESTAGL]. Sobre a determinação dos níveis de teste mais adequados ao projeto podem ser consultados no item Abordagens de Teste em [FDTESTAGL].

O Plano de Teste inicial também deve conter a definição das pessoas da equipe que farão o papel de testadores manuais e que irão desenvolver os testes funcionais automatizados. Por último, devem ser incluídas também as ferramentas escolhidas que serão utilizadas para gerenciamento e automatização de testes ao longo do projeto. Essas ferramentas devem ser instaladas, configuradas e integradas aos ambientes e a EIC, conforme as orientações em [EICV2], durante a *Sprint 0*.

Ambientes:

As informações sobre os ambientes criados na *Sprint 0* podem ser consultadas na seção: 5.4. Atividades de Gestão de Ambientes de TI.

Responsável:

- PO;
- *Scrum Master*;
- Time de Desenvolvimento;
- Equipe Técnica da Prodabel;
- Gerente de Projeto.

Entradas:

- Documento de Análise de Demanda;

- *Backlog da Release*;
- *Roadmap do Produto*.

Saídas:

- Usuários e acessos criados no AD, LDAP e Banco de Dados (Obrigatório);
- Segredos registrados no *Vault* (Obrigatório);
- Sistema cadastrado no SIS (Obrigatório);
- Itens de configuração do sistema cadastrados no SDM (Obrigatório);
- Modelo de Arquitetura inicial validado pela Equipe Técnica da Prodabel (Arquitetura de Sistemas) e cadastrado no SIS (Obrigatório);
- Plano de Teste inicial validado pela Equipe Técnica da Prodabel (Arquitetura de Sistemas) (Obrigatório);
- Repositórios de controle de versão do projeto criados (Obrigatório);
- Imagem base inicial do *container* da aplicação (Obrigatório);
- Ferramentas de automatização de testes instaladas e configuradas em ambiente de desenvolvimento (Opcional);
- Ambientes das aplicações e bancos de dados criados, configurados e integrados à EIC;
- Plano de Implantação dos ambientes, caso seja adotado o MLFDC (Opcional).

5.2.2. Atividades de Construção das *Sprints*

Com o plano da *Release* estabelecido e as atividades prévias ao desenvolvimento realizadas na *Sprint* 0, é possível dar início à execução da primeira *Sprint* e posteriormente das subsequentes. A *Sprint* é uma das principais atividades do projeto, afinal, é por meio desses ciclos de trabalho que os itens de *Backlog* são transformados em um incremento de *software* funcional. As *Sprints* são executadas de forma iterativa e incremental. Na *Figura 8* é apresentado o resumo do fluxo de atividades envolvidas na execução de uma *Sprint*, que será abordado nos tópicos seguintes:

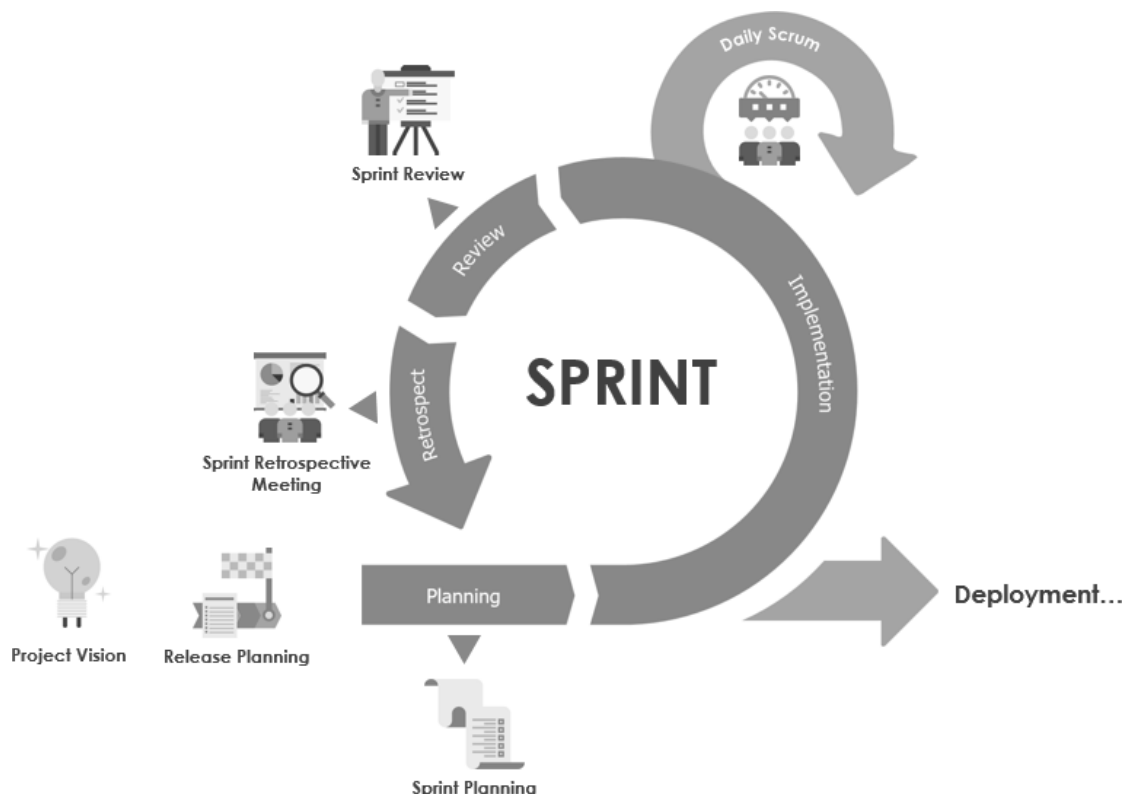


Figura 8. Fluxo de atividades realizadas durante a execução de uma Sprint.

Ao final de cada *Sprint* haverá uma parte executável do produto e a documentação relacionada, cuja parte executável deve ser homologada e aceita pelo PO.

O ciclo de pagamento dos projetos poderá variar entre *Sprint* e/ou a *Release* de acordo com o contrato firmado com o Fornecedor. Com isso, será feita a contagem detalhada de Pontos de Função, após obter o aceite pelo PO da entrega, para determinar o valor a ser pago considerando a parte executável entregue, ou seja, o “Produto Pronto”.

5.2.2.1. Planejar a *Sprint*

Atividade: **Planejar a *Sprint***

Objetivo: A finalidade principal da reunião é realizar o planejamento da *Sprint* a ser executada.

Descrição:

O trabalho a ser realizado na *Sprint* é planejado de forma colaborativa entre todo o Time *Scrum*

(*Scrum Master*, Time de Desenvolvimento e PO).

A reunião deve ser rápida e objetiva, buscando não comprometer mais de 8 horas de duração (*Time-box*) para o planejamento de uma *Sprint* de um mês de duração. Para *Sprints* menores, este evento é usualmente menor. O *Scrum Master* garante que o evento ocorra e que os participantes entendam seu propósito orientando o Time *Scrum* a manter-se dentro dos limites do *time-box*.

O planejamento da *Sprint* responde às seguintes questões:

- Qual o objetivo *Sprint*?
- O que pode ser entregue como resultado do incremento da *Sprint*?
- Como o trabalho necessário para entregar o incremento será realizado?

Tarefas gerais da primeira parte da reunião:

- Selecionar o *Backlog* da *Sprint* a partir do *Backlog* da *Release*: O PO, junto com os *stakeholders*, seleciona os itens prioritários do *Backlog* que serão trabalhados na *Sprint*.
- O PO pode realizar a inclusão, exclusão ou alteração nos itens do *Backlog* da *Sprint* nesta reunião. Alterações nos itens do *Backlog* da *Sprint* não são permitidas durante a execução da *Sprint*, a menos que acordadas entre o PO e o Time de Desenvolvimento, mediante troca de prioridades.
- São fatores que influenciam na seleção dos itens do *Backlog* que serão desenvolvidos na *Sprint*:
 - Prioridade/Valor;
 - Tamanho/Estimativa;
 - Duração da *Sprint*;
 - Restrições/Riscos;
 - Investimento/Custo.

Tarefas da segunda parte da reunião:

Na segunda parte da reunião, o Time de Desenvolvimento trata a questão do “como?”, ou seja, decidem como transformar os itens selecionados do *Backlog*, que foi definido na parte anterior da reunião, em um incremento de *software* funcional.

Nesse momento, a equipe estima o esforço e o PO define o sequenciamento e os critérios de aceitação para elas. Para a estimativa de esforço da equipe poderá ser utilizado uma técnica de estimativa, como por exemplo, o *Planning Poker*, que é uma técnica baseada no consenso prevista no *Scrum*.

Em seguida, os testadores identificam e criam as tarefas de testes que irão validar o desenvolvimento baseando-se nesses critérios. O Plano de Teste é atualizado com a inclusão das tarefas de teste definidas para a *Sprint*.

No item “Validação e verificação dos testes baseada em critérios de aceite” em [FDTESTAGL] podem ser consultadas as características que esses critérios devem abordar para serem testáveis.

O esforço de teste necessário para as tarefas de teste identificadas também deve ser contabilizado durante a estimativa de esforço de desenvolvimento das Histórias de Usuário. Por isso, deve ter também a participação dos testadores em sua definição.

O PO está presente nesta parte da reunião para esclarecer dúvidas sobre os itens do *Backlog* e para ajudar a efetuar mudanças, caso estas sejam necessárias. Outras pessoas da Equipe Técnica da Prodabel podem ser convocadas para a reunião, ex: Analista de Infraestrutura, Gerente de Projetos, Analistas de Banco de Dados, Analistas de Arquitetura de Sistemas, Analista de Métricas, Analista de Qualidade e Segurança, para fornecer apoio técnico ou específico do domínio em questão.

Notas:

- As tarefas de desenvolvimento e de testes são incorporadas ao *Backlog* da *Sprint*, para o seu devido acompanhamento pelo Time de Desenvolvimento.
- O prazo de entrega corresponde ao período de tempo que a Time de Desenvolvimento tem para executar a *Sprint*, realizar os testes funcionais e/ou de requisitos não funcionais e gerar a *build* potencialmente implantável.
- É importante negociar nesta reunião uma agenda de compromissos do Time de Desenvolvimento junto ao PO para o período de execução da *Sprint*.
- Os requisitos não funcionais de Capacidade e Desempenho identificados nas Histórias de Usuário devem ser complementados aos já existentes no Modelo de Arquitetura.

O *Backlog* deve ser organizado/gerenciado através do quadro *Kanban* ou quadro *Scrum* disponível no Git [GITPBH].

Responsáveis:

- PO;
- *Scrum Master*;
- Equipe Técnica da Prodabel (se necessário)
- Time de Desenvolvimento.

Entradas:

- Plano da *Release*;
- *Backlog* da *Release* (priorizado);
- Histórias de Usuário refinadas e detalhadas.

Saídas:

- *Backlog* da *Sprint* com tarefas de desenvolvimento priorizadas com critérios de aceitação definidos e tarefas de testes;
- *Backlog* do Produto (atualizado) inserido na ferramenta de gestão de *Kanban* indicada pela Prodabel [GITPBH];
- Requisitos não funcionais identificados;
- Plano de Teste com as tarefas de teste identificadas e Especificação de Teste de Desempenho atualizados.

5.2.2.2. Executar *Sprint*

Atividade: Executar a *Sprint*.

Objetivo: Construir e entregar um incremento de *software*, para inspeção pelo PO sendo um incremento de *software* uma versão funcional do *software*.

Descrição:

Após a escolha do *Backlog* da *Sprint*, caso necessário, o Time de Desenvolvimento em conjunto com o PO iniciará a especificação complementar das Histórias de Usuário que compõem a *Sprint*.

Nesta atividade deve-se evitar as alterações no *Backlog* da *Sprint* que causem impactos negativos no alcance de sua meta. Portanto, mudanças que impactem na meta devem ser registradas no *Backlog* do produto ou da *Release* para serem implementadas em uma *Sprint* futura. Devemos manter o ambiente estável e favorável para o cumprimento da meta. Mudanças que não causarem riscos à meta da *Sprint* poderão ser realizadas durante a *Sprint*.

Em cada *Sprint* são executadas todas as atividades que forem necessárias para o desenvolvimento do incremento de *software*, tais como:

- Definição/especificação de requisitos;
- Desenvolvimento das Histórias de Usuário;
- Criação e execução de testes automatizados e execução de testes manuais previstos nos Planos de Testes;
- Criação / execução dos *scripts* de teste de desempenho;
- Criação / alteração de modelos de dados conforme necessidades previstas nas Histórias de Usuário;
- Alterações no Modelo de Arquitetura;
- Implementação do código fonte;
- Outros.

Estas atividades são necessárias porque o objetivo da *Sprint* é entregar uma parte do produto final para inspeção do PO.

O *Scrum Master* deverá facilitar o trabalho do Time de Desenvolvimento, removendo desvios e impedimentos encontrados e garantindo o bom andamento do trabalho.

Durante a *Sprint*, a equipe poderá sentir necessidade de consultar o PO, o qual deverá estar disponível de forma a não comprometer os objetivos propostos.

Sobre os prazos para execução:

Para projetos executados com fornecedores externos o prazo de execução da *Sprint* para as atividades de desenvolvimento será de 3 semanas (recomenda-se que dessas 3 semanas, uma semana seja utilizada para testes em QA), após essas 3 semanas o cliente terá 3 dias para validação, ou seja, a *Sprint* passa a ter 3 semanas e 3 dias.

Sobre as atividades de Teste:

Outro aspecto importante na execução da *Sprint* é a criação e execução dos testes. No decorrer da *Sprint* poderão ser criados e executados os testes para as tarefas de testes do *Backlog* da *Sprint* obedecendo os níveis especificados no Plano de Teste.

Os Desenvolvedores do Time de Desenvolvimento poderão criar testes unitários e de componentes para as funcionalidades no ambiente de Desenvolvimento podendo fazer o uso de boas práticas de desenvolvimento de testes como o *Test Driven Development* - TDD. Neste ambiente também poderão ocorrer o desenvolvimento de testes automatizados funcionais e de integração de APIs. Após concluído o desenvolvimento das funcionalidades, elas serão publicadas no ambiente de QA onde os Testadores e Analistas de QA poderão realizar os testes funcionais de confirmação manuais, para validar os critérios de aceitação, e realizar testes exploratórios.

Cada tipo de teste funcional deve ser criado e executado por um perfil de testador mais adequado conforme os papéis e responsáveis estabelecidos no Plano de Teste. Os tipos de teste e as técnicas relacionadas a cada um deles que podem ser aplicados estão disponíveis em [FDTESTAGL].

O *Quadro 2* estabelece a relação entre o tipo de teste, o responsável pela construção e execução e o ambiente em que os testes são construídos e executados:

Tipo de Teste	Responsável Construção / Ambiente	Responsável Execução / Ambiente
Teste de Unidade e de Componente	Desenvolvedor / Ambiente de Desenvolvimento	Desenvolvedor / Ambiente de Desenvolvimento
Teste Funcional Manual	Analista de QA / Ambiente de QA	Analista de QA / Ambiente de QA
Teste Funcional Automatizado	Analista de QA / Ambiente de Desenvolvimento	Analista de QA / Ambiente de QA
Teste de Aceite do Sistema ou do Usuário	PO / Ambiente de HML	PO / Ambiente de HML
Teste de Regressão automatizado (Teste de Unidade e Componente)	Desenvolvedor / Ambiente de Desenvolvimento/QA	Pipeline EIC / Ambiente de QA/HML

Teste de Desempenho	Analista de Teste de desempenho / Ambiente de QA	Analista de Teste de desempenho / Ambiente de *Produção
Teste de Segurança	N/A	Ferramentas de análise de vulnerabilidade / Ambiente de *Produção

Quadro 2. Relacionamento entre os tipos de testes que podem ser aplicados, os responsáveis pela construção e execução e os ambientes em que são construídos e executados.

**O ambiente de Produção pode ser representado por um ambiente análogo com características similares quando não for possível executar os testes em Produção.*

Testes unitários e de componentes produzidos pelo time de desenvolvimento nos ambientes de Desenvolvimento poderão ser desenvolvidos e integrados ao *pipeline* de integração contínua do projeto. Eles serão executados em conjunto com outros procedimentos da EIC e irão prevenir erros de integração de componentes e de regressão ao longo do ciclo de vida do projeto.

A *Figura 9* representa a integração do desenvolvimento com a EIC no processo de construção do *build* de um incremento de *software*. Mais detalhes sobre os processos de integração contínua, testes de regressão e demais práticas ágeis de testes de *software* podem ser consultadas em [FDTESTAGL].

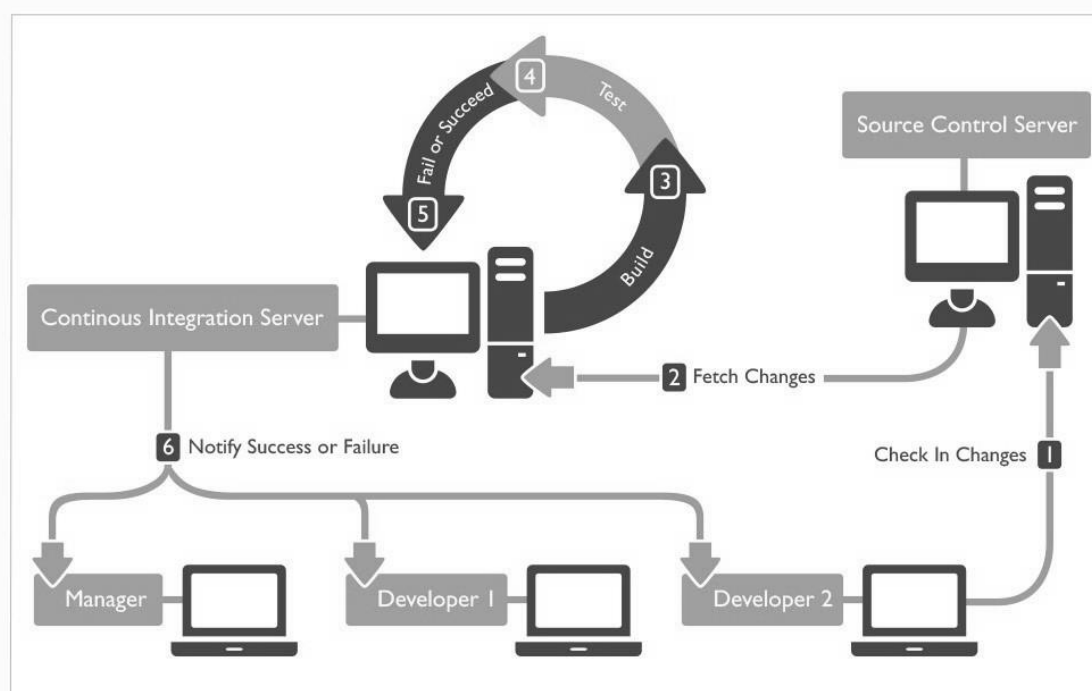


Figura 9. Fluxo de desenvolvimento executando os testes de regressão durante a construção do build de um incremento de software

Além dos testes anteriores, também podem ser realizados nessa etapa testes de vulnerabilidade de segurança e de desempenho caso estejam previstos no Plano de Teste. Esses testes normalmente são executados preferencialmente em ambiente de Produção ou ambiente similar para que as métricas obtidas reflitam os valores do sistema real.

Orientações gerais para o Time Scrum:

- É necessário controlar a concorrência dos recursos do Time de Desenvolvimento com projetos e atividades diferentes;
- As Histórias prontas e seus critérios de aceitação deverão incluir todos os artefatos de entrega para a *Sprint*, tais como: Modelo de Dados e *Scripts*, Histórias de Usuário, Código-fonte e Testes Unitários, Relatório de Cobertura de Testes e Evidências de Testes;
- É importante distinguir as pessoas que atuam como Testadores e Analistas de QA dos Desenvolvedores, já que, os objetivos dos testes são conflitantes com o desenvolvimento e a não separação desses papéis pode fazer com que os testes não obtenham os resultados esperados.

Orientações sobre as atividades que envolvem o Time de Desenvolvimento

Durante a execução da *Sprint* deverão ser observadas as seguintes orientações:

- Utilizar a solução de autenticação e autorização para sistemas da PBH - Acesso PBH [ACESSOPBH];
- Utilizar o Sistema de Arquivos da Prefeitura de Belo Horizonte - SA-PBH, caso o sistema utilize persistência de arquivos, conforme indicado no Wiki do Portal do PSP em [PSPPBH];
- Seguir o “Padrão de envio de e-mails” indicado no Wiki do Portal do PSP em [PSPPBH];
- Seguir as melhores práticas de arquitetura e engenharia de *software* do mercado em conformidade com os padrões estabelecidos pela Prodabel;
- Seguir o Fluxo de Desenvolvimento da EIC [EICV2];
- Criar e executar os testes de acordo com o Plano de Testes da *Sprint*;
- Seguir os padrões de modelagem de dados indicados pela Prodabel e utilizar a Ferramenta de Verificação de Modelos de Dados - PAD [FVMDPAD] para validar os modelos de dados construídos durante as *Sprints*;
- Desenvolver o projeto em conformidade com os artefatos definidos como padrão pela Prodabel no Portal do PSP [PSPPBH] e manter a documentação do projeto atualizada no repositório de controle de versão e no SIS;
- Configurar e manter os Jobs e *Pipelines* de integração contínua do projeto;
- Documentar as APIs de Serviços dos sistemas utilizando o *Swagger* e disponibilizar a documentação no visualizador de documentação da Prodabel [VISSWG];
- Seguir os padrões de interface para aplicações web definidos no PiWeb [PIWEB];
- Seguir as orientações do Processo de Desenvolvimento Seguro da Prodabel/PBH disponível na seção Wiki do Portal PSP: [PSPPBH];
- Seguir as orientações da Metodologia de Adequação à LGPD [MALGPD20];
- Utilizar os padrões de tecnologia de Ambiente/Infraestrutura, Ferramentas, Padrões e Processos, e Linguagens e *Frameworks* recomendados no Catálogo de Tecnologia da PBH/Prodabel [CTGTEC].

Orientações sobre as *Sprints*:

Sobre o cancelamento:

- O poder de cancelar uma *Sprint* é do PO;

- Geralmente a *Sprint* deve ser cancelada se ela não faz mais sentido para o negócio ou porque foi desrespeitado critérios como prazo ou quantidade de entregas previstas.

Sobre os Critérios de Aceitação:

- **Repositório do Projeto:** Todos os artefatos do projeto previstos no PSP, incluindo o quadro *Kanban* de gestão de tarefas do time de desenvolvimento e o código fonte, devem ser construídos obrigatoriamente através do uso da Ferramenta de Gestão de Repositório de artefatos e código Fonte, *Kanban* e execução de processos de CI/CD dos projetos de *software* da PBH [GITPBH].
- **Publicações de versões de *software*:** As publicações de versões de *software* devem seguir as definições incluídas na seção 5.4 Atividades de Gestão de Ambientes de TI;
- **Testes Unitários:** A Prodabel poderá exigir um percentual de cobertura mínima dos testes unitários para o código fonte do projeto como estratégia do plano de teste. Neste caso, os testes unitários deverão ser criados pelo time de desenvolvimento e integrados ao *pipeline* de integração contínua do projeto onde será verificado o cumprimento da métrica estabelecida de cobertura durante a execução.
- **Qualidade do Código Fonte:** A análise estática de código fonte ocorrerá na execução do *pipeline* da aplicação a cada *commit* em qualquer *branch* do repositório. Ela contemplará os códigos desenvolvidos em CSS, HTML, Javascript, Typescript, Java e PHP, podendo ser incluídas novas tecnologias futuramente pela Prodabel. Durante a realização do *merge request* para a *branch main*, o representante do time de desenvolvimento com o perfil *maintainer* do repositório na ferramenta de controle de versão da Prodabel [GITPBH] deverá avaliar o pipeline quando houver problemas a serem resolvidos, sejam eles falhas de análise estática, testes, empacotamento ou *build* da aplicação. O time de desenvolvimento deverá utilizar o profile de análise estática de código fonte definido pela Prodabel e vigente no momento do desenvolvimento do projeto. Além disso, ele será responsável por resolver todas as falhas indicadas no *status* do *Quality Gate* do projeto na ferramenta de análise estática e qualidade de código fonte e no *pipeline* de integração contínua. A comprovação da saúde do código fonte se dará através da análise dos relatórios disponibilizados pela ferramenta de análise estática e de qualidade de código fonte da Prodabel e da avaliação do histórico de execução dos *pipelines* de integração contínua pela SAS nas *Releases* dos projetos. Além disso, poderá ser feita a inspeção e auditoria de código fonte a critério da SAS.
- **Padrões de interface de aplicações web:** A verificação de interface será feita de forma paralela às demais verificações e homologações pelo cliente. Será verificado pela SAS se os sistemas atendem aos padrões do PIWeb [PIWEB].

- **Health Check de containers:** Não deve haver a ocorrência de *warnings* ou erros de *health check* para os *containers* dos sistemas mantidos na EIC.
- **Consumo de bateria em aplicativos móveis:** em projetos que envolvem o desenvolvimento de aplicativos móveis deverão ser realizados testes de consumo de bateria para confirmar a autonomia necessária aos requisitos definidos.
- **Desempenho e Segurança:**
 - Os testes de desempenho e de segurança devem abordar somente as funcionalidades consideradas mais críticas nos sistemas e as execuções desses testes devem ser acordadas com as partes envolvidas nos projetos, tais como, Time de Desenvolvimento, PO, Gerente de Projeto, envolvidos na homologação do produto, etc;
 - Os testes de segurança devem ser feitos através das ferramentas indicadas no Catálogo de Tecnologia [CTGTEC], nos ambientes de Homologação e Produção, e os relatórios devem ser submetidos para avaliação da SAS e SMI. Os sistemas não podem possuir vulnerabilidades do tipo *High*;
 - O teste de desempenho deve ser elaborado e executado através da ferramenta indicada no Catálogo de Tecnologia [CTGTEC], para este fim e deve atender aos critérios definidos nos artefatos de Plano de Teste e Especificação de Teste de Desempenho.
- **LGPD:** as exigências da LGPD previstas na Metodologia de Adequação à LGPD [MALGPD20] devem ser cumpridas.

Responsável:

- *Scrum Master*;
- Time de Desenvolvimento;
- PO.

Entradas:

- Plano da *Sprint*;
- *Backlog* da *Sprint*;
- Plano de Teste aprovado pela Equipe Técnica da Prodabel (Arquitetura de Sistemas);

- Modelo de Arquitetura Aprovado pela Equipe Técnica da Prodabel (Arquitetura de Sistemas);
- Histórias de Usuário preparadas e refinadas com os Critérios de Aceitação definidos.

Saídas:

- Incremento de *Software* das Histórias de Usuários concluídas (código fonte, *scripts* de banco de dados, etc);
- *Scripts* de testes automatizados integrados aos processos de integração contínua (testes de unidade e componentes);
- Especificação de Teste de Desempenho aprovada;
- Scripts e Relatório do Teste do Desempenho executado;
- Relatório do Teste de Análise de Vulnerabilidade e Segurança (Obrigatório);
- Modelo de Dados com relatório de verificação de padrões aprovado (Obrigatório);
- Modelo de Arquitetura Atualizado (Obrigatório);
- Relatório Análise Estática de Código (Obrigatório);
- Relatório de Resultado dos Testes Manuais com as evidências dos testes;
- Relatório de Resultado e Análise de Cobertura dos Testes Automatizados.

5.2.2.3. Refinamento do *Backlog*

Atividade: **Refinamento do *Backlog* (*Backlog Grooming*)**

Objetivo: Refinar e aprimorar as Histórias de Usuário.

Descrição: A reunião de Refinamento (*Grooming*) é uma atividade de refinamento das Histórias de Usuário, na qual o Time de Desenvolvimento se dedica em descobrir mais informações sobre um caso para poder trabalhá-lo melhor. Pode ser abordado qualquer tipo de deficiência no entendimento da equipe referente ao negócio ou técnico. O caso é detalhado e, muitas vezes, subdividido em subcasos detalhados para um melhor entendimento do que deve ser feito. Este é um processo contínuo no qual o PO e o Time de Desenvolvimento colaboram nos detalhes dos itens do *Backlog* do Produto. Durante o refinamento do *Backlog* do Produto, os itens são inspecionados e revisados. O Time de *Scrum* decide como e quando o refinamento está “Pronto”. Este refinamento usualmente não consome mais de 10% da capacidade do Time de

Desenvolvimento. Contudo, os itens do *Backlog* do Produto podem ser atualizados a qualquer momento pelo PO ou a critério do PO.

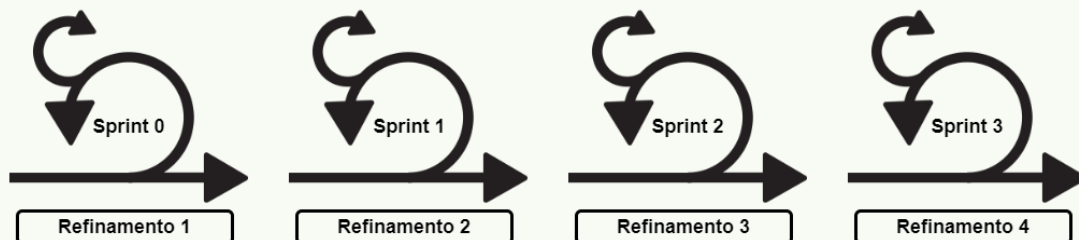


Figura 10. Ocorrência das reuniões de Refinamento do Backlog durante as releases.

O Refinamento do *Backlog* é uma atividade que deve ser realizada durante toda a *Sprint*, como mostrado na Figura 10, garantindo assim, que o *Backlog* do Produto esteja sempre pronto para a reunião de Planejamento da próxima *Sprint*.

Durante uma reunião de Refinamento do *Backlog*, o Time de Desenvolvimento tem a chance de perguntar ao PO dúvidas como:

- O que deve acontecer se o usuário inserir dados errados aqui?
- Todo usuário vai ter permissão para acessar essa parte do sistema?
- O que acontece se...?

Além da chance para a equipe fazer perguntas, a reunião de Refinamento do *Backlog* também envolve:

- A descoberta de novos itens;
- A alteração de itens existentes;
- A remoção de itens antigos ou desnecessários;
- A divisão de Histórias muito grandes em épicos ou temas;
- A priorização dos itens do *Backlog* do Produto;
- Preparar e refinar os itens mais importantes para a próxima reunião de planejamento;
- Estimar e corrigir estimativas dos itens do *Backlog* do Produto (em caso de novas descobertas);
- Incluir critérios de aceitação;

E tudo mais que envolve organização, ordenação e limpeza.

Responsável:

- PO;
- *Scrum Master*;
- Time de Desenvolvimento.

Entradas:

- *Backlog* do Produto com as Histórias de Usuário iniciais

Saída:

- *Backlog* do Produto com as Histórias de Usuário refinadas e detalhadas.

5.2.2.4. Reunião Diária

Atividade: Reunião Diária do *Scrum* (Daily *Scrum*)

Objetivo: Disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho a ser realizado no dia que se inicia.

Descrição:

Prevista no Guia *Scrum* [SCSU13]. Ela é uma ferramenta essencial para qualquer projeto em *Scrum*.

As Reuniões Diárias do *Scrum* normalmente são realizadas no mesmo lugar, na mesma hora do dia. Idealmente são realizados na parte da manhã, para ajudar a estabelecer as prioridades do novo dia de trabalho e também pode ser feito de maneira remota.

Todos os membros do Time *Scrum* podem participar das Reuniões Diárias do *Scrum*. Outras pessoas, como por exemplo os membros da Equipe Técnica da Prodabel, também podem estar presentes, mas só poderão escutar, ou seja, é uma reunião interna do Time de Desenvolvimento. Isso torna as Reuniões Diárias do *Scrum* uma excelente forma para uma equipe disseminar informações sobre o estado do projeto.

As Reuniões Diárias do *Scrum* não devem ser usadas como uma reunião para resolução de problemas. Questões levantadas devem ser levadas para fora da reunião e normalmente tratadas por um grupo menor de pessoas que tenham a ver diretamente com o problema ou possam contribuir para solucioná-lo.

Abaixo são citadas algumas das características específicas desse tipo de reunião:

- Utilizar a ferramenta de gestão de *Kanban* indicada pela Prodabel [GITPBH] para o controle das tarefas do *Backlog* da *Sprint*, através do quadro *Scrum*;
- *Time-box* - duração máxima de 15 minutos e não pode ultrapassar o tempo estipulado;
- Todos os membros do *Time Scrum* devem estar presentes;
- *Scrum Master* deve ser o facilitador da reunião.

Benefícios:

Ao manter o contato diário entre os integrantes da equipe a prática da reunião diária traz vários benefícios para o projeto e para a equipe, tais como:

- *Time Scrum* motivado;
- Novos riscos identificados;
- Riscos são mitigados.

Durante as Reuniões Diárias do *Scrum*, cada membro da equipe provê respostas para cada uma destas três perguntas fundamentais:

- O que eu fiz ontem?
- O que eu farei hoje?
- Quais obstáculos estou enfrentando?

Ao serem respondidas por todos os membros do Time de Desenvolvimento, o trabalho que vem sendo feito é revisado e ao mesmo tempo compartilhado entre os membros do time.

Responsável:

- *Scrum Master*;
- Time de Desenvolvimento;
- *Stakeholders*, quando necessário;

- Gerente de Projeto.

Entradas:

- *Backlog da Sprint.*

Saída:

- Inclusão das tarefas do *Backlog da Sprint* no quadro *Kanban* utilizando a ferramenta indicada pela Prodabel [GITPBH];
- Atualização do Gráfico de *Burndown*.

5.2.2.5. Revisão da *Sprint*

Atividade: Revisão da *Sprint* (*Sprint Review*).

Objetivo: Reunião para apresentar requisitos e funcionalidades desenvolvidas durante a execução da *Sprint*.

Descrição:

A Revisão da *Sprint* é realizada no final da *Sprint* para inspecionar o incremento e adaptar o *Backlog* do Produto, se necessário. Durante a Revisão da *Sprint* o Time *Scrum* e as partes interessadas apresentam o que foi feito na *Sprint*.

Esta é uma reunião de no máximo 4 horas de duração para uma *Sprint* de um mês. O *Scrum Master* garante que o evento ocorra e que os participantes entendam o seu propósito.

A equipe demonstra o incremento de *software* pronto e potencialmente implantável, preferencialmente em ambiente de homologação, e responde a questionamentos do PO.

O PO avalia, em conjunto com os participantes, se o plano da *Sprint* foi efetivamente realizado, se os critérios de aceitação definidos foram atendidos e se a meta de negócio da *Sprint* foi cumprida.

Os participantes incluem o Time *Scrum* e os *Stakeholders* chave convidados pelo PO.

Possíveis ações decorrentes desta reunião:

- Devolver ao *Backlog* funcionalidades não terminadas (que foram previamente acordadas com o PO durante a execução) e repriorizá-las;
- Escolher não avançar mais com o projeto ou não autorizar outra *Sprint*;
- Reprovar a entrega, caso a mesma não tenha atingido os critérios de aceitação;
- Revisão da linha do tempo, orçamento, potenciais capacidades e mercado para a próxima versão esperada de funcionalidade ou de capacidade do produto.

O resultado da Revisão da *Sprint* gera um *Backlog* do Produto revisado que define os prováveis Itens de *Backlog* do Produto para a próxima *Sprint*. O *Backlog* do Produto pode também ser ajustado completamente para atender novas oportunidades.

Esta é uma reunião informal, não uma reunião de *status*, e a apresentação do incremento destina-se a motivar e obter *feedback* e promover a colaboração.

Responsáveis:

- Time de Desenvolvimento (Obrigatório);
- *Scrum Master* (Obrigatório);
- PO;
- Gerente de Projeto;
- *Stakeholders*.

Entradas:

- Plano da *Sprint*;
- *Backlog* do Produto;
- *Backlog* e Quadro *Kanban* da *Sprint*;
- Gráfico de *Burndown* da *Release* (opcional);
- Incremento de *Software* gerado na *Sprint*.

Saídas:

- *Feedback* da Demonstração do Incremento de *Software* com possibilidade de atualização do *Backlog* do Produto.

5.2.2.6. Reunião de Retrospectiva

Atividade: Reunião de Retrospectiva.

Objetivo: Nesta reunião são avaliados os pontos fortes e fracos durante a execução da *Sprint*, visando a melhoria contínua no processo de trabalho.

Descrição:

Retrospectivas são ajustes naturais em um ambiente de trabalho ágil. Ela ajuda as pessoas a melhorarem as práticas e a tratar problemas e obstáculos.

A Retrospectiva da *Sprint* ocorre depois da Revisão da *Sprint* e antes do planejamento da próxima *Sprint*. Esta é uma reunião de no máximo três horas para uma *Sprint* de um mês.

A equipe discute os pontos positivos da *Sprint* e quais problemas foram enfrentados, assim como se esses problemas foram resolvidos.

O PO e Gerente do Projeto devem participar da reunião. Após analisar e debater o trabalho realizado na *Sprint*, deverão ser registradas as lições aprendidas e as ações de melhoria para as próximas *Sprints*.

Alguns objetivos se destacam nesta atividade de melhoria contínua [SCSU13]:

- Inspeccionar como a última *Sprint* foi em relação às pessoas, aos relacionamentos, aos processos e às ferramentas;
- Identificar e ordenar os principais itens que foram bem e as potenciais melhorias;
- Criar um plano para implementar melhorias no modo que o Time *Scrum* faz seu trabalho;
- Finalmente, a melhoria contínua do processo é importante para aumentar a qualidade da execução e, conseqüentemente, a garantia da qualidade do produto entregue.

Recomendação:

- Ao final da Retrospectiva da *Sprint*, o Time *Scrum* deverá ter identificado melhorias que serão implementadas na próxima *Sprint*. A implementação destas melhorias na próxima *Sprint* é a forma de adaptação à inspeção que o Time *Scrum* faz a si próprio.

Responsáveis:

- *Scrum Master*;
- Time de Desenvolvimento;
- Gerente de Projeto;
- PO.

Entradas:

- Ações de melhoria das Iterações anteriores;
- Relatório de desvios e impedimentos, dificuldades ou problemas encontrados.

Saídas:

- Lições aprendidas do projeto registradas junto com a documentação do projeto no repositório de controle de versão;
- Ações de melhoria registradas junto com a documentação do projeto no repositório de controle de versão.

5.3. Atividades de Transição do Projeto

O grupo de atividades de Transição do Projeto compõe as atividades de disponibilização da *Release* (Figura 11) nos ambientes de Homologação e/ou Produção. Essas atividades são responsáveis por garantir uma versão útil do produto ou parte dele nesses ambientes, além de a capacitação dos envolvidos em treinamentos e no processo de sustentação. Os métodos de publicação das *Releases* são descritos na seção 5.4 em Atividades de Gestão de Ambientes de TI.

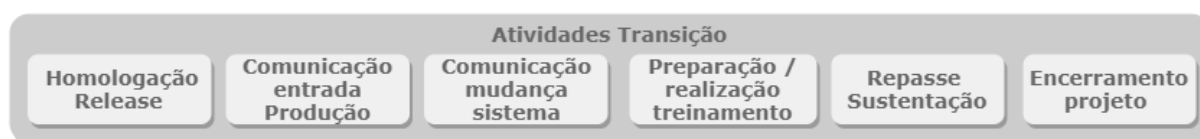


Figura 11. Detalhamento das atividades de transição

5.3.1. Validar / Homologar a *Release*

Atividade: Validar / Homologar a *Release*.

Objetivo: A Validação / Homologação da *Release* é a ratificação dada pelo PO e clientes à integração entre todos os módulos, componentes e incrementos de *software* entregues pelo Time de Desenvolvimento. Os incrementos de *software* produzidos serão validados continuamente através da realização de Testes de Regressão Automatizados executados no *pipeline* de Integração Contínua. Também serão executados Testes de Aceitação do Sistema ou de Usuário pelo PO. Esse último teste é importante para que se produzam efeitos administrativos, como os de faturamento.

Importante: A validação do incremento de *software* da *Sprint* recém-finalizada (N) deve ocorrer ao final da *Sprint*. Essa atividade deve ter um tempo fixo de 3 dias após a reunião de Revisão da *Sprint*.

A referência [FDTESTAGL] pode ser consultada para obtenção de mais informações sobre Testes de Aceite.

Descrição:

A homologação da *Release* corresponde à aceitação do produto de *software* em termos funcionais e negociais, ou seja, o cliente aceita que todas as funcionalidades estão alinhadas com os objetivos de negócio, funções de negócio e com os critérios de aceitação. É importante que a homologação aconteça sempre no ambiente de homologação, ambiente controlado não suscetível a alterações pelo Time de Desenvolvimento. Para mais detalhes sobre os ambientes veja o item 5.4. Atividades de Gestão de Ambientes de TI deste documento.

Caso sejam encontradas não conformidades, estas devem ser registradas e o produto não deve ser homologado pelo PO. O PO valida, registra e encaminha a lista de não conformidades para o Time de Desenvolvimento.

Casos de demora de realização de homologação pelo PO e usuários, após o período definido, podem ser tratados com regras específicas para não atrapalhar o cronograma de *Sprints* ou *Releases*.

Com a realização das entregas previstas pelo Fornecedor ou Equipe Interna e a homologação e aprovação sendo concretizada em conjunto pelo PO e a Prodabel, poderá ser emitido o Termo

de Aceite da entrega. Em projetos desenvolvidos por Fornecedor, antes da assinatura do Termo de Aceite pelo Cliente, será feita contagem detalhada dos Pontos de Função para a determinação do valor pago conforme descrito na seção 5.2.2. Atividades de Construção das *Sprints*. Portanto, o *Software* funcional será a medida primária de progresso.

Responsável:

- PO.

Entradas:

- Definições de Pronto da *Release*;
- *Release do Software*.

Saídas:

- Relatório de Homologação;
- Termo de Aceite da entrega.

5.3.2. Preparar e Realizar Treinamentos

Atividade: Preparar e Realizar Treinamentos.

Objetivo: Esta atividade compreende o planejamento, preparação do material e repasse de todo o conhecimento do sistema ou *release* desenvolvido para as equipes de suporte, sustentação, *stakeholders* e usuários. Esta atividade pode envolver toda a equipe do projeto.

Orientações para o treinamento e repasses:

- O PO e/ou Gerente do Projeto deve planejar a estratégia de transferência de conhecimento para os *stakeholders* e usuários do *software*;
- O Time de Desenvolvimento deve produzir o manual de usuário e o material de treinamento, quando necessário;
- Devem ser repassados, pelo Time de Desenvolvimento, todos os documentos necessários ao suporte e manutenção da solução.

Responsáveis:

- PO;
- Gerente de Projeto;
- Time de Desenvolvimento.

Entradas:

- Versão Funcional do Produto.

Saída:

- Usuários e *stakeholders* do sistema capacitados para o uso;
- Relatórios e avaliações de capacitação dos usuários;
- Manuais, apresentações, vídeos de Treinamento e outras documentações sobre o uso e a operação do Sistema.

5.3.3. Comunicar a entrada de um sistema em Produção ou em *Staging*

Atividade: Comunicar a entrada de um sistema em Produção ou em *Staging*

Objetivo: Formalizar a entrada de um sistema em Produção ou em *Staging* para que os processos de monitoração, comunicação de incidentes e outros procedimentos sejam configurados.

Descrição:

Para que seja possível a gestão de incidentes de indisponibilidade de um sistema é necessária a realização dos procedimentos de configuração de monitoramento. Esses procedimentos são solicitados através do registro manual de um SDM de formalização da entrada do sistema em Produção ou em *Staging*.

Responsáveis:

- Gerente de Projeto;
- Equipe Técnica da Prodabel responsável pela Gestão de RDMs e Infraestrutura de TI.

Entradas:

- *Release* funcional do Produto.

Saída:

- Ticket SDM de formalização de entrada do sistema em Produção ou em *Staging*;
- Mecanismos de monitoração e gestão de indisponibilidade configurados para o sistema.

5.3.4. Comunicar a realização de uma mudança no sistema

Atividade: Comunicar a realização de uma mudança em um sistema

Objetivo: Esta atividade compreende o registro de uma solicitação via SDM para formalizar uma mudança no ambiente de Produção ou em *Staging* uma vez que o sistema já esteja em operação.

Descrição: As publicações de *Releases* dos sistemas em ambiente de Produção ou em *Staging* ocorrem por meio de solicitações de publicações de build registradas via SDM. Em projetos que utilizarem a EIC como método de publicação essas solicitações são criadas automaticamente durante a execução dos *jobs* do *pipeline* de Integração Contínua. Para a abertura da solicitação durante a execução dos *jobs*, haverá a necessidade de informar alguns parâmetros, conforme descrito em [EICV2].

As informações técnicas relacionadas às solicitações de mudanças em sistemas podem ser consultadas na documentação da EIC em [EICV2].

Responsáveis:

- PO;
- Gerente de Projeto;
- Equipe Técnica da Prodabel responsável pela Gestão de RDMS e Infraestrutura de TI.

Entradas:

- *Release* funcional do Produto.

Saída:

- Ticket SDM de publicação de mudança de sistema;

5.3.5. Realizar o repasse para a Sustentação

Atividade: Realizar o repasse para a Sustentação

Objetivo: Esta atividade compreende o planejamento, preparação do material e repasse de todo o conhecimento do sistema ou *Release* desenvolvida para as equipes de suporte e sustentação. Esta atividade pode envolver toda a equipe do projeto.

Descrição: No repasse para sustentação, tanto a parte funcional quanto a não funcional do produto desenvolvido (arquitetura, código fonte, entre outros) devem ser repassados/explicados para equipe da Superintendência de Sustentação de Sistemas (SSS). O repasse irá possibilitar a continuidade nas evoluções e o início da sustentação dos sistemas desenvolvidos. Caso o sistema tenha sido desenvolvido pela mesma equipe que irá sustentar o sistema, esta etapa não é necessária.

Entre as atividades do repasse estão:

- Treinamento para a Equipe Técnica da Prodabel de Sustentação;
- Esclarecimento de dúvidas;
- Detalhamento da Arquitetura;
- Reuniões técnicas de alinhamento;
- Disponibilização das versões finalizadas dos artefatos de *software*.

Orientações para o treinamento e repasses:

- O Gerente do Projeto, junto com o Time de Desenvolvimento deve planejar a estratégia de transferência de conhecimento para equipes de suporte e sustentação;
- Devem ser repassados, pelo Time de Desenvolvimento, todos os documentos necessários ao suporte e manutenção da solução;
- Verificar que toda documentação definida como obrigatória para o projeto está completa e disponível;

- Garantir que o sistema tenha atingido os Critérios de Aceitação definidos na seção 5.2.2 Atividades de Construção das *Sprint*.

Responsáveis:

- PO;
- Gerente de Projeto;
- Time de Desenvolvimento;
- Equipe Técnica da Prodabel de Sustentação.

Entradas:

- *Release* funcional do Produto.

Saída:

- Documentações do Sistema;
- Relatório de Treinamento.

5.3.5. Encerrar o Projeto

Atividade: Reunião de Encerramento do Projeto.

Objetivo: Realizar verificações para garantir que o projeto produzido está apto para ser utilizado em produção e que o *Release* ou projeto pode ser encerrado.

Orientações para esta atividade:

- Analisar e definir ações para o tratamento dos defeitos residuais do *Release*/projeto;
- Confirmar que a solução está operacional e foi transferida para a equipe de suporte e sustentação, verificando que:
 - Foi desenvolvido todo o material de suporte (manuais, vídeos e *scripts* de atendimento);

- Equipes de suporte (1º, 2º e 3º nível) estão preparadas para dar o suporte necessário para a continuidade da solução;
- Confirmar se os usuários foram treinados e estão aptos a utilizar a solução;
- Avaliar os critérios de aceitação da qualidade do produto;
- Aprovar os resultados do *Release* / projeto;
- Caso seja o último *Release* do projeto, aprovar o encerramento do projeto.

O Gerente do Projeto deve tomar algumas ações para o encerramento efetivo do projeto, tais como:

- Atualizar o *status* no portfólio de projetos como encerrado, caso não haja mais *Releases* do projeto;
- Verificar a inclusão do novo sistema no SIS [SISPBH]: isto facilita o gerenciamento dos serviços, tanto para a TI como para a área de negócio no momento de solicitar a evolução ou correção do sistema.

Responsáveis:

- Gerente de Projeto.

Entradas:

- *Roadmap* do Produto;
- Planos das *Releases*;
- Relatório com Lições Aprendidas;
- Relatório de Treinamentos;
- Relatório de Homologação dos *Releases*.

Saída:

- Termo de Encerramento de Projeto.

5.4. Atividades de Gestão de Ambientes de TI

Os ambientes de TI são os locais em que os sistemas e seus dados serão publicados, executados, mantidos e armazenados durante o ciclo de desenvolvimento e manutenção de *software* em projetos e sistemas da Prodabel / PBH. Por padrão, eles são organizados da seguinte forma:

- **Ambiente de Desenvolvimento:** ambiente mantido em infraestrutura própria do Time de Desenvolvimento (ex: máquina local do próprio desenvolvedor);
- **Ambiente de QA (*Quality Assurance*):** ambiente mantido pela infraestrutura da Prodabel e utilizado para realização de validações de integração de código fonte desenvolvido pelo time de desenvolvimento (Testadores e Analistas de QA) durante a execução das *Sprints*;
- **Ambiente de Homologação:** ambiente mantido pela infraestrutura da Prodabel e utilizado para a validação das *Releases* pelo PO e realização de Testes de Desempenho;
- **Ambiente de Staging:** ambiente mantido pela infraestrutura da Prodabel que possui características similares ao de Produção, utilizado em cenários específicos para sistemas corporativos em que há a exigência de um ambiente estável para realização de integrações com outros sistemas. A utilização deste ambiente é opcional e sua adoção deve ser avaliada com a equipe de Arquitetura de Sistemas da Prodabel;
- **Ambiente de Produção:** ambiente mantido pela infraestrutura da Prodabel e disponibilizado para o uso pelo Usuário final do sistema.

*Outros ambientes, como ambientes de treinamento por exemplo, poderão ser criados sob avaliação da equipe da Arquitetura de Sistemas da Prodabel.

Para a publicação dos sistemas nesses ambientes a Prodabel recomenda o uso da Esteira de Integração Contínua – EIC [EICV2], que é o método mais moderno e eficiente existente atualmente para publicação e manutenção de sistemas na infraestrutura de TI do Datacenter.

Em situações específicas, que devem passar por análise pelas equipes de Arquitetura de Sistemas e de Infraestrutura da Prodabel, alguns projetos e sistemas poderão ser publicados através do MLFDC (Modelo Lógico Funcional do Datacenter) por meio de um Plano de

Implantação, que é um documento que descreve os passos necessários para a configuração e instalação manual dos ambientes. As informações sobre esta última forma de publicação estarão disponíveis na página de Wiki no Portal do PSP [PSPPBH].

A *Figura 12* resume os grupos de atividades que estão relacionados nas Atividades de Gestão de Ambientes de TI.



Figura 12. Grupos de Atividades de Gestão de Ambientes de TI

5.4.1. Publicar *software* com a utilização da EIC

Atividade: Publicação de ambientes com a utilização de Integração e Entrega Contínua.

Objetivo: Manter os ambientes dos projetos utilizando a infraestrutura e as ferramentas da Esteira de Integração Contínua - EIC v2.

Descrição: A EIC é um método de publicação de *software* que aplica automação contínua em todo o ciclo de desenvolvimento, incluindo as etapas de teste e integração, além da entrega e implantação. Esse método é fundamentado pelos conceitos de Integração Contínua (CI), Entrega Contínua e da tecnologia de *containers* [EICV2].

A arquitetura base da EIC é formada por um conjunto de ferramentas e processos que se dividem em três módulos: Integração Contínua, Entrega Contínua e Ambientes operacionais conforme a *Figura 13*.



Figura 13 – Módulos da EIC e suas responsabilidades [EICV2]

Integração Contínua

É o módulo em que estarão os processos e ferramentas responsáveis pela execução dos testes automatizados, análise estática de código fonte e o empacotamento e implantação dos incrementos de *software* produzidos pelo time de desenvolvimento.

Entrega Contínua

Este módulo contém as ferramentas e processos responsáveis por fazer a ligação entre a integração contínua e os ambientes operacionais, aplicando as configurações de ambiente definidas e operacionalizando as atualizações de versão.

Disponibilização dos ambientes para o time de desenvolvimento

A disponibilização dos ambientes da EIC para o time de desenvolvimento ocorre mediante solicitação para o time de Arquitetura de Sistemas / *DevOps* da Prodabel durante a Sprint 0, conforme descrito no item 5.2.1.2 Sprint 0.

Outras informações sobre a manutenção dos ambientes na EIC e sobre os processos envolvidos podem ser obtidas na Documentação Oficial da EIC [EICV2] e no Portal do PSP [PSPPBH].

Responsável:

- Equipe Técnica da Prodabel de Arquitetura de Sistemas;

- Time de Desenvolvimento

Entradas:

- Pré-requisitos definidos para *Sprint 0* para criação de ambientes conforme a seção 5.2.1.2 *Sprint 0*.

Saídas:

- Ambientes disponíveis de acordo com o solicitado na *Sprint 0*.

5.4.1. Publicar Bancos de Dados

Atividade: Criação e manutenção de objetos de Bancos de Dados

Objetivo: Criação e manutenção de objetos de bancos de dados nos ambientes das aplicações

Descrição:

Em projetos da EIC, a criação e alteração de objetos de banco de dados, tais como, tabelas, triggers, procedures, bem como as cargas necessárias para o funcionamento do sistema, são mantidas através da utilização do *Liquibase*, que é uma ferramenta de gestão e versionamento de alterações em schemas de bancos de dados dos sistemas. O processo para utilização do *Liquibase* está disponível na seção sobre Ferramentas na Documentação da EIC [EICV2].

Em situações especiais, que devem ser acordadas antes com a equipe de Arquitetura de Sistemas, alguns sistemas poderão realizar a alteração de objetos de banco de dados através de uma solicitação para a Equipe Técnica da Prodabel de Banco de Dados.

Em casos em que o sistema necessite de acesso aos bancos de dados de Homologação e / ou Produção é necessário haver um pedido de autorização para a Equipe Técnica da Prodabel de Banco de Dados por meio “Termo de Autorização de Acesso a Banco de Dados” disponível no Portal do PSP em [PSPPBH].

Responsável:

- Time de Desenvolvimento;
- Equipe Técnica da Prodabel de Banco de Dados.

Entradas:

- Processo via *Liquibase*: *Tag* criada no repositório de código fonte do projeto com os *scripts* de banco de dados para carga de dados ou criação/modificação/exclusão de objetos de banco de dados configurados no *Liquibase*;
- Processo via solicitação para a Equipe Técnica de Banco de Dados da Prodabel: solicitação com os *scripts* para a execução no Banco de Dados.

Saídas:

- Logs dos *scripts* executados nos ambientes de Banco de Dados.

5.5. Atividades de Acompanhamento de Projeto

As atividades de acompanhamento do projeto devem ser realizadas durante todo o projeto. Alguns artefatos de controle e acompanhamento podem ser preenchidos com o apoio do Time de Desenvolvimento de *software*, como por exemplo Gráficos de *Burndown*. Já outros serão de responsabilidade exclusiva do Gerente do Projeto, como por exemplo, o Plano de Projetos.

As atividades de acompanhamento do projeto estão representadas na *Figura 14*:

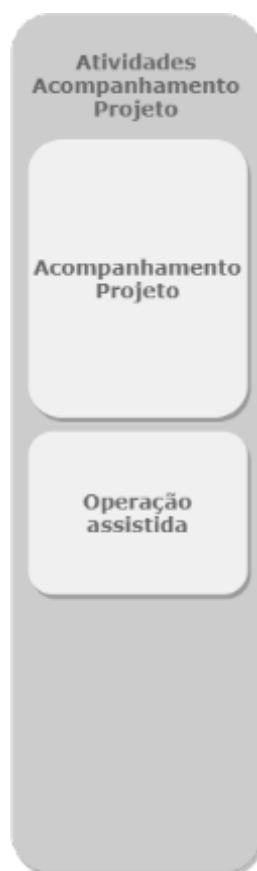


Figura 14 – Atividades de acompanhamento de projeto

5.5.1. Acompanhar Projeto

Atividade: Acompanhamento do Projeto

Objetivo: Realizar esta atividade significa acompanhar cronograma e riscos (desvios e impedimentos), atualizar os artefatos e comunicar informações importantes sobre o andamento do projeto.

Descrição: O acompanhamento do projeto (*Figura 13*) é realizado pelo Gerente de Projeto. Caso seja necessário ele deve marcar reuniões periódicas com o PO e a equipe para avaliar o andamento do projeto.

O projeto deve ser acompanhado periodicamente para verificar se seu andamento está de acordo com os prazos definidos (OS). E caso seja identificado problemas ou desvios, providências devem ser tomadas o mais cedo possível, para evitar impactos nas entregas do projeto.

Toda vez que o *Time Scrum*, em suas reuniões (Reunião Diária, Revisão da Sprint e Retrospectiva) ou execução de tarefas, deparar com desvios e impedimentos técnicos do projeto de responsabilidade da Prodabel, que impactam o andamento do projeto, o PO e o Gerente de Projetos devem ser informados. O PO deve buscar, com o apoio do Gerente do Projeto, solucionar esses problemas junto aos responsáveis. Quanto mais rápido as providências forem tomadas, menor será o impacto no projeto.

Recomenda-se que os projetos de *software* executados pela TI sejam cadastrados no portfólio de projetos e atualizados periodicamente.

Informações como os *status* de cada projeto (andamento, suspensos, cancelados), prazos (se atrasados ou não) e impedimentos devem ser atualizados frequentemente. Essas informações são atualizadas no portfólio de projetos, pelo Gerente do Projeto e utilizadas para informar os envolvidos no projeto.

Pode ser acordado entre o Gerente de Projeto e o *Time Scrum*, a realização de reuniões de ponto de controle para acompanhamento do andamento da *Sprint* em periodicidade previamente negociada.

Um Gráfico de *Burndown* pode ser gerado para o projeto para avaliação periódica de informações sobre o andamento do projeto, conforme acordado com o *Time Scrum*.

Responsáveis:

- Gerente de Projeto.

Entradas:

- Ordens de Serviço;
- Documento de Visão;
- *Roadmap* do Produto;
- Plano da *Release*;
- Plano da *Sprint*;
- *Backlog* do produto;
- Gráficos de *Burndown*;
- Relatos do acompanhamento com a equipe.

Saída:

- Plano de Projeto Atualizado;
- Gráficos de *Burndown* atualizado;
- Portifólio de Projetos Atualizado (*status* do projeto).

5.5.2. Operação Assistida

Atividade: Operação Assistida

Objetivo: Acompanhar projetos recém entregues a fim de detectar desvios e facilitar a correção de bugs

Descrição: A fim de facilitar a resolução de *bugs* e o acionamento da equipe técnica após a implantação de um sistema/projeto, o escritório de projetos irá acompanhar a operação do sistema em produção no período de trinta dias úteis. Caso um problema seja encontrado, este deve ser relatado pelo cliente na Ferramenta de Gestão de Bugs da Prodabel [FGBUG] e o Gerente de Projetos de articular a resolução deste *bug* junto com o time de desenvolvimento.

Responsáveis:

- Gerente de Projeto;
- Cliente.

Entradas:

- Sistema em produção.

Saída:

- Problemas relatados na Ferramenta de Gestão de Bugs da Prodabel [FGBUG]

No Portal do PSP [PSPPBH] é possível encontrar outras informações de apoio para o desenvolvimento de *software* na PBH / Prodabel.

6. REFERÊNCIAS BIBLIOGRÁFICAS

[AAES] *Subway Map to Agile Practices*, *Agile Alliance: Agile Essentials*. Disponível em: <<https://www.agilealliance.org/agile101/subway-map-to-agile-practices/>>. Acesso em 17/07/2020.

[ACESSOPBH] Solução de autenticação e autorização para os sistemas da Prefeitura de Belo Horizonte. Disponível em: <<https://acesso.pbh.gov.br/auth/>>.

[AGAL15] *Agile Alliance Guide web site*. Disponível em: <<http://guide.agilealliance.org/subway.html>>. Acesso em: maio de 2015.

[AGALT] *Timebox, Agile Alliance*, Disponível em: <<http://guide.agilealliance.org/guide/timebox.html>>. Acesso em 17/07/2020.

[AM01] *Agile Manifest web site*. 2001. Disponível em: <<http://agilemanifesto.org/iso/ptbr/>>. Acesso em: junho de 2014.

[DO15] *DevOps*. Disponível em: <<http://www.scaledagileframework.com/devops/>>. Acesso em: maio de 2015.

[CTGTEC] Catálogo de Tecnologia da Prodabel/PBH. Disponível em: <<https://catalogotecnologia.pbh.gov.br/>>.

[EXPR] História de Usuário, Extreme Programming. Disponível em: <<http://www.extremeprogramming.org/rules/userstories.html>>. Acesso em 17/07/2020.

[EICV2] Documentação da Esteira de Integração Contínua da Prodabel / PBH. Disponível em: <<https://eic.pbh.gov.br>> (acessível somente via RMI).

[FDTESTAGL] Fundamentos de Teste de Software e Teste Ágil. Disponível em: <<http://psp.pbh.gov.br/qualidade-software>>.

[FGBUG] Ferramenta de Gestão de Bugs da Prodabel – Mantis Bug Tracker. Disponível em:

<<https://bugs.pbh.gov.br/>>.

[FVMDPAD] Ferramenta de Verificação de Modelo de Dados - PAD. Disponível em: <<http://pad.pbh.gov.br/>>.

[GITPBH] Ferramenta de Gestão de Repositório de artefatos e código Fonte, *Kanban* e execução de processos de CI/CD dos projetos de *software* da PBH – GitLab. Disponível em: <<https://gitlab.pbh.gov.br/>>.

[IEEE04] *Institute Of Electrical and Eletronics Engineers (IEEE). Guide to the Software Engineering Body of Knowledge (SWEBOK). 2004 Version*

[IEEE90] *Institute Of Electrical and Eletronics Engineers (IEEE). Standard Glossary of Software Engineering Terminology; IEEE Std 610.12-1990, 1990.*

[IIBA11] *International Institute. Business Analysis. Guia BABOK: Um guia para o corpo de conhecimento de Análise de Negócios Versão 2.0. Toronto: International Institute of Business Analysis, 2011.*

[MAFO09] *Technical Debt Quadrant*, Martin Fowler, 14 October 2009. Disponível em: <<http://martinfowler.com/bliki/TechnicalDebtQuadrant.html>>. Acesso em 17/07/2020.

[MAFO19] *Technical Debt*, Martin Fowler, 21 May 2019, Disponível em: <<http://martinfowler.com/bliki/TechnicalDebt.html>>. Acesso em 17/07/2020.

[MALGPD20] Metodologia de Adequação à LGPD. Disponível em: <<http://gsi.pbh/>> (acessível somente via RMI), 2020.

[PMI17] *A Guide to the Project Management Body of Knowledge*, 2017, PMI - Project Management Institute. Disponível em: <https://www.pmi.org/pmbok-guide-standards/foundational/pmbok>. Acesso em 17/07/2020.

[PGESTPBH] Portal da Gestão da PBH. Disponível em: <<https://portaldagestao.pbh.gov.br/>>.

[PSPPBH] Portal do Processo de Software da PBH. Disponível em:

<<https://psp.pbh.gov.br>>.

[PIWEB] Padrão de Interfaces de Sistemas Web – PIWeb. Disponível em: <<https://piweb.pbh.gov.br/>>.

[SAF14] *Scaled Agile Framework web site. A Proven, Publicly Available Framework for Applying Lean / Agile Practices at Enterprise Scale.* Disponível em: <<http://scaledagileframework.com/>>. Acesso em out, 2014.

[SCSU13] Schwaber, Ken; Sutherland, Jeff. *The Definitive Guide to Scrum: The Rules of the Game.* Scrum.org, 2017. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>>.

[SIM14] Simões, Roberto. Scrumex - Expandindo a Aplicação e os Benefícios do Scrum. Disponível em: <<http://www.scrumex.com.br/blog>>. Acesso em out, 2014.

[SISPBH] Catálogo de Sistemas da Prefeitura de Belo Horizonte. Disponível em: <<https://sis.pbh.gov.br/>>.

[SISP15] Guia de Projetos de Software com Práticas de Métodos Ágeis para o SISP. Disponível em: <http://www.sisp.gov.br/guiaagil/wiki/download/file/Guia_de_Projetos_Ageis_Final>. Acesso em: setembro de 2019.

[VISSWG] Visualizador de documentação de APIs Swagger. Disponível em: <<https://api-docs.pbh.gov.br/>> (acessível somente via RMI).

7. REVISÕES DO DOCUMENTO

Versão	Data da revisão	Descrição das alterações
1.2.0	08/07/2021	Inclusão da seção de revisões do documento.
1.1.0	06/07/2021	- Remoção das referências para a requisição de mudança ágil (RDM Ágil); - Revisão de texto da Sprint 0; - Revisão dos critérios de aceitação da execução da Sprint.

ANEXO I - CONCEITOS E FUNDAMENTOS SOBRE O MODELO DE REFERÊNCIA

Conceitos Gerais

- **Adaptação:** Significa que toda vez que uma variação prejudicial ao projeto é identificada, o processo deve ser ajustado imediatamente, como forma de evitar outros desvios.
- **Brainstorming:** técnica dedicada a produzir um conjunto diverso de opções para um problema ou oportunidade.
- **DevOps:** é relação de interdependência entre os desenvolvedores de software e os profissionais de tecnologia da informação que viabilizam a operação do software e necessidades afins por meio de um sistema de produção e integrado de pessoas, informação, ativos de *hardware*, serviços, ferramentas e aplicativos, que apoiam as atividades de gestão de projetos, produção e gerência de requisitos de software, construção de *software*, prototipação de interfaces gráficas de usuário, construção e disponibilização do banco de dados, testes automatizados, inspeção automatizada da qualidade dos produtos, integração contínua dos componentes de software, disponibilização automática do *release* em ambiente apropriado, gestão informatizada dos ambientes de disponibilização, gestão de configuração de artefatos de engenharia e gestão de mudanças. Todas estas atividades de engenharia de software são desenvolvidas por meio de processos, métodos e ferramentas, alguns destes são informatizados, promovendo o princípio da impessoalidade e a diminuição do esforço da execução pela automação de processos operacionais. Os profissionais especializados em *DevOps* são responsáveis pela sua instalação, configuração, disponibilização, operação, conectividade, continuidade, segurança e manutenção. O *DevOps* exige comunicação estreita entre os profissionais, colaboração contínua e cooperação máxima [SAF14], [IEEE04] [DO15];

Veja mais informações em:

- Agile Alliance Guide web [AGAL15];
- *DevOps* [DO15].
- **Cobertura de Testes:** os testes podem ser qualificados quanto a sua extensão e escopo. A cobertura é o limite de incidência do teste, podendo atingir os seguintes níveis de abstração: negócio (teste de aceitação), requisitos (teste de sistema), projeto (teste de integração) e código (testes unitários).
- **Débito técnico:** pendências técnicas introduzidas no código em virtude de ter sido implementado sem o design ou cuidados necessários. Estas pendências podem estar relacionadas com: qualidade, bom design que permita fácil adaptação, manutenção, testes etc. A existência de débito técnico indica que a tarefa não foi totalmente concluída e que a equipe deverá ter mais trabalho futuramente em virtude destas pendências deixadas no projeto.

Veja mais informações em:

- *Technical Debt* [MAFO19]
- *Technical Debt Quadrant* [MAFO09].
- **Defeito:** problema no software que faz com que ele se comporte de forma distinta da esperada/desejada.
- **Impedimento:** é todo problema que afeta o ritmo de trabalho da equipe do projeto, podendo comprometer o alcance da meta estabelecida para a *Sprint* ou *Release*.
- **Definição de Pronto:** é um entendimento compartilhado entre os envolvidos no projeto (Instituição e contratada) do que significa trabalho completo, assegurando a transparência. O “Pronto” será utilizado para definir um nível de maturidade e ou qualidade de entregáveis em cada grupo de atividades do processo. Por exemplo: artefatos de planejamento, requisitos, incremento de software (*Sprint*) e versão funcional do software (*release*).
- **Épico:** é uma necessidade de negócio não refinada, correspondendo a algum caso de negócio. O épico do produto ou do produto-parte de uma solução informatizada (de

um determinado problema) será decomposto em características macrofuncionais que o refinam. A solução deve agregar valor perceptível ao negócio do cliente [SAF14].

- **Funcionalidade:** é a menor parte e mais elementar de um sistema informatizado, que agrega algum valor ao usuário. É uma história de usuário conforme definido pelo *framework XP* e que, para o Processo Unificado, seria equivalente à decomposição de um caso de uso ou mesmo à descrição do cenário de um caso de uso;
- **História de Usuário (*user story*):** descrição curta de uma característica do produto contada na perspectiva do usuário, utilizando uma linguagem comum ao negócio.

Veja mais informações em:

- *User Stories* [EXPR].
- **Incremento de Software:** acréscimo de funcionalidades do produto de software final que é gerada a cada *Sprint*. O incremento de software deve ser algo observável e, preferencialmente funcional, para que o PO e demais usuários possam experimentá-lo da forma mais realista possível.
- **Inspeção:** Todos os aspectos do processo de entrega que possam impactar o resultado final do projeto devem ser inspecionados frequentemente, para que qualquer variação prejudicial possa ser identificada e corrigida o mais rápido possível.
- ***Sprint*:** um ciclo de desenvolvimento, de duração fixa e curta, que corresponde a uma subdivisão do *release* e que produz uma versão estável e executável do produto de software.

Veja mais informações em:

- *Iteration* [AGAL].
- **Meta:** breve descrição em função dos objetivos de negócio e prazo que a equipe pretende alcançar durante um *release* ou *Sprint*.
- **Produto:** resultado final de um projeto, cuja finalidade é responder a demandas associadas ao software, ajustes na organização, necessidade de um ou mais clientes, avanços tecnológicos ou obrigações legais. O termo produto está relacionado a um produto físico, serviço ou associação de ambos. Independente da origem da demanda, um produto sempre é destinado a um cliente ou grupo de clientes, internos ou externos [SIM14].

- **Release:** conjunto de funcionalidades extraídas do *Backlog* do produto. Representa entrega de um software funcional incluindo funcionalidades de uma ou mais iterações.
- **Requisitos Funcionais:** é a descrição do comportamento e a informação que a solução gerenciará. Descrevem capacidades que o sistema será capaz de executar em termos de comportamentos e operações – ações ou respostas específicas de aplicativos de tecnologia da informação [IIB11], [IEEE90].
- **Requisitos Não Funcionais:** são condições que não se relacionam diretamente ao comportamento ou funcionalidade da solução, mas descrevem condições ambientais sob as quais a solução deve permanecer efetiva, ou qualidades que os *softwares* precisam possuir. Também são conhecidos como requisitos de qualidade ou suplementares. Podem incluir requisitos relacionados à capacidade, velocidade, segurança, disponibilidade, arquitetura da informação e apresentação da interface com o usuário [IIBA11].
- **Refinamento dos Requisitos:** considerando os aspectos do desenvolvimento ágil, define-se refinamento dos requisitos (retrabalho) como sendo os refinamentos referentes a funcionalidades que estão em desenvolvimento dentro de um mesmo *release*, ou seja, mudanças entre as iterações do *release*.
- **Scrum:** é um *framework* de desenvolvimento ágil de software iterativo e incremental para o gerenciamento de desenvolvimento de produtos [SCSU13].

Veja mais informações em:

- *Subway Map to Agile Practices* [AAES]
- **Stakeholder:** indivíduo ou organização que tem um direito, ação, declaração ou interesse no *software* em desenvolvimento; envolvido; interessado [PMI17].
- **Time-box:** a tradução literal é “caixa de tempo” e deve ser entendido como um intervalo de tempo para realização de uma atividade. É uma técnica utilizada para assegurar que atividades importantes do projeto sejam cumpridas e com o menor desperdício possível.

Veja mais informações em:

- *Timebox* [AGALT].

- **Teste de Software:** consiste numa verificação dinâmica do comportamento de um programa em um conjunto finito de casos de teste contra o comportamento esperado. [IEEE04], [IEEE90].
- **Teste de Aceitação:** teste escrito sob a perspectiva do cliente [BECK04]. Esse tipo de teste, geralmente, especifica o resultado esperado após a execução de determinada funcionalidade [IEEE90].
- **Teste de Sistema:** o objetivo é executar as funcionalidades do sistema sob a ótica de quem utilizará, explorando as funcionalidades em busca de inconformidades em relação aos objetivos e especificações funcionais. Os Testes de Sistema são executados em condições operacionais similares, por exemplo, ambiente, interfaces sistêmicas e massas de dados, àquelas que um usuário utilizará na sua rotina de operação do *software* [IEEE90].
- **Teste de Integração:** encontrar falhas provenientes da integração interna dos componentes de um *software*, ele conduz ao descobrimento de possíveis falhas, erros e defeitos associados à interface do *software*. A critério da gerência de projetos, os testes de comunicação entre interfaces com outros *softwares* (integração entre eles) poderão ser realizados [IEEE90].
- **Teste de Unidade:** teste escrito sob a perspectiva do programador, aplicado sobre a menor unidade de execução do código. Cada caso de teste deve avaliar um possível comportamento do código [IEEE90].
- **Teste de Desempenho:** teste realizado para avaliar a adequação de um sistema ou componente com determinados requisitos de desempenho [IEEE90].

ANEXO II - PAPÉIS DA EQUIPE NO MODELO DE REFERÊNCIA

- **Analista de Infraestrutura de TI:** é o representante da área de infraestrutura para o projeto.

Responsabilidades:

- Definir e executar projetos de implantação de infraestrutura de TI;
 - Identificar e instalar infraestrutura de *hardware*, *software* e serviços para ambientes de TI (Teste, Homologação, Produção);
 - Planejar, executar, acompanhar e controlar os projetos de infraestrutura de TI, identificando riscos e impactos e garantindo o cumprimento das tarefas e prazos;
 - Desenvolver e analisar indicadores de desempenho a fim de mensurar os objetivos e desempenho da área;
 - Garantir a padronização dos processos internos, disponibilizando internamente os documentos relativos à área de infraestrutura.
- **Analista de Qualidade:** representa diferentes perfis ligados a avaliação de qualidade de *software*, tais como: Analista de Teste, Arquiteto de Softwares, Analista de Banco de Dados, Analista de Configuração e Mudança e outros.

Responsabilidades:

- Avaliar e definir processos de verificação da qualidade dos produtos entregues pelo Fornecedor (artefatos, incremento de *software* e *release*);
 - Realizar testes de sistemas.
- **Analista de Métricas:** é o responsável pelos serviços de medição de *software*.

Responsabilidades:

- Realizar medições de *software* utilizando a métrica definida no contrato com a empresa de desenvolvimento de *software*.
- **Cliente:** representante da área de negócio.

Responsabilidades:

- É responsável pela solicitação e homologação dos serviços de desenvolvimento de *software*.
- **PO:** representante da área de negócio com conhecimento suficiente para definir e priorizar requisitos do negócio e responder aos questionamentos da equipe de desenvolvimento. É o representante do cliente e sua atuação tem por finalidade garantir a entrega do valor esperado através das entregas parciais e do produto final do projeto.

Responsabilidades:

- Estabelecer a Visão do Produto;
- Aprovar o *Roadmap* de *releases* do produto;
- Gerenciar o *Backlog* do produto, fornecendo e priorizando os requisitos;
- Definir prioridades de negócio;
- Conhecer e transmitir o processo de negócio e seus objetivos;
- Detalhar os requisitos de cada funcionalidade;
- Aceitar ou rejeitar produtos e serviços;
- Gerenciar as expectativas dos *stakeholders*;
- Resolver os itens de *Backlog* não planejados;
- Aprovar mudanças no projeto;
- Preparar e realizar treinamentos;

- Dar apoio ao Time *Scrum* durante todo o projeto.

***Obs:** O PO é uma pessoa e não um comitê. Ele pode representar o desejo de um comitê no *Backlog* do Produto, mas aqueles que quiserem uma alteração nas prioridades dos itens de *Backlog* devem convencer o PO.

- **Usuário:** tipos de *stakeholder* que usa o produto de *software*. Pode ser interno ou externo à instituição.

Responsabilidades:

- Auxiliar na definição dos itens de *Backlog*;
- Auxiliar na homologação e validação das entregas do projeto.
- **Scrum Master:** é um dos representantes do Time *Scrum*. Deve trabalhar para o sucesso do projeto através da entrega de valor para o cliente, viabilizada pela atuação eficaz da equipe de desenvolvimento e apoio ao PO. O papel de *Scrum Master* é, normalmente, exercido por um líder técnico, mas é um papel que pode ser desempenhado por qualquer membro da equipe. Deve trabalhar para o sucesso do projeto através da entrega de valor para o cliente, viabilizada pela atuação eficaz da equipe de desenvolvimento e apoio ao PO.

Responsabilidades:

- Remover impedimentos da equipe de desenvolvimento (problemas técnicos, administração de conflitos, itens não planejados);
- Auxiliar o PO com técnicas para o gerenciamento efetivo do *Backlog* do produto;
- Comunicar claramente a visão, objetivo e itens do *Backlog* do produto para a equipe de desenvolvimento;
- Ser o facilitador da equipe de desenvolvimento, garantindo sua plena produtividade;
- Garantir a colaboração entre os diversos papéis e funções da equipe de desenvolvimento;

- Facilitar eventos e reuniões do *Scrum* conforme exigidos ou necessários;
- Aplicar e disseminar valores e práticas *Scrum*.
- **Time de Desenvolvimento:** é um grupo de cinco a nove integrantes do Time *Scrum*, que deve possuir característica multifuncional e auto organização, contando com analistas, arquitetos, programadores, testadores, administrador de banco de dados, entre outros. Individualmente os integrantes do Time de Desenvolvimento podem ter habilidades especializadas e área de especialização, mas a responsabilidade pertence a toda equipe de desenvolvimento.

Responsabilidades:

- Fazer as estimativas necessárias;
- Definir as tarefas que serão realizadas;
- Desenvolver o produto;
- Realizar repasses de conhecimento para as equipes de suporte e sustentação;
- Garantir a qualidade do produto;
- Apresentar o produto ao cliente.
- **Cliente**
 - Identificar e qualificar os problemas ou necessidades de melhorias/automatização em seus processos ou sistemas atuais;
 - Encaminhar suas demandas, via Gerente de Relacionamento, para apresentação nas reuniões do Comitê Gestor;
 - Participar das reuniões do Comitê Gestor, tanto ordinárias quanto extraordinárias, a fim de validar as decisões tomadas pelo grupo;
 - Estar disponível para o PO a fim de esclarecer dúvidas e impedimentos pertinentes ao projeto;
 - Validar os itens entregues no tempo pactuado entre as partes do projeto;

- Auxiliar na definição dos itens de *Backlog*;
- Auxiliar na homologação e validação das entregas do projeto.

- **Comitê de Tecnologia da Prodabel**

- O Comitê de Tecnologia da Prodabel é composto pela PREDE, DSI, DIE e DAU;
- As deliberações para aprovação de Proposições de Projetos serão feitas eletronicamente através do Portal da Gestão;
- Excepcionalmente, caso haja uma grande complexidade, importância ou necessidade de discussão, será solicitada uma reunião presencial do Comitê de Tecnologia para deliberações;
- Toda proposição de projeto, que tenha sido inicialmente aprovada como Projeto, ensejará a elaboração do EVT junto ao GR. Quando concluído, ele retornará para avaliação e deliberação do Comitê de Tecnologia.

- **Comitê Gestor**

- Comitê formalmente constituído, responsável por analisar, triar e priorizar as demandas repassadas pelos clientes, bem como por definir os rumos de evolução das soluções de Tecnologia da Informação. O Comitê Gestor será formado pelo Gerente de Relacionamento, Coordenador da CR, Cliente, podendo ser demandados outros profissionais conforme necessidade;
- Receber, analisar, realizar triagem e priorizar as demandas em função da necessidade estratégica do negócio;
- Avaliar e deliberar as solicitações de mudança que impactem nos cronogramas/prazos/orçamento previamente acordados;
- Deliberar sobre alteração na ordem de priorização das demandas que estejam em execução;
- Acompanhar periódica e metodicamente as atividades em execução pela Célula de Resultados para atendimento a uma determinada solução, através

de reuniões promovidas pelo Coordenador da Célula e/ou Gerente de Relacionamento;

- Analisar as demandas de sustentação e projeto e organizá-las em versionamentos;
- Antecipar as necessidades de evolução tecnológica e a necessidade de novos projetos.

- **Coordenador**

Responsabilidades / atribuições em uma célula de resultados:

- Profissional com domínio e conhecimentos técnicos e do negócio da área temática na qual atua e cujas habilidades facilitem a execução, coordenação e organização das atividades da equipe sob sua responsabilidade, de forma a garantir o atendimento satisfatório às demandas que lhe forem atribuídas;
- Responsável pelo bom funcionamento da Célula de Resultados;
- Garantir que a equipe cumpra com suas atividades dentro do previsto;
- Auxilia o cliente na priorização dos eventos;
- É o profissional responsável por montar os times ágeis com os profissionais que estão sob sua coordenação;
- Participa do Comitê que prioriza as ações e recebe as tarefas identificadas e priorizadas pelo Comitê Gestor e distribuí-las à equipe das Células Resultados;
- Atua como Líder técnico e/ou analista de negócio, ajudando a equipe;
- Avaliação da qualidade dos serviços da célula;
- Organizar a distribuição das tarefas à equipe com o intuito de melhorar a produtividade, remover impedimentos e alinhar as informações necessárias para o desenvolvimento do processo;
- Participar das reuniões de acompanhamento de projetos quando solicitado fornecendo informações sobre os projetos e demandas em que ele atua;

- Garantir que os processos de qualidade sejam realizados corretamente pela equipe, conforme especificado e planejado;
- Atuar como elo de comunicação interna entre todos os envolvidos nas atividades da Célula de Resultado, de forma a alinhar interesses e expectativas, visando manter cada profissional da equipe focado nas suas respectivas tarefas, para obter os melhores resultados possíveis.
- Favorecer o estreitamento da relação entre o Gerente de Relacionamento e a equipe da Célula de Resultado;
- Compartilhar a responsabilidade do projeto com a equipe envolvida;
- Favorecer a construção de uma equipe multidisciplinar e auto gerenciável;
- Gerenciar as expectativas da equipe, bem como eventuais conflitos internos e externos.

- **Gerente de Projetos**

- Profissional que detém conhecimentos da disciplina de gestão de projetos, e que se responsabiliza pelo planejamento, coordenação e execução dos projetos constantes no portfólio de projetos da PBH/PRODABEL, abrangendo as fases de planejamento, execução, monitoramento, controle, encerramento e pelo acompanhamento da execução de projetos de TI, do ponto de vista de escopo e cronograma e demais áreas de conhecimento de gestão de projetos;
- Garantir que a adesão da equipe do projeto aos métodos e processos ágeis, através de reuniões, definições de papéis e *feedbacks* constantes, sendo um facilitador para o *Scrum Master*;
- Atuar na resolução de conflitos em relação ao que é esperado pelos clientes e usuários, ajudando a definir melhor, adequar a escalada de Histórias de Usuários e assegurando a prestação de contas;
- Garantir que a equipe terá “proteção” de influências externas;
- Assumir tarefas especializadas em projetos guiados por métodos ágeis. As tarefas a serem supervisionadas pelo Gerente de Projetos em equipes ágil

podem incluir:

- Finanças do projeto;
 - Relatórios de *status*;
 - Governança do projeto;
 - Identificação de papéis em falta e/ou recursos;
 - Comunicação das partes interessadas de negócios;
 - Comunicação e gestão de riscos;
 - Planejamento do projeto;
 - Gestão da mudança.
- Deve seguir a metodologia de gerenciamento de projetos;
 - Garantir que as métricas e indicadores estabelecidos sejam adequadamente coletadas e cumpridas;
 - Garantir durante as fases do projeto a utilização dos padrões, Catálogo de Tecnologia [CTGTEC], normas e metodologia vigentes na Prodabel bem como pela preparação e manutenção dos documentos relativos às atividades dos projetos estabelecidos na metodologia adotada pela Prodabel;
 - Planejar, executar e acompanhar as aquisições previstas em projetos utilizando os padrões, as normas e a metodologia vigente na Prodabel, documentar e preservar o histórico das atividades realizadas no desenvolvimento de projetos;
 - Buscar solução para acontecimentos inesperados e quaisquer outras situações adversas que surgirem durante o desenvolvimento das atividades e que possam comprometer o resultado ou as entregas;
 - Acompanhar o levantamento dos requisitos do projeto e suas prioridades junto ao Gerente de Relacionamento e ao Cliente;
 - Identificar pontos de melhoria do processo e registrar as lições aprendidas;

- Promover transparência sobre o andamento do projeto, registrando os resultados de reuniões, ocorrências e cronogramas;
 - Fornecer informações para elaboração dos relatórios analíticos de acompanhamento do projeto pelo analista de projeto em execução pelas Células de Resultado e distribuí-los sistematicamente aos membros do Comitê Gestor e aos níveis estratégicos da Prodabel;
 - Definir quais serão os impactos de solicitações de mudanças dos clientes, com respectivos prazos e custos, e encaminhar ao Comitê Gestor para avaliação e deliberação;
 - Apoiar o Coordenador da Célula de Resultado nas tarefas de planejamento e acompanhamento das demandas quando solicitado.
- **Gerente de Relacionamento (GR)**

Responsabilidades / atribuições em uma célula de resultados:

- Profissional responsável por fazer a interlocução com o cliente, atuando como seu porta voz quando da apresentação de suas demandas ao Comitê Gestor e no alinhamento de suas expectativas com os demais integrantes da Célula de Resultados;
- Encaminhar as demandas de projetos ou serviços, devidamente estimadas, para o Comitê Gestor do Cliente e se aprovado encaminhar ao Comitê de Tecnologia da Prodabel (Diretoria), registrando-as no Portal da Gestão para deliberação;
- Atuar como ponto focal de relacionamento entre o nível estratégico da PBH e a Prodabel em relação às demandas de projetos e serviços de TIC;
- Representar o interesse do cliente e gerenciar suas expectativas e necessidades, bem como assessorá-lo nos estudos de viabilidade técnica para suas demandas de projetos;
- Fazer a ponte entre a área de negócios (Cliente) e a equipe da Célula de Resultado, acompanhando a tramitação de todas as fases de execução das

atividades de interesse de seus clientes;

- Entender as necessidades e prioridades do cliente e agir como seu porta-voz, para garantir que a solução será desenvolvida de acordo com as especificações;
 - Participar do Comitê Gestor na análise e definição das prioridades de atendimento às demandas apontadas pelos clientes;
 - Definir, em conjunto com o coordenador da Célula de Resultado e/ou Gerente de Projeto, os critérios de aceitação para cada entrega ao cliente e garantir que tais critérios estão especificados e que os testes que verificam esses critérios foram executados;
 - Confirmar com o cliente, por meio de pesquisa de satisfação, se o produto entregue atendeu de forma satisfatória às expectativas, bem como retroalimentar o Comitê Gestor em caso insatisfatório;
 - Colaborar com a equipe da Célula de Resultado, de forma direta e estreita, esclarecendo dúvidas e prestando informações sobre as demandas dos clientes durante todas as etapas de desenvolvimento das várias atividades;
 - Estar preparado para apoiar o cliente na tomada de decisões mais sérias como mudanças de priorização, escopo, prazos e orçamento;
 - Receber, tramitar e acompanhar outras demandas de seus clientes, que não sejam objeto de atuação da Célula de Resultados;
 - Responsável por auxiliar o cliente na priorização, juntamente com o Coordenador (PO's ou outros envolvidos necessários), de incidentes para a célula.
- **Equipe do Projeto:** todos os envolvidos no projeto, equipe da instituição e o Time *Scrum*.