

Metodologia de Desenvolvimento de Sistemas (MDS)

Processo de Desenvolvimento Ágil de Software
(PDAS)



Câmara Legislativa do Distrito Federal

Diretoria de Modernização e Inovação Digital - DMI

Setor de Administração e Desenvolvimento de Sistemas - SEASI



Sumário

1	Introdução	5
2	Conceitos	5
2.1	Modelos Ágeis	6
2.2	Framework Scrum	6
2.2.1	Papel <i>Scrum Product Owner</i> (PO)	7
2.2.2	Papel <i>Scrum Master</i>	8
2.2.3	Time de Desenvolvimento	8
2.2.4	Ciclo de desenvolvimento <i>Scrum</i>	9
3	Fluxo do Processo	10
3.1	Processo de Desenvolvimento Ágil de Software	12
3.2	Subprocesso Sprint	12
3.3	Subprocesso Realizar Ateste Técnico	13
3.4	Subprocesso Realizar Reuniões de Planejamento	13
4	Classificação de Demandas	14
4.1	Sistema novo	14
4.2	Manutenção evolutiva	14
4.3	Manutenção corretiva	15
4.4	Refatoração	15
5	Atores do Processo	15
5.1	CLDF	15
5.1.1	Área de Negócios (AN)	15
5.1.1.1	Líder de Negócios (LN)	16
5.1.2	Área de TI (AT)	16
5.1.2.1	Líder Técnico (LT)	16
5.2	Fábricas	18
5.2.1	Fábrica de Software (FS)	18
5.2.1.1	Time de Desenvolvimento da Fábrica de Software	18
5.2.1.2	<i>Scrum Master</i> da Fábrica de Software	18
5.2.1.3	Representante Técnico da Fábrica de Software	18
5.2.2	Fábrica de Métricas (FM)	18
5.2.2.1	Representante Técnico da Fábrica de Métricas	18
5.2.3	Fábrica de Qualidade (FQ)	18
5.2.3.1	Representante Técnico da Fábrica de Qualidade	19
6	Reuniões	19
6.1	Reuniões de Planejamento	19
6.1.1	Realizar Reuniões entre TI e Negócios	19
6.1.2	Realizar Reuniões entre CLDF e Fábricas	20
6.1.2.1	Objetivo: Apresentar Líderes CLDF e Representantes Fábricas	20
6.1.2.2	Objetivo: Apresentar este MDS e o PDAS	20



6.1.2.3	Objetivo: Apresentar o “Documento de Visão” do projeto que será desenvolvido	20
6.1.2.4	Objetivo: Elaborar e aprovar o “Documento de Definição de Artefatos”	20
6.1.2.5	Objetivo: Elaborar e aprovar a “Forma da Entrega” dos entregáveis .	20
6.1.2.6	Objetivo: Elaborar e aprovar a “Definição de Pronto”	21
6.1.2.7	Objetivo: Determinar a “Duração da <i>Sprint</i> ”	21
6.1.2.8	Objetivo: Amadurecer o <i>Product Backlog</i>	21
6.2	Reuniões de Desenvolvimento	21
6.2.1	Reunião de Planejamento da <i>Sprint</i>	21
6.2.1.1	Fase Inicial	21
6.2.1.2	Fase Final	22
6.2.2	Reunião Diária	22
6.2.3	Reunião de Encerramento da Sprint	22
6.3	Reunião de Encerramento	22
7	Artefatos	23
7.1	Artefatos originais <i>Scrum</i>	23
7.1.1	Documento de Visão	23
7.1.2	Product Backlog	24
7.1.3	Sprint Backlog	24
7.1.4	História de usuário	25
7.1.4.1	Formato Clássico	25
7.1.4.2	Sugestões para produzir boas histórias de usuário	25
7.1.4.3	Histórias de usuário na prática	27
7.1.5	Definição de Pronto	28
7.1.6	Incremento do Produto	30
7.2	Artefatos mínimos do processo	30
7.2.1	Documento de Definição de Líder Técnico e Líder de Negócios	30
7.2.2	Documento de Definição de Artefatos	30
7.2.3	Forma das Entregas	30
7.2.4	Pacote de Produtos de Testes e Controle de Qualidade	30
7.2.5	Pacote de Produtos de Medição de Sistemas	31
7.3	Artefatos sugeridos	31
7.3.1	Fase de Planejamento	31
7.3.2	Fase de Desenvolvimento	31
7.3.2.1	Gráficos de Acompanhamento do Trabalho	31
7.3.2.2	Relatório Gerencial	32
7.3.3	Fase de Encerramento	34
7.3.4	Artefatos de Medição de Sistemas	34
7.3.5	Artefatos adicionais	34
7.3.5.1	Termo de Aceite da Fase (TAF)	34
7.3.5.2	Atas de Reunião	35



8 Detalhes do Processo	35
8.1 Processo de Desenvolvimento Ágil de Software	35
8.1.1 Evento: “Recebimento de Demanda”	36
8.1.2 Subprocesso: “Realizar Reuniões de Planejamento”	37
8.1.3 Subprocesso: “Realizar Ateste Técnico”	38
8.1.4 Decisão: “Validado?”	39
8.1.5 Processo: “Verificar Aplicação de Sanções”	40
8.1.6 Processo: “Realizar Aceitação da Fase”	41
8.1.7 Subprocesso: “Sprint”	42
8.1.8 Decisão: “Haverá outra Sprint?”	43
8.1.9 Processo: “Realizar Reunião de Encerramento”	44
8.1.10 Evento: “Final do Processo”	44
8.2 Subprocesso <i>Sprint</i>	45
8.2.1 Evento: “Sinal de Início”	45
8.2.2 Processo: “Realizar Reunião de Planejamento da Sprint”	46
8.2.3 Processo: “Desenvolver”	47
8.2.4 Processo: “Entregar”	48
8.2.5 Processo: “Acompanhar Sprint”	49
8.2.6 Processo: “Realizar Reunião de Encerramento da Sprint”	50
8.2.7 Evento: “Sinal de Fim”	51
8.3 Subprocesso Realizar Ateste Técnico	51
8.3.1 Evento: “Sinal de Início”	51
8.3.2 Processo: “Receber Entregáveis”	52
8.3.3 Decisão: “Todos os produtos entregues?”	52
8.3.4 Decisão: “Será precido medir?”	53
8.3.5 Processo: “Solicitar Serviço de Medição”	54
8.3.6 Processo: “Executar Serviço de Medição de Sistemas”	54
8.3.7 Decisão: “Será preciso testar?”	55
8.3.8 Processo: “Solicitar Serviço de Qualidade”	56
8.3.9 Processo: “Executar Serviço de Controle de Qualidade”	56
8.3.10 Processo: “Realizar Ateste Técnico”	57
8.3.11 Evento: “Sinal de Fim”	57
8.4 Subprocesso Realizar Reuniões de Planejamento	58
8.4.1 Evento: “Sinal de Início”	58
8.4.2 Processo: “Realizar Reuniões TI e Negócios”	59
8.4.3 Processo: “Realizar Reuniões CLDF e Fábricas (FS e FQ)”	60
8.4.4 Evento: “Sinal de Fim”	61
9 Normas Complementares	62
9.1 Normas Complementares para Contagem de Pontos de Função	62
9.1.1 Reunião de Mediação de Divergência de Aferição de Pontos de Função	62
10 Considerações Finais	62
DOCUMENTOS DE DIRETRIZES	63



1 Introdução

Uma metodologia pode ser definida como um conjunto de regras e procedimentos estabelecidos para se atingir um determinado fim. Do mesmo modo, uma Metodologia de Desenvolvimento de Sistemas (MDS) é uma coleção de conceitos e boas práticas em desenvolvimento de sistemas. O objetivo é apresentar os conceitos e estabelecer o processo necessário para guiar o desenvolvimento e a manutenção de sistemas corporativos.

Esse documento visa estabelecer o processo de interação entre a Câmara Legislativa do Distrito Federal e as Fábricas de Software, Qualidade e Métricas contratadas para fazer o desenvolvimento e manutenção de sistemas, avaliação de aspectos de qualidade e a medição dos produtos desenvolvidos, respectivamente, com o objetivo de entregar soluções corretas, com qualidade e no prazo que atendam aos requisitos de negócio no âmbito dessa instituição. Portanto, além dos conceitos e definições, o principal produto da metodologia é o Processo de Desenvolvimento Ágil de Software (PDAS). Trata-se de um fluxo de processos e subprocessos que deverão guiar essa interação.

Esse documento foi escrito com base em outras metodologias elaboradas por órgãos e autarquias do governo que também optaram pela estratégia de *outsourcing*. Em particular, o processo foi adaptado da Metodologia Midas (IPHAN, 2013) juntamente com elementos de metodologias de desenvolvimento ágil do Senado Federal, da Câmara dos Deputados e da Agência Nacional de Energia Elétrica (ANEEL). Também tomamos o cuidado de pesquisar a jurisprudência de órgãos de controle identificando acórdãos e atas sobre o tema de forma a não repetir práticas inadequadas. Além disso, o desenvolvimento levou em consideração os elementos de mitigação de riscos da análise realizada na fase de planejamento da contratação. O resultado é uma metodologia desenvolvida à luz do modelo ágil *Scrum*, mas adaptada e adequada à realidade institucional da Câmara Legislativa do Distrito Federal.

Assim, iniciamos o documento nesse capítulo 1 onde introduzimos o que é uma Metodologia de Desenvolvimento de Sistemas (MDS) e qual é o seu propósito. O capítulo 2 apresenta os principais conceitos envolvidos: descrevemos o que são modelos ágeis e explicamos como funciona o *framework* original *Scrum*. Em seguida, o Fluxo do Processo é apresentado no capítulo 3. Neste momento, apresentamos o Processo de Desenvolvimento Ágil de Software (PDAS) e também os subprocessos que o integram. Os demais capítulos apresentam os elementos necessários para entender o processo como um todo. Dessa forma, no capítulo 4, as demandas por desenvolvimento de sistemas são classificadas.

Desse ponto segue-se identificando os principais atores do processo, as reuniões e os principais artefatos utilizados (capítulos 5, 6 e 7). Em seguida, o capítulo 8 consolida o entendimento detalhando cada atividade que compõe o processo.

Finalmente, o capítulo 9 dispõe sobre as Normas Complementares ao PDAS, detalhando procedimentos operacionais e regras de execução contratual aplicáveis aos processos e subprocessos da metodologia. O capítulo 10 apresenta as considerações finais e, ao final, o anexo contendo os Documentos de Diretrizes consolida os padrões técnicos e normativos aplicáveis ao desenvolvimento, implantação e manutenção de sistemas no âmbito institucional.

2 Conceitos

O intuito desta seção é apresentar alguns conceitos básicos necessários para se compreender o processo de desenvolvimento ágil elaborado para atender as necessidades da Câmara Legislativa do Distrito Federal.



2.1 Modelos Ágeis

Agilidade é a capacidade de se adaptar a mudanças de requisitos, de equipe e de tecnologia. Um processo ágil é necessário quando os requisitos e prioridades são instáveis, quando tanto o projeto e a construção são realizados simultaneamente ou quando durante o planejamento as fases de análise, projeto, implementação e testes não são tão previsíveis.

Modelos ágeis começaram a aparecer a partir do ano de 2001 quando um grupo de 17 especialistas se reuniram na estação de ski *Snowbird*, em Utah, nos Estados Unidos, para discutir maneiras de desenvolver software de uma forma mais leve, rápida e centrada em pessoas (Prikladnicki et al., 2014). Eles cunharam os termos “Desenvolvimento Ágil de Software” e “Métodos Ágeis” e criaram o Manifesto Ágil apresentado a seguir:

Manifesto Ágil

“Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

1. Indivíduos e interações mais que processos e ferramentas;
2. Software em funcionamento mais que documentação abrangente;
3. Colaboração com o cliente mais que negociação de contratos;
4. Responder a mudanças mais que seguir um plano.

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.”

A partir de então surgiram diversos modelos ágeis, dentre os quais podemos elencar: XP (*eXtreme Programming*), ASD (*Adaptative Software Development*), DSDM (*Dynamic Systems Development Method*), FDD (*Feature Driven Development*), *Kanban*, *Crystal* e o *SCRUM*. Esse último tem sido utilizado por organizações tanto públicas como privadas para desenvolver sistemas. A próxima seção apresenta de forma resumida o *framework Scrum*: principal referência do Processo de Desenvolvimento Ágil de Software (PDAS) apresentado nesse documento.

2.2 Framework Scrum

Ken Schwaber e Jeff Sutherland são os criadores do *Scrum*. Esse método ágil é descrito no famoso guia “*Scrum Guide - Um guia definitivo para o Scrum: As regras do Jogo*” (Sutherland e Schwaber, 2017). De acordo com esse guia, o *Scrum* é um *framework* para desenvolver, entregar e manter produtos complexos. Seu uso não se limita na gestão e desenvolvimento de software, mas para gerenciar quase tudo que usamos em nosso dia-dia, como indivíduos e sociedades. Particularmente, no âmbito do desenvolvimento ágil de software, o *Scrum* têm sido utilizado por diversas instituições públicas devido à sua efetividade e eficiência para tratar e resolver problemas complexos e adaptativos, enquanto entrega produtos com o mais alto valor possível para a organização.

De acordo com Prikladnicki et al. (2014), o *Scrum* distribui a gestão entre três papéis: O *Product Owner* (PO), o *Scrum Master* e o Time de Desenvolvimento. Cada papel *Scrum* possui um conjunto de responsabilidades bem definidas de forma a evitar conflitos entre os envolvidos no projeto. Os referidos papéis *Scrum* serão descritos nas próximas seções.



2.2.1 Papel Scrum Product Owner (PO)

O *Product Owner* (PO) representa os interesses dos *stakeholders* de negócio e deve ter conhecimento suficiente do negócio para responder aos questionamentos da equipe de desenvolvimento. É o responsável pelo Retorno de Investimento (ROI) e pela macrogestão do projeto. Representa todos os interessados.

As características e as atividades recomendadas para que o *Product Owner* desempenhe com sucesso o seu papel são:

- Conhecer o processo de negócio e seus objetivos;
- Ser o maior interessado no prazo;
- Ser representativo para o produto;
- Ter conhecimento e poder suficiente para tomar decisões rápidas e adequadas;
- Gerenciar as expectativas dos *stakeholders* entendendo suas necessidades e gerenciando expectativas e conflitos;
- Gerenciar o produto decidindo o que vai para o *product backlog* e, igualmente importante, o que não vai;
- Único que pode alterar o *product backlog*;
- Ordena os itens para entregar o valor mais elevado;
- Melhora continuamente a qualidade das histórias de usuário;
- Trabalhar em equipe e colaborar com o time de desenvolvimento;
- Realiza, com o Time de Desenvolvimento, o planejamento da *Sprint* e colabora sempre que necessário (refinamento);
- Participa das reuniões diárias (opcional);
- Não interfere no planejamento já realizado;
- Realiza o feedback apenas quando for solicitado;
- Aceitar mudanças;
- Aceitar ou rejeitar as entregas na reunião de encerramento da *Sprint*.
- Realiza a homologação do incremento do Produto;
- Ter iniciativa;
- Estar sempre acessível e disponível para tomar decisões e esclarecer dúvidas.



2.2.2 Papel *Scrum Master*

O *Scrum Master* é um facilitador para o trabalho do time de *Scrum*, ou seja, dos membros do time de desenvolvimento e do *Product Owner*. Ele promove a autonomia, a boa relação de trabalho e a comunicação entre os membros do time de *Scrum* no seu dia a dia, de forma a se tornarem cada vez mais efetivos.

As características e as atividades recomendadas para que o *Scrum Master* desempenhe com sucesso o seu papel são:

- Facilitador hábil;
- Garante o uso do *Scrum*;
- Responsável por garantir que os valores, práticas e regras do *Scrum* estejam sendo entendidos e seguidos por todo o time;
- Promover mudanças organizacionais necessárias;
- Atuar como um agente de mudanças na organização onde está inserido o time de *Scrum*. Ou seja, ele trabalha para promover as mudanças no contexto do time de *Scrum* necessárias para que a equipe possa realizar seu trabalho com mais efetividade;
- Remover impedimentos organizacionais e administrativos;
- Facilitador de eventos;
- Além de facilitar o dia a dia de trabalho do time de *Scrum*, o *Scrum Master* tem a importante função de atuar como um facilitador nos eventos do *Scrum*. Ele estimula os envolvidos a participarem ativamente das discussões, ajuda-os a manter o foco nos objetivos do evento e a chegarem a suas próprias conclusões.

2.2.3 Time de Desenvolvimento

O Time de Desenvolvimento é um grupo multidisciplinar de pessoas responsáveis por realizar o trabalho de desenvolvimento do sistema. A partir das prioridades definidas pelo *Product Owner*, o Time de Desenvolvimento gera, em cada *Sprint*, um incremento do produto “pronto”, de acordo com a “Definição de Pronto”, e que significa valor visível para os clientes do projeto.

O Time de Desenvolvimento gerencia o seu trabalho de desenvolvimento ou manutenção do sistema. É ele que determina tecnicamente como o produto será desenvolvido, planeja esse trabalho e acompanha seu progresso. Para tal, tem propriedade e autoridade sobre suas decisões e, ao mesmo tempo, é responsável e responsabilizado por seus resultados.

Para realizar esse trabalho, o Time de Desenvolvimento:

- Planeja seu trabalho, definindo com o *Product Owner* o que será realizado no decorrer da *Sprint*, para então detalhar, de forma autônoma, como esse trabalho será realizado;
- Realiza as tarefas de desenvolvimento do produto para atingir a Meta da *Sprint*, garantindo a qualidade do que é produzido, além de acompanhar seu progresso no *Sprint* em direção a essa meta;



- Interage com o *Product Owner*, sempre que necessário, para ter dúvidas esclarecidas ou solicitar decisões quanto ao produto, e colabora com ele para refinar e aprimorar o *Product Backlog*, preparando-o para o próximo *Sprint*;
- Identifica e informa ao *Scrum Master* sobre impedimentos que obstruam seu trabalho e previne-se deles, quando possível;
- Obtém *feedback* dos clientes do projeto e demais partes interessadas sobre o trabalho realizado durante e ao final da *Sprint*;
- Entrega valor com frequência para os clientes do projeto na forma de “*Incremento de Produto*”;
- O Time de Desenvolvimento é:
 - Multidisciplinar, possuindo todas as habilidades e conhecimentos necessários para gerar, em cada *Sprint*, o “*Incremento do Produto*” “Pronto”, de acordo com a “Definição de Pronto”;
 - Auto-organizado, planejando e executando seu trabalho com autonomia, propriedade e responsabilidade;
 - Suficientemente pequeno (de 3 a 9 pessoas), de forma que seus membros se comuniquem efetivamente e se auto organizem, sendo capazes de produzir incrementos do produto prontos que representem valor visível para os clientes;
 - Motivado, uma vez que possua o ambiente, apoio e a confiança necessários para realizar seu trabalho;
 - Orientado à excelência técnica, buscando sempre aprender e realizar seu trabalho com qualidade e consciência;
 - Focado nas metas estabelecidas junto ao *Product Owner*.

Esses papéis interagem entre si durante o ciclo de desenvolvimento *Scrum* participando de eventos, produzindo e utilizando artefatos diversos durante ciclos iterativos onde os incrementos do produto são desenvolvidos.

2.2.4 Ciclo de desenvolvimento *Scrum*

A figura 1 apresenta o ciclo de desenvolvimento *Scrum* de forma simplificada. Inicialmente o *Product Owner* define a “Visão do Produto” que representa sua necessidade e é o que deve ser satisfeito no fim do projeto.

Em seguida, o *Product Owner*, com auxílio do *Scrum Master*, cria uma lista inicial de necessidades que precisam ser produzidas para que a visão do projeto seja bem sucedida. Essa lista é denominada de *Product Backlog*. Ela consiste de histórias de usuário priorizadas pelo *Product Owner*.

A seguir ocorre a reunião “Planejamento da Sprint”, ou seja, antes de iniciar cada iteração (*Sprint*) o time deve se reunir e definir o que deverá ser entregue ao final do ciclo. Essa reunião tem duas fases geralmente com 4h de duração cada uma. Na primeira fase, o *Product Owner* deverá definir a meta da *Sprint* e expor para o time os itens mais prioritários do *Product Backlog*. O time, por sua vez, deve estimar os itens em tamanho e definir o que acredita que pode ser implementado dentro da *Sprint*. Essa listagem é chamada de *Selected Product Backlog*. Na segunda fase, o time deverá colher mais detalhes do *Selected Product Backlog* e decompô-los em tarefas gerando assim o artefato *Sprint Backlog*. Tanto na primeira quanto na segunda fase dessa reunião o *Scrum Master* deve atuar como facilitador.

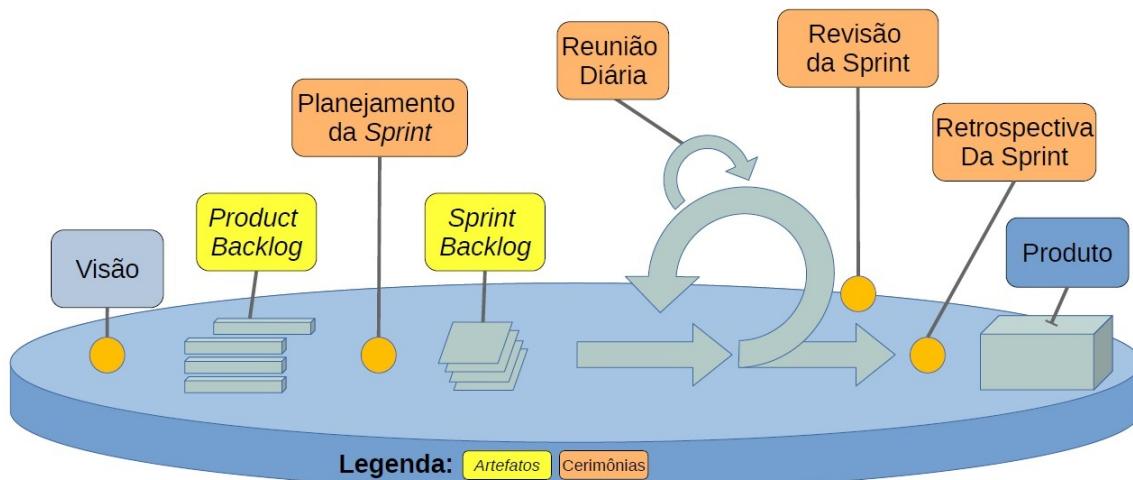


Figure 1: Ciclo do Scrum

Dessa forma, segue-se com a realização da *Sprint*, que é um período de 2 a 4 semanas em que o time vai desenvolver o incremento potencialmente entregável do produto. Durante a *Sprint* deverão ocorrer reuniões diárias entre os membros da equipe de desenvolvimento. Essas reunião são conhecidas como “Scrums Diárias”, ou simplesmente “Reuniões Diárias”. Essas reuniões devem durar no máximo 15 minutos e é o momento em que cada membro responde aos outros membros as perguntas: O que eu fiz desde a última Scrum Diária? O que pretendo fazer até a próxima Scrum Diária? Existe algo me impedindo de concluir alguma tarefa? Havendo problemas, a equipe se auto-organiza para resolvê-los. Aqueles problemas que a equipe não se considera apta para resolver são classificados como impedimentos e repassados para o *Scrum Master* resolver.

Ao final da *Sprint*, ocorre uma outra reunião denominada “Revisão da *Sprint*”. Nessa reunião, com duração máxima de 4 horas, o time deve apresentar os resultados para o *Product Owner* avaliar se a meta foi ou não atingida. Nessa reunião o *Product Owner* também pode fazer anotações que poderão se tornar novos itens do *Product Backlog*.

Há uma segunda e ultima reunião denominada “Retrospectiva da *Sprint*” com duração máxima de 3 horas mediada pelo *Scrum Master*. Essa é uma reunião para o time avaliar lições aprendidas, identificar o que foi bom e o que deve ser melhorado. Esta reunião representa a proposta do espírito de inspeção-adaptação do *SCRUM* e caso o time concorde, pode haver a participação do *Product Owner*.

Em suma, o ciclo de desenvolvimento original do *framework Scrum* discutido anteriormente foi adaptado e adequado à realidade institucional da Câmara Legislativa do Distrito Federal resultando no Processo de Desenvolvimento Ágil de Software cujo fluxo será apresentado no próximo capítulo.

3 Fluxo do Processo

As próximas seções irão apresentar o fluxo do Processo de Desenvolvimento Ágil de Software para guiar a interação entre os atores da CLDF e das Fábricas contratadas. Iniciamos apresentando o macro fluxo principal e depois apresentamos os subprocessos envolvidos.

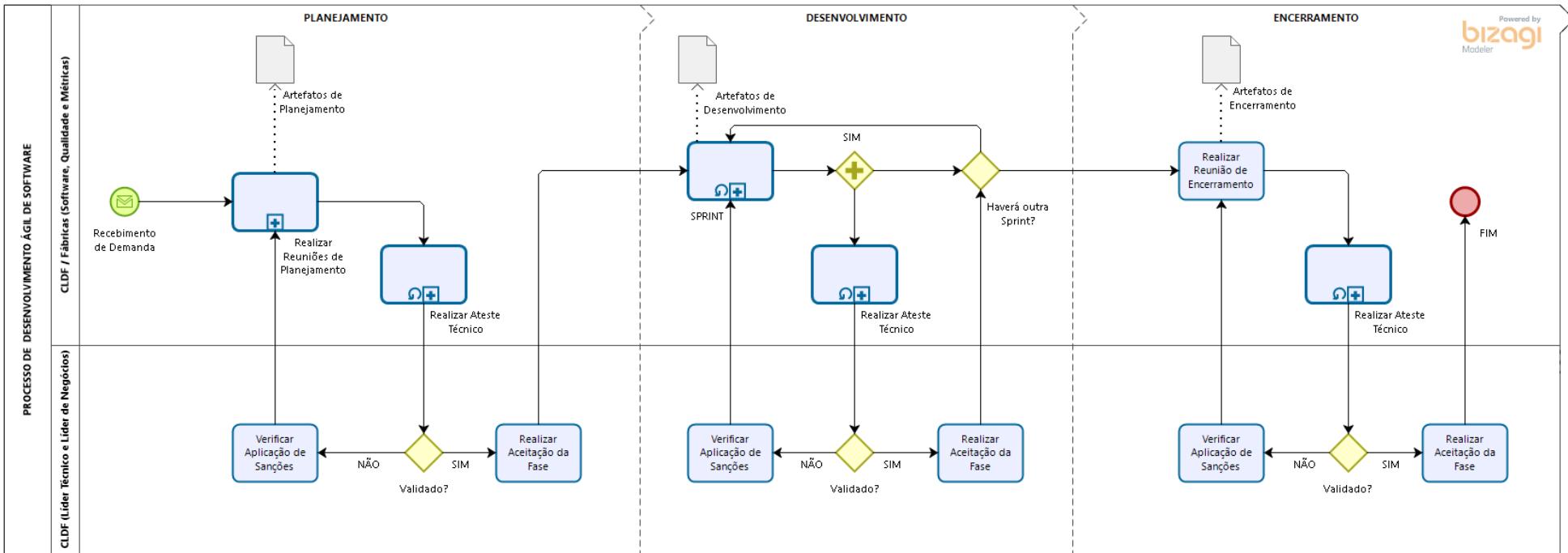


Figure 2: Processo de Desenvolvimento Ágil de Software (PDAS)



3.1 Processo de Desenvolvimento Ágil de Software

A figura 2 apresenta o Processo de Desenvolvimento Ágil de Software que se divide nas fases de Planejamento, Desenvolvimento e Encerramento.

O processo se inicia na Fase de Planejamento com o “Recebimento da Demanda” e segue para o “Subprocesso Realizar Reuniões de Planejamento” onde as reuniões de planejamento são realizadas e os artefatos de planejamento são elaborados e definidos.

Depois ele segue para o “Subprocesso Realizar Ateste Técnico” onde os artefatos de planejamento elaborados são validados. Se eles forem válidos, o processo segue para o subprocesso “Realizar Aceitação da Fase”. Se não, o processo segue para o subprocesso “Verificar Aplicação de Sanções”. O primeiro realiza a aceitação da fase de planejamento, enquanto o último verifica se cabem sanções, aplica-as e encaminha o processo para que sejam realizados novas reuniões de planejamento de sorte que os artefatos sejam reavaliados, corrigidos e completados.

Após o aceite da fase de planejamento, o processo segue para a Fase de Desenvolvimento. É nesta fase que os *sprints* de desenvolvimento serão realizados. Dessa forma, os artefatos de saída do Subprocesso Sprint também passarão pelo Subprocesso Realizar Ateste Técnico e pelo processo de validação tal qual ocorreu com os artefatos de planejamento da fase anterior. A diferença é que o subprocesso “Realizar Aceitação da Fase” pode aceitar a *sprint* como válida, mas mesmo assim exigir que uma nova *sprint* seja realizada. E haverão tantas *sprints* quanto forem necessárias até que se decida que não são necessárias mais *sprints* de desenvolvimento.

Em seguida, o processo chega na Fase de Encerramento. Nesta fase realiza-se uma “Reunião de Encerramento” que deve gerar “Artefatos de Encerramento”. Novamente ocorre um ciclo de ateste técnico e validação até a aceitação da fase ser realizada e o processo chegar ao seu fim.

3.2 Subprocesso Sprint

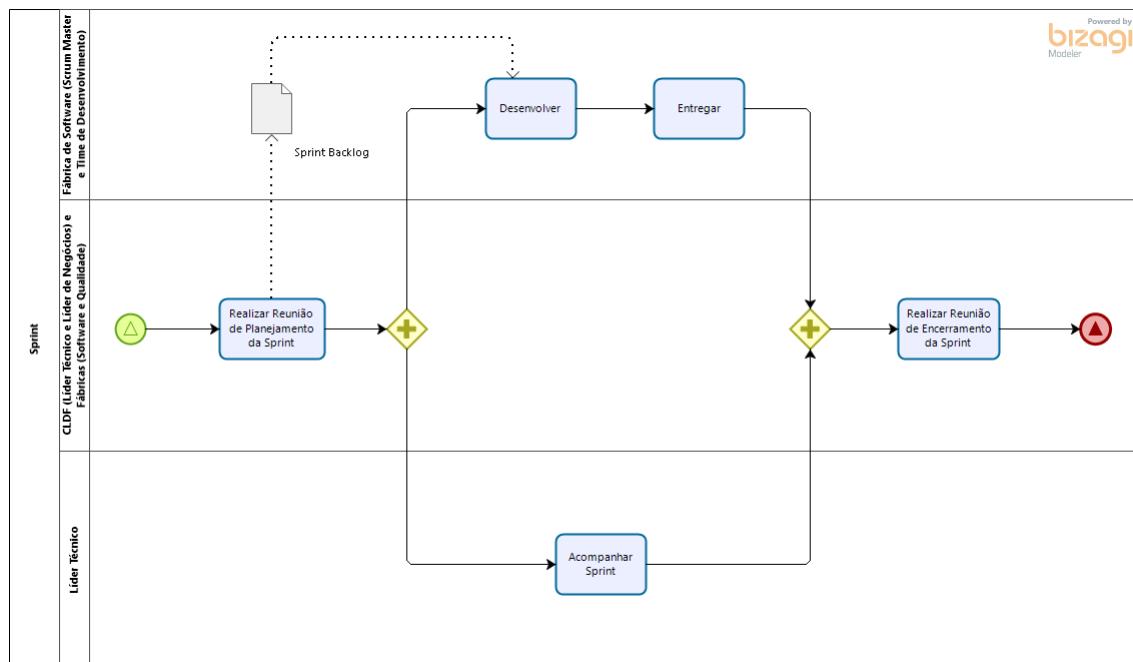


Figure 3: Subprocesso Sprint



A figura 3 apresenta o subprocesso *Sprint* que é onde a Fábrica de Software realiza o desenvolvimento do sistema propriamente dito.

O processo começa com a “Reunião de Planejamento da Sprint”. Esta é uma reunião onde a *Sprint Backlog* para esta *sprint* será definida.

Em seguida ocorrem duas atividades em paralelo: Enquanto a fábrica de software desenvolve e entrega os entregáveis definidos na reunião, os líderes da CLDF (TI e Negócios) acompanham esse desenvolvimento.

Finalmente, o processo termina após a realização da “Reunião de Encerramento da Sprint”.

3.3 Subprocesso Realizar Ateste Técnico

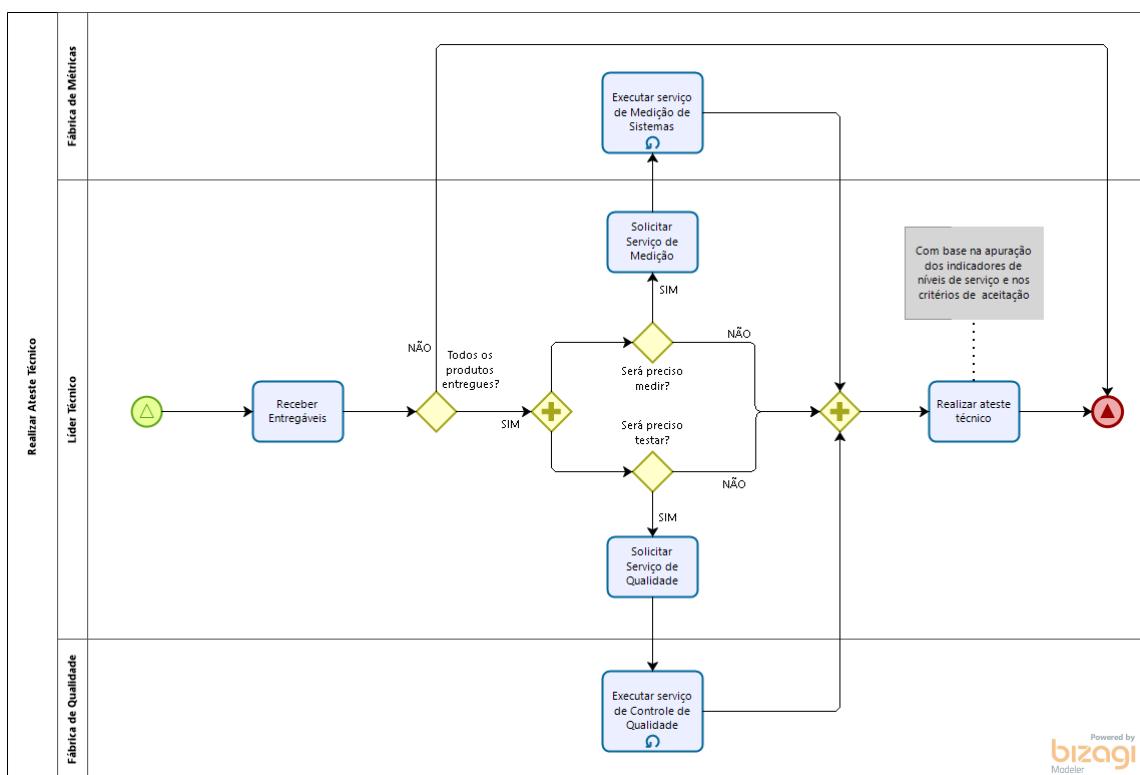


Figure 4: Subprocesso Realizar Ateste Técnico

Já na figura 4 apresentamos o subprocesso que ilustra como os atestes técnicos das diversas fases são realizados. Basicamente são dois tipos de ateste: Serviço de Medição e Serviço de Qualidade. Sua realização não é obrigatória e a decisão de realizar esses serviços, ou não, é tomada pelos líderes de TI e Negócios.

3.4 Subprocesso Realizar Reuniões de Planejamento

O Subprocesso Realizar Reuniões de Planejamento é mostrado na figura 5. Inicialmente, uma ou mais reuniões entre as Áreas de TI e de Negócios da Câmara Legislativa do Distrito Federal são necessárias para que estas áreas definam seus respectivos líderes. Nessas reuniões as Histórias de Usuário do “Product Backlog” já podem começar a ser criadas.

Em seguida, deve-se preparar uma ou mais reuniões entre os líderes da CLDF e os representantes das Fábricas de Software e de Qualidade. Essas reuniões deverão gerar os “Artefatos de Planejamento”.

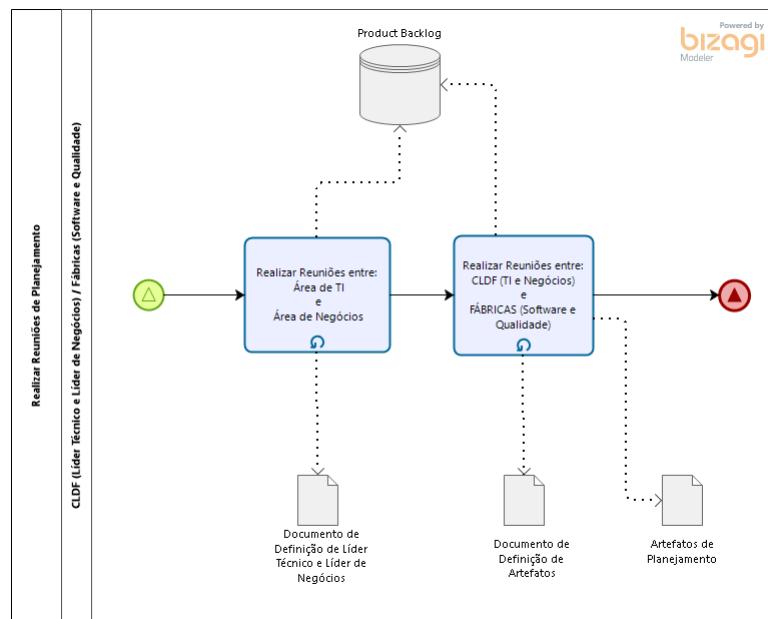


Figure 5: Subprocesso Realizar Reuniões de Planejamento

Esses artefatos são acordos, como por exemplo, critérios de aceitação de qualidade, etc.

Quando todos os artefatos forem definidos, o subprocesso se encerra.

4 Classificação de Demandas

As demandas por desenvolvimento de sistemas serão divididas em tipos:

1. Sistema Novo
2. Manutenção Evolutiva
3. Manutenção Corretiva
4. Refatoração

4.1 Sistema novo

Serão considerados neste tipo de demanda os projetos de:

1. Sistema a ser integralmente desenvolvido;
2. Sistema reconstruído a partir de um legado;
3. Sistema desenvolvido a partir de outros sistemas pré-existentes, em todo ou em parte, que nunca entraram em produção;
4. Sistema desenvolvido a partir de um ou mais sistemas de outro(s) órgão(s) ou entidade(s), cujo código fonte foi, em todo ou em parte, cedido ou repassado à CLDF, ou obtido por outros meios.

4.2 Manutenção evolutiva

Manutenção evolutiva aborda as mudanças em requisitos funcionais da aplicação, ou seja, a inclusão de novas funcionalidades, alteração ou exclusão de funcionalidades em aplicações implantadas (MP, 2016).

A Manutenção Adaptativa, Perfectiva e Cosmética são tipos de Manutenção Evolutiva.



4.3 Manutenção corretiva

Manutenção corretiva é a intervenção em um sistema cujas funcionalidades passaram a apresentar defeito, afetando sua qualidade funcional, executada para mantê-lo em estado operacional.

É importante destacar que as demandas por manutenção corretiva precisam ser atendidas com urgência.

Caso o sistema esteja em garantia, sob responsabilidade da empresa que o desenvolveu, esta será acionada para suas devidas correções, nas condições e prazos estabelecidos; nesse caso, a manutenção corretiva será considerada acionamento da garantia.

4.4 Refatoração

A refatoração é uma forma disciplinada de reestruturar o código quando pequenas mudanças são feitas nele para melhorar o design. Um aspecto importante de uma refatoração é que ela melhora o design sem mudar o comportamento do design; uma refatoração não adiciona nem remove funcionalidade (Fowler, 2018). Nesta metodologia, a refatoração é uma demanda de adequação do sistema, a fim de alterar uma funcionalidade que já foi implementada, entregue e validada. A alteração ou adequação do sistema para finalizar uma funcionalidade que necessite ser implementada em mais de uma Sprint não será considerada uma refatoração.

5 Atores do Processo

Um ator representa um conjunto coerente de papéis que os usuários do processo desempenham quando de sua execução. Tipicamente, um ator representa um papel que uma entidade desempenha durante a execução de um processo. Nesse contexto, ele é visto como um conjunto de atribuições, funções e/ou responsabilidades que um ator possui.

5.1 CLDF

A Câmara Legislativa do Distrito Federal (CLDF) é a principal entidade beneficiada pelos sistemas corporativos desenvolvidos usando esta metodologia.

Sua estrutura organizacional envolve diversas unidades administrativas. A Diretoria de Modernização e Inovação Digital (DMI) é a unidade que tem por finalidade o assessoramento especializado em computação à Mesa Diretora e o contínuo aperfeiçoamento do Sistema de Informação da CLDF.

Portanto, podemos categorizar as unidades administrativas da CLDF em duas áreas: A área de TI (DMI) e a área não-TI à qual devem pertencer todas as outras unidades. Em computação, o nome que se dá a essa área é Área de Negócio. Trata-se da área de conhecimento onde um sistema será desenvolvido e cuja unidade administrativa será a principal gestora do sistema.

5.1.1 Área de Negócios (AN)

Área de negócios é a unidade administrativa da Câmara Legislativa do Distrito Federal que demanda a solução de tecnologia da informação.



5.1.1.1 Líder de Negócios (LN)

Servidor representante da área de negócios, indicado pela autoridade competente dessa área, com capacidade técnica relacionada à área de negócio em que a mesma atua.

Dentre as responsabilidades do Líder de Negócios que o diferencia do Líder Técnico (descrito adiante), podemos elencar:

- Direcionar o projeto sob a perspectiva do negócio, priorizando demandas e alinhando expectativas com as áreas envolvidas e com a Alta Administração;
- Garantir o patrocínio institucional do projeto e sua aderência às diretrizes estratégicas;
- Atuar como representante da área gestora do sistema, fornecendo ou articulando as informações necessárias ao adequado levantamento de requisitos;
- Efetuar o levantamento e a consolidação dos requisitos de negócio, definindo as características e funcionalidades do produto a ser entregue;
- Assegurar que as demandas encaminhadas à empresa contratada estejam alinhadas às necessidades dos usuários e aos objetivos institucionais;
- Validar, ao final de cada Sprint, as funcionalidades entregues sob a perspectiva do negócio, verificando sua aderência aos requisitos definidos;
- Manifestar aceite ou solicitar ajustes nas entregas quanto ao atendimento das regras de negócio, sem prejuízo da avaliação técnica realizada pelo Líder Técnico;
- Homologar as funcionalidades aprovadas para fins de entrada em produção;
- Assinar o termo de recebimento das funcionalidades sob a ótica negocial.

5.1.2 Área de TI (AT)

Área de Tecnologia da Informação é a unidade setorial ou seccional do SISP, bem como área correlata, responsável por gerir a Tecnologia da Informação na CLDF. Conforme mencionado, no âmbito da Câmara Legislativa do Distrito Federal a área de TI é a Diretoria de Modernização e Inovação Digital (DMI).

5.1.2.1 Líder Técnico (LT)

Líder técnico é o servidor representante da área de tecnologia da informação, indicado pela autoridade competente dessa área, com conhecimento técnico relacionado à solução.

Em relação aos papéis *Scrum*, o Líder Técnico é uma das partes interessadas (*stakeholder*) do processo. Embora, em modelos tradicionais, o papel de *Product Owner* seja exercido por representante exclusivo da Área de Negócios, a experiência institucional da Câmara Legislativa do Distrito Federal demonstra particularidades relevantes.

Na CLDF, a responsabilidade pelo desenvolvimento e pela sustentação dos sistemas é atribuída à Diretoria de Modernização e Inovação Digital (DMI). Em projetos de maior porte, é incomum a existência de um único servidor da Área de Negócios com visão sistêmica abrangente do produto. Em regra, há a participação de dois ou três servidores representando a área demandante, o que exige coordenação técnica centralizada para assegurar consistência arquitetural, coerência evolutiva e integridade do produto.



Nesse contexto, consolidou-se como prática institucional que o Líder Técnico exerce também o papel de *Product Owner*, sendo responsável pelo gerenciamento do *product backlog*, pela harmonização das demandas negociais com as diretrizes técnicas e pela preservação da visão integrada do sistema. Tal arranjo busca garantir continuidade, consistência e qualidade na evolução das soluções desenvolvidas pela CLDF.

Assim, dentre as responsabilidades do Líder Técnico que o diferencia do Líder de Negócios, podemos elencar:

- Assegurar que o time de desenvolvimento funcione plenamente e com alta produtividade, removendo impedimentos, barreiras e burocracias que comprometam seu desempenho;
- Proteger a equipe de interferências externas e garantir ambiente adequado ao desenvolvimento;
- Estimular a colaboração entre os integrantes do projeto e promover alinhamento contínuo com as áreas envolvidas;
- Garantir que a metodologia esteja sendo corretamente aplicada e que todos os participantes exerçam adequadamente seus papéis;
- Influenciar e orientar o Líder de Negócios, detalhando requisitos em nível técnico e apoiando a priorização do *product backlog*;
- Definir, em conjunto com a equipe, as estratégias de desenvolvimento, assegurando coerência arquitetural e aderência aos padrões técnicos da CLDF;
- Acompanhar o time de desenvolvimento, monitorando evolução, riscos e impedimentos;
- Participar dos processos de medição dos serviços, inclusive junto à Fábrica de Métricas;
- Participar dos processos de testes e garantia da qualidade, inclusive junto à Fábrica de Qualidade;
- Avaliar as entregas sob a perspectiva técnica, podendo aceitá-las, rejeitá-las ou solicitar ajustes quanto à qualidade e conformidade com os padrões institucionais;
- Verificar o atingimento dos Níveis Mínimos de Serviço (NMS);
- Assinar a Ordem de Serviço;
- Elaborar e assinar o Termo de Recebimento Provisório;
- Providenciar as evidências de cumprimento ou descumprimento dos SLAs para subsidiar a elaboração do Relatório Técnico de Fiscalização;
- Revisar e assinar o Relatório Técnico de Fiscalização.

O Líder Técnico e o Líder de Negócios trabalham juntos representando os interesses da Câmara Legislativa do Distrito Federal no projeto e, por isso, muitas vezes serão chamados de Líderes CLDF.



5.2 Fábricas

5.2.1 Fábrica de Software (FS)

Fábrica de software é a entidade contratada formada pelo conjunto de profissionais, recursos materiais, processos e metodologias para o desenvolvimento de softwares, sistemas e aplicações, englobando desde a criação de histórias de usuário, análises de requisitos até a fase de manutenção. A Fábrica de Software engloba o time de desenvolvimento, o *Scrum Master* e um Representante Técnico, conforme será detalhado nas próximas subseções.

5.2.1.1 Time de Desenvolvimento da Fábrica de Software

O Time de Desenvolvimento consiste de profissionais da fábrica de software contratada que realizam o trabalho de entregar um incremento potencialmente entregável do produto “Pronto” ao final de cada *Sprint*.

5.2.1.2 Scrum Master da Fábrica de Software

Scrum Master da fábrica de software é o profissional da fábrica de software contratada que atua segundo as características anteriormente descritas na seção 2.2.2. Ele trabalha em conjunto com o time de desenvolvimento e realiza o trabalho de entregar um incremento potencialmente liberável do produto “Pronto” ao final de cada Sprint.

5.2.1.3 Representante Técnico da Fábrica de Software

Representante da fábrica de software, responsável por atuar como interlocutor principal junto à área de tecnologia de informação, incumbido de receber, diligenciar, encaminhar e responder as principais questões técnicas, legais e administrativas referentes aos softwares que estiverem sendo desenvolvidos.

Não deve-se confundir o Representante Técnico da Fábrica de Software com o *Scrum Master* da fábrica de software. O primeiro atua na esfera técnica, legal e administrativa enquanto o *Scrum Master* deve fazer parte do time de desenvolvimento da fábrica de software.

5.2.2 Fábrica de Métricas (FM)

Entidade responsável pela prestação dos serviços de medição dos sistemas.

5.2.2.1 Representante Técnico da Fábrica de Métricas

Representante da fábrica de métricas, responsável por atuar como interlocutor principal junto à área de tecnologia de informação, incumbido de receber, diligenciar, encaminhar e responder as principais questões técnicas, legais e administrativas referentes aos softwares que estiverem sendo avaliados em relação a aspectos de medição.

5.2.3 Fábrica de Qualidade (FQ)

Entidade responsável pela prestação dos serviços de controle da qualidade de sistemas.



5.2.3.1 Representante Técnico da Fábrica de Qualidade

Representante da fábrica de qualidade, responsável por atuar como interlocutor principal junto à área de tecnologia de informação, incumbido de receber, diligenciar, encaminhar e responder as principais questões técnicas, legais e administrativas referentes aos softwares que estiverem sendo avaliados em relação a aspectos de qualidade.

6 Reuniões

O PDAS apresenta diversas reuniões. Duas delas são reuniões que ocorrem nas fases de Planejamento e Encerramento do macro processo principal. As *Sprints* ocorrem na fase de desenvolvimento e, em cada *Sprint* deverão ocorrer duas reuniões: uma para planejar a *Sprint* e outra para encerrar.

Essa seção visa descrever cada uma dessas reuniões.



6.1 Reuniões de Planejamento

As reuniões de planejamento são talvez as reuniões mais importantes do processo pois são nelas que as regras do jogo serão definidas e compreendidas por todos os atores antes de se iniciar o desenvolvimento do software propriamente dito.

Nessas reuniões, uma série de artefatos mínimos deverão ser elaborados para se seguir para a fase de desenvolvimento.

As reuniões de planejamento são apresentadas em um processo próprio chamado “Realizar Reuniões de Planejamento”, que por sua vez é dividido em dois subprocessos:

1. Processo “Realizar Reuniões entre TI e Negócios”
2. Processo “Realizar Reuniões entre CLDF e Fábricas”

O leitor já deve ter compreendido que existem dois tipos de reuniões que devem acontecer: o primeiro tipo envolve apenas integrantes das áreas internas da CLDF. E o segundo tipo vai envolver essas áreas e as fábricas.

6.1.1 Realizar Reuniões entre TI e Negócios

As reuniões entre as áreas de TI e Negócios existem para que os seguintes artefatos sejam elaborados, de preferência nesta ordem:

1. **Documento de Definição de Líder Técnico e Líder de Negócios** - O primeiro artefato que deve ser elaborado é o Documento de Definição de Líder Técnico e Líder de Negócios descrito na seção 7.2.1, ou seja, as áreas de TI e de negócios devem realizar tantas reuniões quanto forem necessárias para definir esses atores.
2. **Documento de Visão** - Em seguida esses líderes devem se reunir para elaborar o “Documento de Visão”, artefato descrito na seção 7.1.1.
3. **Product Backlog inicial** - Finalmente, os líderes já podem começar a desenvolver um *Product Backlog* inicial de acordo com o descrito na seção 7.1.2 criando e adicionando “Histórias de



“Usuário” (seção 7.1.4) a ele. Nesse momento, é importante já configurar o ambiente de gerenciamento das *Sprints* que será utilizado (Redmine, etc.).

Assim, quando for o momento de envolver as fábricas, os líderes já terão uma compreensão mais consistente do projeto que irão desenvolver.

6.1.2 Realizar Reuniões entre CLDF e Fábricas

Após as reuniões internas e a criação dos artefatos enumerados na seção anterior, os líderes já possuem uma visão mais consistente do projeto que pretendem desenvolver. Dessa forma, chega a hora de realizar reuniões com as fábricas. Assim, os líderes devem convocar os representantes das fábricas para reuniões com os seguintes objetivos:

6.1.2.1 Objetivo: Apresentar Líderes CLDF e Representantes Fábricas

Inicialmente os atores devem se apresentar e se conhecer. Dessa forma o objetivo é apresentar quem serão os Líderes Técnico e de Negócios da CLDF e quem são os Representantes das Fábricas de Software e Qualidade

6.1.2.2 Objetivo: Apresentar este MDS e o PDAS

Uma cópia deste documento já pode ter sido enviada para as Fábricas tomarem conhecimento do processo esperado antes mesmo da reunião.

6.1.2.3 Objetivo: Apresentar o “Documento de Visão” do projeto que será desenvolvido

O “Documento de Visão” deve ser apresentado para os representantes das Fábricas de Software e Qualidade.

6.1.2.4 Objetivo: Elaborar e aprovar o “Documento de Definição de Artefatos”

Naturalmente, um projeto mais complexo terá artefatos que um projeto mais simples não precisa ter. Dessa forma, diferentes projetos podem ter diferentes conjunto de artefatos listados no **Documento de Definição de Artefatos** elaborado nessas reuniões.

Os líderes e as fábricas precisam definir os artefatos que serão necessários em cada fase do projeto (planejamento, desenvolvimento e encerramento). As Fábricas de Software e de Qualidade poderão propor artefatos diversos, mas cabe aos Líderes Técnico e de Negócios definir quais serão exigidos ou não e quem são os responsáveis pela elaboração desses artefatos.

A lista de artefatos deve ser formalizadas no “Documento de Definição de Artefatos” que explicita que artefatos serão exigidos em cada fase do processo e quem são os responsáveis por sua elaboração. A seção “Artefatos sugeridos” (seção 7.3) apresenta uma série de artefatos que podem ser utilizados nos projetos.

6.1.2.5 Objetivo: Elaborar e aprovar a “Forma da Entrega” dos entregáveis

A “Forma das Entregas” é apresentada na seção 7.2.3. Trata-se de um acordo entre CLDF e as Fábricas da forma que cada entregável deve ser entregue pelos responsáveis por sua elaboração de forma que os líderes possam fazer o ateste técnico.

Recomenda-se a forma de entrega de cada artefato seja formalizada no mesmo “Documento de Definição de Artefatos” elaborado anteriormente.



6.1.2.6 Objetivo: Elaborar e aprovar a “Definição de Pronto”

A Definição de Pronto é apresentada com detalhes na seção 7.1.5. Trata-se de um documento que deve ser elaborado em conjunto com as fábricas de forma que todos tenham o mesmo entendimento do que significa um trabalho realizado durante a fase de desenvolvimento (as *Sprints*) estar “Pronto”.

6.1.2.7 Objetivo: Determinar a “Duração da Sprint”

Finalmente, a duração da *Sprint* deve ser acordada entre os envolvidos. Recomenda-se que seja escolhida a duração de 2 ou 3 semanas. Projetos mais complexos podem necessitar de *Sprints* de 3 semanas enquanto projetos mais simples podem ser realizados em *Sprints* de menor duração.

6.1.2.8 Objetivo: Amadurecer o *Product Backlog*

O *Product Backlog* inicial elaborado nas reuniões internas anteriores (entre as Áreas de TI e Negócios) deve ser apresentado para a Fábrica de Software e esta deve colaborar com seu desenvolvimento propondo novos itens e “Histórias de Usuário”. Esses itens serão avaliados e priorizados pelo Líder de Negócios. Dessa feita, quando o processo seguir para a fase de desenvolvimento, o *Product Backlog* já apresentará “Histórias de Usuário” consistentes priorizadas que poderão ser desenvolvidas.

6.2 Reuniões de Desenvolvimento

As Reuniões de Desenvolvimento são as que ocorrem dentro das *Sprints*. Uma vez que uma *Sprint* começa, sua duração é fixada e não pode ser reduzida ou aumentada. Os eventos restantes podem terminar sempre que o propósito do evento é alcançado, garantindo que uma quantidade adequada de tempo seja gasta sem permitir perdas no processo.

6.2.1 Reunião de Planejamento da *Sprint*

Sempre antes de cada nova *Sprint* deve ocorrer a Reunião de Planejamento da *Sprint* onde o Time *SCRUM* deve realizar o planejamento do que deverá ser entregue ao final do ciclo da *Sprint* que está prestes a ser realizada.

Essa reunião é dividida em 2 fases: Fase Inicial e Fase Final cada uma com no máximo 4 horas de duração.

6.2.1.1 Fase Inicial

A fase inicial deve responder à pergunta: **O que pode ser “Pronto” nesta *Sprint*?**

Para responder a essa pergunta, sugere-se a seguinte ordem de ações na reunião a serem desempenhadas por cada ator:

1. O Líder de Negócios (*Product Owner*) auxiliado pelo Líder Técnico deve definir a Meta da *Sprint* e expor ao Time de Desenvolvimento os itens mais prioritários do *Product Backlog*.
2. O Time de Desenvolvimento deve estimar os itens em tamanho e definir o que acredita que pode ser implementado dentro da *Sprint*. Essa seleção é chamada de *Selected Product Backlog*.



6.2.1.2 Fase Final

A fase final deve responder à pergunta: **Como o trabalho necessário para entregar o incremento será realizado?**

Da mesma forma, sugere-se a seguinte ordem de ações:

1. O Time de Desenvolvimento colhe mais detalhes do *Selected Product Backlog* decompondo-os em tarefas gerando assim o *Sprint backlog*.
2. Após isso, cada membro do time de desenvolvimento deve selecionar as atividades que deseja executar na *Sprint* e estimá-la em horas.

6.2.2 Reunião Diária

É um evento de 15 minutos, para que o Time de Desenvolvimento possa sincronizar as atividades e criar um plano para as próximas 24 horas. É uma reunião onde cada participante fala sobre o progresso conseguido, o trabalho a ser realizado e/ou o que o impede de seguir avançando. É mantida no mesmo horário e local todos os dias para reduzir a complexidade. O Scrum Master assegura que o Time de Desenvolvimento tenha a reunião, mas o Time de Desenvolvimento é responsável por conduzir a Reunião Diária.

6.2.3 Reunião de Encerramento da Sprint

Nesta reunião, com duração máxima de 4 horas, deve-se realizar uma reunião com o propósito de verificar os resultados da *Sprint* e fazer uma retrospectiva dos aspectos que podem ser melhorados na próxima *Sprint*.

De forma resumida, sugere-se que no mínimo as seguintes ações sejam realizadas:

1. O Time de Desenvolvimento deve apresentar todos os resultados para o *Product Owner*;
2. O *Product Owner* avalia se a Meta da Sprint foi ou não atingida.

No entanto, essa reunião também destina-se a motivar, obter feedback e promover a colaboração entre todas as partes envolvidas. Dessa forma, a Reunião de Encerramento da Sprint também pode conter os seguintes elementos:

- Além do Time Scrum, os *Stakeholders* chave convidados pelo *Product Owner* podem ser incluídos para participar dessa reunião;
- O Time de Desenvolvimento discute o que foi bem durante a Sprint, quais problemas ocorreram dentro da Sprint, e como estes problemas foram resolvidos;
- O grupo todo colabora sobre o que fazer a seguir, e é assim que a Revisão da Sprint fornece valiosas entradas para o Planejamento da Sprint subsequente;
- O *Product Owner* pode fazer anotações que podem se tornar novos itens no *Product Backlog*.

6.3 Reunião de Encerramento

A reunião de encerramento deve ocorrer no momento em que for decidido que não haverão mais *Sprints*. Nessa reunião as Fábricas deverão entregar quaisquer artefatos de encerramento que tiverem sido definidos nas reuniões de planejamento realizadas no início do projeto. Geralmente, uma única



reunião será necessária para isso, mas caso no ateste técnico se verifique que ainda há pendências, outras reuniões podem ser convocadas para realizar o encerramento formal.

A reunião de encerramento também é o momento em que os Líderes podem solicitar atividades de encerramento como treinamentos e etc.

7 Artefatos

O *Scrum Guide* (2017) informa que os artefatos são projetados para maximizar a transparência das informações chave de modo que todos tenham o mesmo entendimento dos artefatos.

Dessa forma, o *Scrum* exige que alguns artefatos mínimos sejam utilizados. Por exemplo, não é possível dispensar o *Product Backlog*. Por outro lado, cada projeto pode necessitar de outros artefatos para alinhar os entendimentos. Por exemplo, um determinado projeto pode estabelecer determinados padrões de usabilidade que o sistema deverá atender.

Desse feita, essa seção divide os artefatos em tipos e depois apresenta cada um deles. Os tipos de artefatos são:

- **Artefatos originais Scrum:** São artefatos obrigatórios do *framework Scrum*;
- **Artefatos mínimos do processo:** São artefatos obrigatórios utilizados no processo;
- **Artefatos sugeridos:** São artefatos que não são de uso obrigatório, a priori. Eles **podem** se tornar obrigatórios, ou seja, podem ser exigidos se assim for decidido pelos Líderes CLDF durante as reuniões de planejamento. Assim, cada novo projeto pode apresentar conjuntos de artefatos distintos de acordo com as necessidades e características específicas do projeto.

7.1 Artefatos originais Scrum

Inicialmente, apresentamos os artefatos originais utilizados no *framework Scrum* cujo uso é indispensável no PDAS dessa instituição.

7.1.1 Documento de Visão

Representa a necessidade, o que deve ser produzido e apresentado ao final do projeto. As informações para definição da visão são obtidas junto ao cliente, usuários finais, *stakeholders* e executivos. A área de negócio com apoio da área de TI elabora o documento de visão 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 de informação, descrição das características-chaves do produto, descrição dos consumidores da solução (humanos ou não humanos), descrição dos requisitos de ambientes, descrição dos requisitos da documentação, descrição dos requisitos do produto, menção de tecnologias importantes, entre outros.



Dessa forma, o documento de visão deve ser construído buscando responder as seguintes questões:

- Qual problema, oportunidade, benefícios e necessidades que este produto resolve ou aproveita?
- Quais são os objetivos específicos de negócio do produto (Objetivos)?



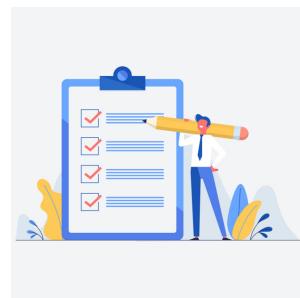
- Quais os clientes e usuários interessados na solução (Atores)?
- Como clientes e usuários poderão atingir os objetivos de negócio (Impacto)?
- Quais as características-chaves (ou macrofunções) do produto final: quais funcionalidades de negócio, performance, segurança, escalabilidade, precisam ser entregues (*Features Entregáveis*)?
- Quais os processos de negócio envolvidos nesta solução?
- Quais tarefas e atividades do processo de negócio serão automatizadas?
- Qual o ciclo de vida das entidades do negócio?
- Quais ambientes, padrões, aplicações, terá que suportar?
- Escopo e abordagem da Solução?
- Quais são os principais riscos e restrições do projeto?
- Qual a expectativa de custos e prazos?
- Quais premissas devem ser consideradas?
- Quais os diferenciais em relação à solução atual ou outra existente?

Recomenda-se utilizar as técnicas descritas no *Guia de Projetos de Software com práticas de métodos ágeis para o SISP* (MP, 2015) no que tange a atividade “Construir a Visão do Produto”.

7.1.2 Product Backlog

O *Product Backlog* contém todas as necessidades ou objetivos de negócios dos clientes do projeto e demais partes interessadas e pode também conter melhorias a serem realizadas no produto, correções de problemas, questões técnicas, pesquisas que forem necessárias etc. Assim, tudo o que pode vir a ser desenvolvido para se alcançar a Visão do Produto é adicionado como um item priorizado e ordenado no *Product Backlog*.

Os itens do *Product Backlog* são ordenados de acordo com o grau de importância de seu desenvolvimento visando, sempre, garantir e maximizar o retorno sobre o investimento (ROI) realizado por eles no projeto.



O *Product Backlog* está em constante evolução e, assim, nunca está terminado ou completo. Conforme o produto evolui, o *Product Backlog* é frequentemente modificado com a adição, subtração, reordenamento e modificação de seus itens. O produto evolui à medida que o ambiente muda e à medida que tanto clientes quanto Time de Desenvolvimento vão conhecendo e entendendo melhor o produto que está sendo construído.

7.1.3 Sprint Backlog



É uma lista de itens selecionados do alto do *Product Backlog* para o desenvolvimento do Incremento do Produto no *Sprint* (o quê), adicionada de um plano de como esse trabalho será realizado (como).

O plano de como os itens do *Sprint Backlog* serão desenvolvidos é geralmente expresso por um conjunto de tarefas correspondente a cada item, além da indicação do andamento de cada tarefa, ou seja, se ela ainda não foi iniciada, se está em andamento ou se já está terminada.



O *Sprint Backlog* pertence ao Time de Desenvolvimento, que é responsável por seu uso e manutenção. Ele deve refletir o momento atual e, para isso, o Time de Desenvolvimento o atualiza sempre que houver qualquer mudança.

7.1.4 História de usuário

História de usuário é uma descrição concisa de uma necessidade do usuário do produto (ou seja, de um “requisito”) sob o ponto de vista desse usuário. A história de usuário busca descrever essa necessidade de uma forma simples e leve. Os detalhes de negócios podem ser documentados de diferentes formas e anexados à história de usuário. Questões técnicas (como de refatoração de código ou população de um banco de dados, por exemplo), de pesquisa ou de correção de problemas dificilmente podem ser descritas sob a perspectiva do usuário. Embora pertençam ao *Product Backlog*, não devem, assim, ser representadas por história de usuário.

Nas próximas seções serão apresentados o formato clássico de uma história de usuário. Em seguida serão descritos um conjunto de sugestões para se escrever boas histórias de usuário. Finalmente, vamos descrever como essa ferramenta deve ser utilizada na prática.

7.1.4.1 Formato Clássico

Uma história de usuário clássica geralmente responde às perguntas:

- Porque?
- Quem?
- O quê?

Dessa forma, geralmente seguem o seguinte formato:

“Eu, no papel de... (para quem estamos fazendo) quero.... (o que de fato precisa ser feito) para... (o porquê da história)”

7.1.4.2 Sugestões para produzir boas histórias de usuário

Brasileiro (2019) apresenta sugestões do que deve ser feito para se produzir uma boa história de usuário:

1. **Focar no cliente ou usuário final** - o foco deve estar no que o usuário final deseja. Por isso a história é de “usuário”. Portanto, identificar quem será o usuário final é fundamental para que as histórias sejam escritas para atender aos interesses dele.
2. **Utilizar personas** - Personas são personagens que irão representar o usuário final, ou seja, deve-se criar um ou mais personagens que tenham as mesmas características dos usuários e a partir dessas definições levantar os desejos dessas personas.



3. **Histórias de Usuário são informais** - Histórias de Usuário são informais e servem tão somente para representarem os desejos dos usuários que vão utilizar os sistemas. Elas devem ser tratadas como conversas informais escritas no papel. Não confundir história de usuário com a documentação. Cada uma serve para uma coisa diferente.
4. **Deve representar o valor de negócio** - Uma história de usuário deve ter um motivo para existir. Assim, ao se escrever histórias de usuário, além das perguntas “o que?” e “como?” não pode-se deixar de responder a pergunta: “Porque” ou “Para que” essa história existe?

5. Deve atender aos critérios INVEST

História de Usuário devem atender aos critérios INVEST abaixo descritos:

- **Independente:**

As Histórias de Usuário devem ser a mais independente ou desacoplada possível umas das outras, ou seja, história de usuário com grande número de dependências em outras histórias de usuário devem ser evitadas. Essa independência visa ser viável alterar livremente a ordem que serão desenvolvidas e, ao fazê-lo, não ser necessário alterar suas estimativas.

Cabe adicionar que uma história de usuário se traduzirá sempre em uma funcionalidade de ponta a ponta, que representa valor para o usuário. Além disso, deve ser possível entender uma história de usuário sem ser necessário ler quaisquer outras.

- **Negociável e negociada:**

Seus detalhes serão discutidos, negociados e definidos com o *Product Owner* e o Time de Desenvolvimento.

- **Valiosa:**

Devem representar valor de negócio para os clientes do projeto. Ao dividir-se uma história de usuário, o resultado dessa divisão deve ser histórias menores que também representem funcionalidades de ponta a ponta, e não partes de trabalho técnico.

- **Estimável:**

O Time de Desenvolvimento deve possuir detalhes suficientes — tanto técnicos quanto de negócios — para estimar o trabalho de se transformar a história de usuário em parte do produto, de forma que o *Product Owner* possa ordená-la apropriadamente.

- **Pequena:**

Pequena ou (“small”, em inglês) ou dimensionada apropriadamente: apenas história de usuário pequenas e com um bom nível de detalhes podem ser colocadas em desenvolvimento.

- **Testável:**

Deve ser possível verificar e confirmar que a história de usuário está pronta, ou seja, que foi transformada em parte do produto e está funcionando conforme esperado. A verificação é realizada por meio dos Critérios e Testes de Aceitação.

6. Não confiar apenas nas histórias

Histórias de usuário são informais e não vão substituir a documentação, mas devem ser completadas com tudo o que for necessário e que agregue valor para o time de desenvolvimento.

7. Identificar um “Épico” e criar histórias a partir dele



Épicos são grandes histórias que geralmente não atendem ao critério INVEST de independência já que elas dependem de uma ou mais histórias. Após identificar um épico, as histórias de usuário são mais facilmente elaboradas a partir dele. Tome como exemplo, o seguinte épico: **“Como usuário gostaria de pagar minha compra para poder receber os produtos”**. Essa história é um Épico pois para o usuário realizar o pagamento ele terá que realizar outras ações como, escolher a forma de pagamento, selecionar o cartão de crédito, informar se vai dividir ou não, e para receber o produto, uma série de outras ações devem ser realizadas por diversas pessoas.

Para ilustrar, apresentamos alguns exemplos de História de Usuário para o desenvolvimento fictício de um sistema de controle de inscrição em cursos:

Histórias de Usuário - Sistema de Controle de Inscrições (exemplo)

- Eu como usuário, quero consultar cursos e turmas disponíveis, para poder escolher qual turma cursar.
- Eu como usuário, quero poder me inscrever no curso que escolhi, para poder realizar o treinamento.
- Para gerenciar as inscrições no sistema, eu no papel de administrador, desejo poder confirmar os inscritos nas turmas.
- Para gerenciar as inscrições no sistema, eu no papel de administrador, desejo poder trocar os inscritos de turma.

7.1.4.3 Histórias de usuário na prática

Com a popularização da metodologia *Scrum*, outros elementos passaram a acompanhar as histórias de usuário como, por exemplo, critérios de aceitação e propriedades como prioridade, tamanho, etc. No entanto, pela falta de um termo melhor, as pessoas continuaram chamando este conjunto de elementos (história, critérios, propriedades, etc) com o mesmo nome de “História de Usuário”.

Dessa forma, na prática, sugere-se que uma História de Usuário seja acompanhada de outros elementos a saber:

- | | | |
|-------------|--------------------------|--------------|
| • Alias | • Critérios de Aceitação | • Prioridade |
| • Narrativa | • Conversação | • Tamanho |

Vamos descrever cada elemento:

- **Alias:**

Em computação, alias é um apelido, ou seja, trata-se de uma descrição curta da história usada para referenciá-la de forma simples e intuitiva.

- **Narrativa:**

Trata-se da descrição da história de usuário propriamente dita de acordo com o formato clássico e as sugestões apresentadas nas seções anteriores.

- **Critérios de Aceitação:**



HISTÓRIA DE USUÁRIO		
ALIAS: APELIDO		
Prioridade:	[1-5]	Tamanho:
X PFs		
NARRATIVA		
Eu Papel Desejo/Quero/Preciso NECESSIDADE DO NEGÓCIO Porque/Para BENEFÍCIO ESPERADO		
CONVERSAÇÃO		
Informações coletadas durante as conversas com o Dono do Produto, requisitos não funcionais entre outros que poderão originar novas HISTÓRIAS DE USUÁRIOS.		
CRITÉRIOS DE ACEITAÇÃO		
Dado que PREMISSAS [e] Quando AÇÃO		

HISTÓRIA DE USUÁRIO	
Alias: Aluno realiza inscrição no curso	
Prioridade:	4
Tamanho: 10	
NARRATIVA	
Eu no papel de ALUNO Desejo/Quero/Preciso REALIZAR A MINHA INSCRIÇÃO EM UM CURSO Porque/Para REALIZAR O TREINAMENTO EM CUMPRIMENTO AO PLANO DE CAPACITAÇÃO DA MINHA UNIDADE.	
CONVERSAÇÃO	
Deverá ser gerado um código de inscrição pelo instrutor do curso que será encaminhado aos alunos. Quando o aluno realizar a inscrição o sistema deverá notificar o instrutor.	
CRITÉRIOS DE ACEITAÇÃO	
Dado que TENHO O PERFIL DE ALUNO E TENHA RECEBIDO O CÓDIGO DE INSCRIÇÃO EM UM CURSO Quando ACIONO O COMANDO REALIZAR INSCRIÇÃO Então O SISTEMA DEVERÁ INCLUIR O MEU USUÁRIO COMO ALUNO DO CURSO E NOTIFICAR O INSTRUTOR	

(a) Sugestão

(b) Exemplo

Figure 6: Sugestão e exemplo de História de Usuário na prática

É uma descrição dos testes e critérios usados para aceitação do incremento do produto. Geralmente, apresentam o seguinte formato:

“Dado que PREMISSAS

[e]

Quando AÇÃO

Então RESULTADO”

• **Conversação:**

Informações coletadas durante as conversas com o *Product Owner*, requisitos não funcionais, dentre outras observações, que poderão originar novas histórias de usuário.

• **Prioridade:**

Medida de aferição de prioridade. Sugere-se que seja um número inteiro entre 1 (maior prioridade) e 5 (menor prioridade) usado para priorizar essa história;

• **Tamanho:**

Tamanho estimado em pontos de função.

Para ilustrar, apresentamos a figura 6 com uma sugestão de História de Usuário na prática (a) e um exemplo de história preenchida (b).

7.1.5 Definição de Pronto

A Definição de “Pronto” é um acordo formal entre *Product Owner* e o Time de Desenvolvimento sobre o que é necessário para se considerar que um trabalho realizado na Sprint está pronto. São, portanto, critérios definidos por ambos para garantir a transparência, por meio da compreensão compartilhada do que significa quando o Time de Desenvolvimento afirma que qualquer item ou o Incremento do Produto está pronto.



Exemplo de uma Definição de “Pronto”

Somente consideraremos o incremento do produto pronto se estiver:

- Codificado
- Passando nos testes unitários
- Passando nos testes de aceitação
- Integrado ao sistema ABCD
- Com o manual do usuário atualizado

Definição de Pronto para projetos de desenvolvimento e suporte de sistemas:

- Todas as funcionalidades do backlog da sprint ou da release foram implementadas e estão livres de erros ou defeitos impeditivos;
- Elaboração ou atualização dos artefatos previstos para cada fase do processo;
- Sistema, software ou aplicativo disponível funcionalmente no ambiente de homologação, ou outro definido pela Contratante;
- Conformidades com os guias ou documentos de apoio (arquitetura, padrão visual, política de administração de dados);
- A qualidade do código satisfaz aos indicadores preestabelecidos e foram aprovados pela Contratante, conforme Relatório de Qualidade e de Segurança;
- Artefatos corretamente identificados quanto à autoria e versionamento em repositório da Contratante;
- Funcionalidade validada e aprovada pelo *Product Owner*;
- Release ou Sprint validada pelo Fiscal do contrato.

Definição de pronto para projetos de documentação de sistemas:

- Artefatos escritos de forma adequada e, se conter requisitos, que esses possibilitem identificar se são mensuráveis, em termos de função transacional e de dados;
- Requisitos adequados materialmente (não-ambiguidade, não-redundância, ausência de conflitos entre requisitos ou histórias de usuários, não-divisibilidade, ausências de inconsistências, documento claro e coeso, requisitos rastreáveis até o nível do *backlog* do produto e da *release* e requisitos testáveis);
- Artefatos padronizados em *templates* definidos pela Contratante;
- Artefatos corretamente identificados quanto à autoria e versionamento em repositório da Contratante;
- Artefato validado e aprovado pelo dono do produto, se for o caso;
- Artefato validado pelo Líder de projeto e pelo Fiscal do contrato



7.1.6 Incremento do Produto

Em cada Sprint do projeto, o Time de Desenvolvimento trabalha nos itens selecionados para o Sprint Backlog, do mais importante para o menos importante, visando atingir a Meta do Sprint. O Incremento do Produto é o resultado desse trabalho, ou seja, é a soma de todos os itens completos no Sprint.

O Time de Desenvolvimento demonstra o Incremento do Produto para os clientes do projeto ao final de cada Sprint, na reunião de Sprint Review. O principal objetivo dessa reunião é obter feedback dessas pessoas sobre o trabalho realizado. O Incremento do Produto deve representar valor visível para eles, ou seja, funcionalidades que podem ser experimentadas e utilizadas.

7.2 Artefatos mínimos do processo

7.2.1 Documento de Definição de Líder Técnico e Líder de Negócios

Documento que formaliza, para determinado projeto, quem da Área de TI da CLDF fará o papel de Líder Técnico e quem da Área de Negócios da CLDF fará o papel de Líder de Negócios. Esse documento deve ser elaborado nas reuniões de planejamento.

7.2.2 Documento de Definição de Artefatos

Documento que lista quais artefatos serão exigidos em cada fase do PDAS (planejamento, desenvolvimento e encerramento) e quem é o responsável por sua elaboração. Esse documento é criado nas reuniões de planejamento. A seção 7.3 apresenta uma série de artefatos que podem ser usados nos projetos em cada fase e quais seriam os responsáveis por sua elaboração.

7.2.3 Forma das Entregas

Uma entrega é um bem ou serviço tangível ou intangível produzido como resultado de um projeto que se destina a ser entregue a um cliente. Uma entrega pode ser um relatório, um documento, um produto de software, uma atualização do servidor, um pacote de produtos, ou qualquer outro componente de um projeto geral.

Durante a realização das *Sprints* o entregável mais comum é o “Incremento do Produto”, mas podem ocorrer casos em que sejam necessários entregar outros artefatos, definidos previamente, nas reuniões de planejamento.

Assim, a forma da entrega de um entregável particular deve ser definida. A pergunta que deve ser respondida é:

“*Como o entregável será entregue?*”

Por exemplo, os Líderes podem definir que a forma de entregar o “Incremento de Produto” seja por meio da instalação do software em um ambiente de homologação disparando-se um e-mail para o Líder Técnico assim que essa instalação tenha sido realizada. Dessa forma, é importante definir, para cada entregável, a forma da entrega.

7.2.4 Pacote de Produtos de Testes e Controle de Qualidade

Um pacote de produtos de testes de sistemas e controle de qualidade é um conjunto de entregáveis destinados à apuração e comprovação de que um determinado entregável atende ou não a determinados critérios de qualidade.



7.2.5 Pacote de Produtos de Medição de Sistemas

Um pacote de produtos de medição de sistemas é um conjunto de entregáveis destinados a medir a quantidade de pontos de função de um determinado produto segundo uma métrica e um sistema de medição previamente definido. Além da medição, outros relatórios auxiliares podem fazer desse pacote.

7.3 Artefatos sugeridos

Conforme foi explicado, diferentes projetos podem exigir diferentes conjuntos de artefatos. Essa seção apresenta uma lista de artefatos sugeridos categorizados de acordo com as fases em que eles podem ser usados. Também sugerimos quem seriam os responsáveis pela elaboração desses artefatos.

7.3.1 Fase de Planejamento

A tabela 1 presenta um conjunto de artefatos que poderiam ser exigidos na fase de planejamento do PDAS.

Artefatos x Responsável (Fase de Planejamento)					
ARTEFATOS	LN	LT	FS	FQ	FM
Glossário de termos técnicos do negócio	✓	—	—	—	—
Proposta de Identidade Visual	—	—	✓	—	—
Estratégia de Teste	—	—	—	✓	—
Relatório de Verificação de Requisitos	—	—	—	✓	—

Table 1: Artefatos sugeridos para a fase de planejamento

7.3.2 Fase de Desenvolvimento

Na fase de desenvolvimento do PDAS, além do “Incremento do Produto”, outros artefatos poderiam ser exigidos das Fábricas de Software e Qualidade durante a execução de suas atividades. Uma lista de artefatos sugeridos são apresentados na tabela 2.

Dentre os diversos artefatos sugeridos, alguns merecem destaque e são melhor apresentados na seções a seguir:

7.3.2.1 Gráficos de Acompanhamento do Trabalho

Os Gráficos de Acompanhamento do Trabalho são ferramentas para a visualização do progresso do cumprimento de uma quantidade mensurável estimada de trabalho em um determinado tempo. Eles servem para criar visibilidade de uma forma simples, ajudando a rapidamente identificar problemas e, assim, reduzir os riscos do projeto.

Os Gráficos de Release Burndown e de Sprint Burndown são os dois tipos de Gráficos a serem utilizados.



Artefatos x Responsável (Fase de Desenvolvimento)					
ARTEFATOS	LN	LT	FS	FQ	FM
Gráficos de Acompanhamento do Trabalho	—	—	✓	—	—
Relatório Gerencial	—	—	✓	—	—
Documentos de Regras de Negócio	—	—	✓	—	—
Código fonte	—	—	✓	—	—
Código compilado e/ou executável	—	—	✓	—	—
Pacote com testes unitários e de integração	—	—	✓	—	—
Evidências de Testes	—	—	✓	—	—
Modelo de Entidade de Relacionamento	—	—	✓	—	—
Planilha de Contagem Detalhada	—	—	✓	—	—
Manual do sistema	—	—	✓	—	—
Manual do usuário	—	—	✓	—	—
Ajuda do sistema	—	—	✓	—	—
Cenários e Casos de Teste	—	—	—	✓	—
Roteiro de Teste	—	—	—	✓	—
Relatório de Não Conformidade (Evidência de Defeito)	—	—	—	✓	—
Evidência de Teste	—	—	—	✓	—

Table 2: Artefatos sugeridos para a fase de desenvolvimento

Esses gráficos têm como objetivo mostrar o esforço restante para a conclusão da Sprint/Release, bem como mostrar o quanto próximo ou distante a equipe está de atingir a meta da Sprint ou do release.

Neste gráfico, tem-se uma coluna vertical, que representa a quantidade de esforço (quantidade de trabalho que ainda precisa ser realizada) e uma coluna horizontal, representando as unidades de tempo, como, por exemplo, as horas ou dias para finalizar uma Sprint.

Conforme a figura abaixo, uma linha na cor verde indica o fluxo ideal de trabalho. Quando o gráfico está acima da linha verde, a equipe de desenvolvimento está distante de atingir a meta. Em cima da linha verde, a equipe de desenvolvimento está em um fluxo de trabalho ideal.

7.3.2.2 Relatório Gerencial

Documento que contém a lista de artefatos elaborados na Release e é criado na fase de Preparação e atualizado a cada encerramento de Sprint e/ou de Release.

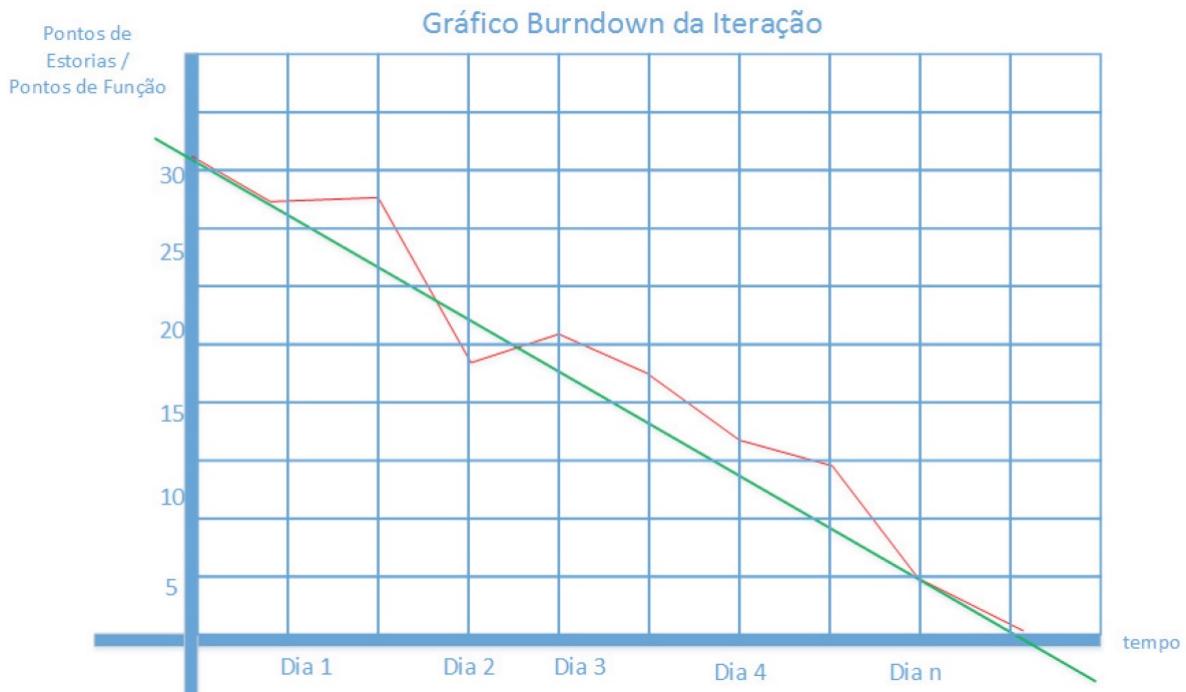


Figure 7: Gráfico Burndown da *Sprint*

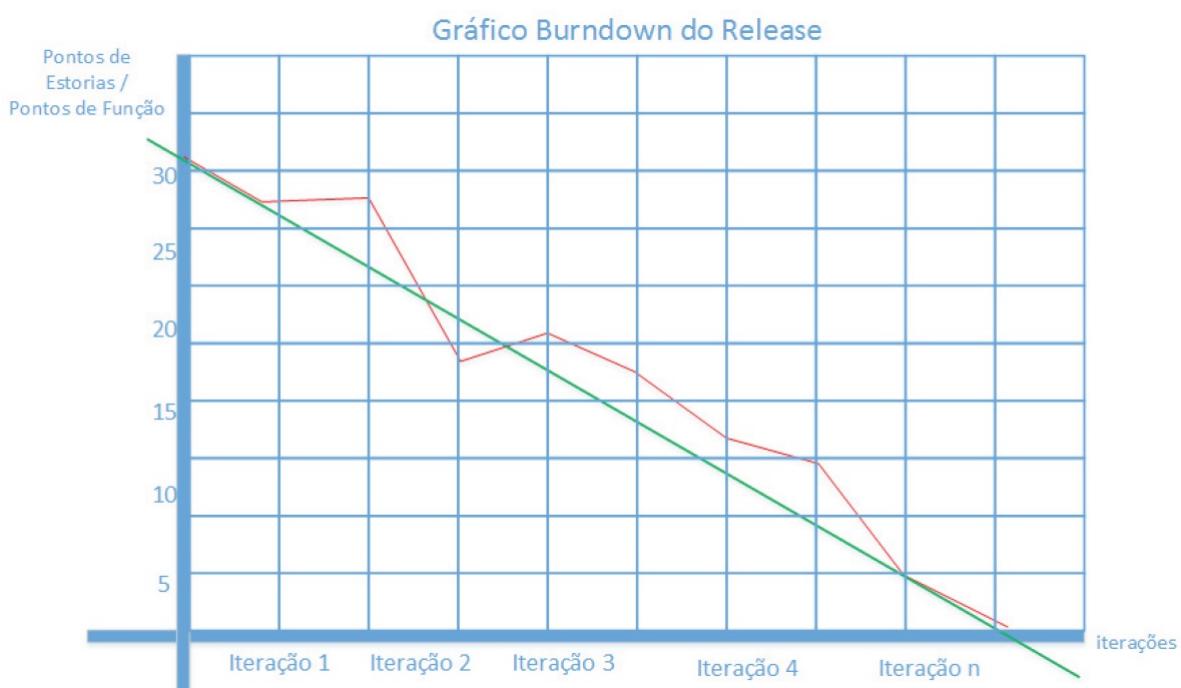


Figure 8: Gráfico Burndown da *Release*



Recomenda-se que um relatório gerencial inclua os seguintes artefatos:

- Backlog do Produto priorizado e segregado em Sprints;
- Sprint Backlog e Definição de Pronto (DoD);
- Histórias de Usuários detalhadas;
- Regras de Negócio;
- Protótipo;
- Atas de Reunião;
- Documento de Arquitetura;
- Modelo de Dados;
- Plano de Implantação;
- Relatório de Testes;
- Relatório de Qualidade de Código;
- Relatório de Segurança do sistema;
- Incremento de Produto (se for o caso);
- Manual do Usuário (se for o caso);
- Documento de Lições Aprendidas na Sprint e na Release;

7.3.3 Fase de Encerramento

A reunião de encerramento do projeto é o momento ideal para exigir alguns outros artefatos das fábricas. Exemplo de artefatos que poderiam ser exigidos são apresentados na tabela 3.

Artefatos x Responsável (Fase de Encerramento)					
ARTEFATOS	LN	LT	FS	FQ	FM
Plano de Treinamento	—	—	✓	—	—
Material Didático de Treinamento	—	—	✓	—	—
Relatório de Lições Aprendidas	—	—	✓	—	—
Sumário de Testes	—	—	—	✓	—

Table 3: Artefatos sugeridos para a fase de encerramento

7.3.4 Artefatos de Medição de Sistemas

O Processo “Executar Serviço de Medição de Sistema” deverá ocorrer em diversos momentos durante a fase de desenvolvimento do software. Assim, alguns artefatos de medição poderão ser exigidos da Fábrica de Métricas. Esses artefatos são apresentados na tabela 4.

7.3.5 Artefatos adicionais

Alguns artefatos podem ser utilizados em todas as fases do processo. É o caso do “Termo de Aceite de Fase” e das atas de reunião.

7.3.5.1 Termo de Aceite da Fase (TAF)

O termo de aceite da fase é um documento que formaliza o aceite de uma fase. Trata-se de um marco que encerra a fase atual e inicia a fase seguinte.



Artefatos x Responsável (Medição de Sistemas)					
ARTEFATOS	LN	LT	FS	FQ	FM
Planilha de Contagem	—	—	—	—	✓
Sumário de Contagem	—	—	—	—	✓

Table 4: Artefatos sugeridos para acompanhar os processos de medição dos sistemas

7.3.5.2 Atas de Reunião

Uma ata de reunião, ou simplesmente ata, é um registro dos eventos importantes que ocorreram em uma reunião deliberativa. São tradicionalmente feitos por meio de escrita e durante o andamento da própria reunião, simultaneamente à ocorrência desses eventos. Recomenda-se que as deliberações realizadas nas reuniões sejam formalizadas por meio de atas de reunião.

8 Detalhes do Processo

8.1 Processo de Desenvolvimento Ágil de Software

O PDAS apresenta processos e subprocessos semelhantes que “se repetem” nas três fases do fluxo principal: planejamento, desenvolvimento e encerramento. Por exemplo, o Subprocesso “Realizar Ateste Técnico” ocorre nessas três fases. No entanto, o leitor deve entender que embora o processo seja o mesmo, as situações em que ele é utilizado são distintas. O que é atestado na fase de planejamento é diferente do que é atestado na fase de desenvolvimento. Assim, para simplificar, processos como este, que ocorrem várias vezes em situações diferentes mas que apresentam a mesma função, foram apresentados uma única vez neste documento.



8.1.1 Evento: “Recebimento de Demanda”

Evento: “Recebimento de Demanda”	
 Recebimento de Demanda	Objetivo: Este é um evento de início. Ele significa que a “vontade” de iniciar um novo projeto, seja ele um sistema novo ou a manutenção de um software pré-existente, de acordo com a classificação de demandas, “manifestou-se” na Área de TI. Não importa onde essa demanda “nasceu”. O que importa é que a Área de TI recepcionou essa demanda e as condições para iniciar o desenvolvimento são satisfeitas.
PARTICIPANTES	⇒ Área de TI ⇒ Área de Negócios
ENTRADAS	⇒ PEI ⇒ PDTI ⇒ Contratos ⇒ Demanda Interna
SAÍDAS	⇒ Convocação de Reuniões de Planejamento
⇐ ATIVIDADE ANTERIOR PRÓXIMA ATIVIDADE ⇒	
—	Subprocesso “Realizar Reuniões de Planejamento”

Outras informações:

Sugere-se que primeiro sejam convocadas reuniões internas entre a Área de TI e a Área de Negócios e depois sejam realizadas reuniões com as Fábricas. O processo de realização de reuniões é detalhado no Subprocesso “Realizar Reuniões de Planejamento” (seção 8.4).

A convocação de uma reunião geralmente é formalizada por meio de um e-mail destinado aos interessados. Mas pode ocorrer de outras formas como, por exemplo, um memorando.



8.1.2 Subprocesso: “Realizar Reuniões de Planejamento”

Subprocesso: “Realizar Reuniões de Planejamento”	
 Realizar Reuniões de Planejamento	Objetivo: Definir formalmente os Líderes CLDF (Técnico e de Negócios), definir os artefatos que serão usados e exigidos nas fases de planejamento, desenvolvimento e encerramento e elaborar os artefatos de planejamento.
PARTICIPANTES	⇒ Área de TI ⇒ Área de Negócios ⇒ Fábrica de Software ⇒ Fábrica de Qualidade
ENTRADAS	⇒ PEI ⇒ PDTI ⇒ Demandas Internas ⇒ Contratos
SAÍDAS	⇒ Definição dos Líderes ⇒ Documento de Visão ⇒ <i>Product Backlog</i> inicial ⇒ Definição de artefatos para cada fase ⇒ Artefatos de Planejamento ⇒ Atas de Reunião
== ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ==
Evento de Início ou Sanções	Subprocesso “Realizar Ateste Técnico”

Outras informações:

A seção 6.1 apresenta um detalhamento melhor de como deve ser feito essas reuniões e os resultados esperados delas. Este processo pode ser expandido em detalhes conforme seção 8.4.



8.1.3 Subprocesso: “Realizar Ateste Técnico”

Subprocesso: “Realizar Ateste Técnico”	
	Objetivo: As saídas do subprocesso anterior são avaliadas e validadas.
Realizar Ateste Técnico	
PARTICIPANTES	⇒ Líder Técnico ⇒ Fábrica de Qualidade ⇒ Fábrica de Métricas
ENTRADAS	⇒ Entregáveis ⇒ Pacotes de Produtos ⇒ Atas de Reunião
SAÍDAS	⇒ Decisão de Validação
== ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ==
⇒ Subprocesso “Realizar Reuniões de Planejamento” ou ⇒ Subprocesso “SPRINT” ou ⇒ Subprocesso “Realizar Reuniões de Encerramento”	Decisão: “Ateste Técnico foi Validado?”

Outras informações:

Este processo é expandido em subprocessos de acordo o detalhamento apresentado na seção 8.3.



8.1.4 Decisão: “Validado?”

Decisão: “Validado?”	
	Objetivo: Decidir, com base nos resultados da fase, se os entregáveis aderem aos requisitos de entrada e aos planos e regras definidos.
Validado?	
PARTICIPANTES	⇒ Líder Técnico ⇒ Líder de Negócios
ENTRADAS	⇒ Resultados do Processo “Realizar Ateste Técnico”
SAÍDAS	SIM ⇒ Processo “Realizar Aceitação da Fase” NÃO ⇒ Processo “Verificar Aplicação de Sanções”
← ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE →
Subprocesso “Realizar Ateste Técnico”	VER SAÍDAS



8.1.5 Processo: “Verificar Aplicação de Sanções”

Processo: “Verificar Aplicação de Sanções”	
	Objetivo: Analisar os desvios gerados na fase e decidir sobre a aplicação de sanções e/ou encaminhamento de demandas de correção.
Verificar Aplicação de Sanções	
PARTICIPANTES	⇒ Líder Técnico ⇒ Líder de Negócios
ENTRADAS	⇒ Documentos de Validação ⇒ Contratos
SAÍDAS	⇒ Sanções e/ou Glosas
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
DECISÃO (SAÍDA NÃO)	⇒ Subprocesso “Realizar Reuniões de Planejamento” ou ⇒ Subprocesso “SPRINT” ou ⇒ Subprocesso “Realizar Reuniões de Encerramento”



8.1.6 Processo: “Realizar Aceitação da Fase”

Processo: “Realizar Aceitação da Fase”	
	Objetivo: Realizar a aceitação formal da fase de forma a prosseguir para a fase posterior do PDAS.
Realizar Aceitação da Fase	
PARTICIPANTES	⇒ Líder Técnico ⇒ Líder de Negócios ⇒ Gestor do Contrato
ENTRADAS	⇒ Documentos de Validação
SAÍDAS	⇒ Artefatos da fase
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
DECISÃO (SAÍDA SIM)	⇒ Subprocesso “Sprint” ou ⇒ Decisão “Haverá outra Sprint?” ou ⇒ Final do Processo

Outras informações:

O aceite da fase pode ser formalizado com um “Termo de Aceite da Fase” caso tenha-se acordado o uso desse artefato nas reuniões de planejamento.



8.1.7 Subprocesso: “Sprint”

Subprocesso: “Sprint”	
	Objetivo: Realizar o ciclo de trabalho de desenvolvimento de um incremento do produto.
Sprint	
PARTICIPANTES	⇒ Líder Técnico ⇒ Líder de Negócios ⇒ Fábrica de Software
ENTRADAS	⇒ Sinal proveniente do aceite da fase de planejamento (subprocesso “Realizar Aceitação da Fase”).
SAÍDAS	⇒ Incremento do Produto ⇒ Outros Artefatos de Desenvolvimento
== ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ==
Processo “Realizar Aceitação da Fase (Planejamento)”	<i>Gateway</i> de ativação incondicional em paralelo: ⇒ Subprocesso “Realizar Ateste Técnico” ⇒ Decisão “Haverá outra Sprint?”

Outras informações:

O processo de desenvolvimento é dividido em ciclos regulares ao longo do tempo. Uma *Sprint* é, portanto, um ciclo de desenvolvimento onde requisitos são implementados tendo como resultado um incremento do produto “pronto”. Conforme o guia, cada *Sprint* pode ser considerada um projeto com horizonte não maior que um mês. Como os projetos, as *Sprints* são utilizadas para realizar algo. Cada *Sprint* tem uma meta do que é para ser construído, um plano previsto e flexível que irá guiar a construção, o trabalho e o produto resultante do incremento. Este processo é detalhado em subprocessos na seção 8.2.



8.1.8 Decisão: “Haverá outra Sprint?”

Decisão: “Haverá outra Sprint?”	
◆	Objetivo: Decidir se uma nova <i>Sprint</i> será necessária ou se o software será encerrado.
Haverá outra Sprint?	
PARTICIPANTES	⇒ Líder Técnico ⇒ Líder de Negócios
ENTRADAS	⇒ Resultados do Processo “Realizar Aceitação da Fase”
SAÍDAS	SIM ⇒ Subprocesso “ <i>Sprint</i> ” NÃO ⇒ Processo “Realizar Reunião de Encerramento”
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
⇒ <i>Gateway</i> de ativação incondicional em paralelo ⇒ Processo “Realizar Aceitação da Fase”	VER SAÍDAS



8.1.9 Processo: “Realizar Reunião de Encerramento”

Processo: “Realizar Reunião de Encerramento”	
	Objetivo: Entrega de artefatos de encerramento e realização de atividades pertinentes.
Realizar Reunião de Encerramento	
PARTICIPANTES	⇒ Líder Técnico ⇒ Líder de Negócios ⇒ Fábrica de Software ⇒ Fábrica de Qualidade ⇒ Fábrica de Métricas
ENTRADAS	⇒ Software desenvolvido
SAÍDAS	⇒ Artefatos de Encerramento
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Decisão: “Haverá outra Sprint?”	Subprocesso “Realizar Ateste Técnico”

Outras informações:

Vide seção 6.3 para uma descrição mais detalhada da reunião de encerramento.

8.1.10 Evento: “Final do Processo”

Evento: “Final do Processo”	
	Objetivo: Este evento indica que o processo chegou ao fim.
Final do Processo	
PARTICIPANTES	—
ENTRADAS	—
SAÍDAS	—
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Processo “Realizar Aceitação da Fase”	—



8.2 Subprocesso *Sprint*

8.2.1 Evento: “Sinal de Início”

Evento: “Sinal de Início”	
	Objetivo: Sinal de início do processo <i>Sprint</i> .
Sinal de Início	—
PARTICIPANTES	—
ENTRADAS	—
SAÍDAS	—
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
—	Processo “Realizar Reunião de Planejamento da <i>Sprint</i> ”



8.2.2 Processo: “Realizar Reunião de Planejamento da Sprint”

Processo: “Realizar Reunião de Planejamento da Sprint”	
	Objetivo: Planejar o ciclo de desenvolvimento e o escopo da <i>Sprint</i> . O time de desenvolvimento seleciona itens do <i>Product Backlog</i> com os quais se compromete a concluir, criando o <i>Sprint Backlog</i> .
PARTICIPANTES	⇒ Líder Técnico ⇒ Líder de Negócios ⇒ <i>Scrum Master</i> FS ⇒ Time Desenv. FS ⇒ Representante FQ (Opcional)
ENTRADAS	⇒ <i>Product Backlog</i>
SAÍDAS	⇒ <i>Sprint Backlog</i>
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
—	<i>Gateway</i> de ativação incondicional paralelo: ⇒ Processo “Desenvolver” ⇒ Processo “Acompanhar Sprint”

Outras informações:

Esta reunião ocorre em duas fases conforme foi descrito na seção 6.2.1.



8.2.3 Processo: “Desenvolver”

Processo: “Desenvolver”	
	Objetivo: Nesta atividade é desenvolvido o trabalho (funcionalidades) definido no <i>Sprint Backlog</i> (todo o trabalho ou o que for possível).
PARTICIPANTES	⇒ <i>Scrum Master</i> FS ⇒ Time Desenv. FS
ENTRADAS	⇒ <i>Sprint Backlog</i>
SAÍDAS	⇒ Incremento do Produto ⇒ Outros entregáveis previamente definidos
⇐ ATIVIDADE ANTERIOR ⇒	
Processo “Realizar Reunião de Planejamento da Sprint”	Processo “Entregar”



8.2.4 Processo: “Entregar”

Processo: “Entregar”	
	Objetivo: Essa atividade envolve entregar o que foi desenvolvido durante a <i>Sprint</i> para o Líder Técnico.
Entregar	
PARTICIPANTES	⇒ <i>Scrum Master FS</i> ⇒ Time Desenv. FS
ENTRADAS	⇒ Incremento do Produto ⇒ Outros entregáveis previamente definidos
SAÍDAS	⇒ Desenvolvimento entregue da forma e no formato previamente definidos em reunião
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Processo “Desenvolver”	Processo “Realizar Reunião de Encerramento da Sprint”

Outras informações:

A forma e o formato da entrega, ou seja, se a Fábrica de Software deverá instalar o software em um ambiente de homologação e outros detalhes deverão ser acordados com o Líder Técnico previamente, preferencialmente, nas reuniões de planejamento iniciais (6.1) ou na reunião de planejamento da *Sprint* (6.2.1).



8.2.5 Processo: “Acompanhar Sprint”

Processo: “Acompanhar Sprint”	
Acompanhar Sprint	Objetivo: Acompanhar a <i>Sprint</i> de modo a remover os impedimentos internos à CLDF, visando assegurar que o objetivo seja atingido.
PARTICIPANTES	⇒ Líder Técnico
ENTRADAS	—
SAÍDAS	—
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Processo “Realizar Reunião de Planejamento da Sprint”	Processo “Realizar Reunião de Encerramento da Sprint”

Outras informações:

Caberá ao Líder Técnico, durante os ciclos de desenvolvimento, tão somente acompanhar o desenrolar das atividades do *Scrum Master* e do time de desenvolvimento da Fábrica de Software visando remover quaisquer impedimentos internos ao ambiente do órgão para que o ciclo possa atingir seu objetivo.



8.2.6 Processo: “Realizar Reunião de Encerramento da Sprint”

Processo: “Realizar Reunião de Encerramento da Sprint”	
	Objetivo: Apresentar o resultado da Sprint.
Realizar Reunião de Encerramento da Sprint	
PARTICIPANTES	⇒ Líder Técnico ⇒ Líder de Negócios ⇒ <i>Scrum Master</i> FS ⇒ Time Desenv. FS ⇒ Representante FQ (Opcional)
ENTRADAS	⇒ Incremento do Produto ⇒ Desenvolvimento entregue da forma e no formato previamente definidos em reunião ⇒ Outros entregáveis
SAÍDAS	⇒ Avaliação de Atingimento da Meta da Sprint ⇒ Anotações de próximos passos a serem tomados ⇒ Anotações de itens para atualização do <i>Product Backlog</i>
== ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ==
<i>Gateway</i> de ativação incondicional paralelo: ⇒ Processo “Entregar” ⇒ Processo “Acompanhar Sprint”	Sinal de Fim

Outras informações:

A reunião de encerramento de *Sprint* é explicada na seção 6.2.3.



8.2.7 Evento: “Sinal de Fim”

Evento: “Sinal de Fim”	
	Objetivo: Sinal de finalização desta iteração de desenvolvimento.
Sinal de Fim	
PARTICIPANTES	—
ENTRADAS	—
SAÍDAS	—
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Processo “Realizar Reunião de Encerramento de Sprint”	Sinal de Início do Subprocesso “Realizar Ateste Técnico”

8.3 Subprocesso Realizar Ateste Técnico

8.3.1 Evento: “Sinal de Início”

Evento: “Sinal de Início”	
	Objetivo: Sinal de início do processo “Realizar Ateste Técnico”.
Sinal de Início	
PARTICIPANTES	—
ENTRADAS	—
SAÍDAS	—
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Sinal de Fim do Subprocesso “Sprint”	Processo “Receber Entregáveis”



8.3.2 Processo: “Receber Entregáveis”

Processo: “Receber Entregáveis”	
	Objetivo: Receber os entregáveis definidos para a fase na forma e no formato definidos para posterior análise de sua aderência aos padrões, requisitos e critérios de aceitação acordados.
PARTICIPANTES	⇒ Líder Técnico
ENTRADAS	⇒ Entregáveis
SAÍDAS	⇒ Decisão
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Processo “Receber Entregáveis”	Decisão “Todos os produtos entregues?”

Outras informações:

Cabe ressaltar que, segundo o critério de completude, o processo não prevê a aceitação parcial de entregáveis.

8.3.3 Decisão: “Todos os produtos entregues?”

Decisão: “Todos os produtos entregues?”	
	Objetivo: Verificar se todos os produtos especificados para a fase foram entregues.
Todos os produtos entregues?	
PARTICIPANTES	⇒ Líder Técnico
ENTRADAS	⇒ Entregáveis
SAÍDAS	SIM ⇒ <i>Gateway</i> de decisões: ⇒ Decisão “Será preciso medir?” ⇒ Decisão “Será preciso testar?” NÃO ⇒ Aciona evento de fim e dispara sinal para Processo “Verificar Aplicação de Sanções”
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Processo “Receber Entregáveis”	VER SAÍDAS



8.3.4 Decisão: “Será precido medir?”

Decisão: “Será precido medir?”	
◆	Objetivo: Verificar necessidade de realizar medições relacionadas às entregas da fase.
Será precido medir?	
PARTICIPANTES	⇒ Líder Técnico
ENTRADAS	⇒ Entregáveis
SAÍDAS	SIM ⇒ Processo “Solicitar Serviço de Medição” NÃO ⇒ <i>Gateway</i> de convergência ⇒ Processo “Realizar Ateste Técnico”
← ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Decisão “Todos os produtos entregues?” ⇒ SIM	VER SAÍDAS

Outras informações:

Caso a decisão de saída seja negativa, o processo entra em um *gateway* de ativação incondicional em paralelo de convergência, ou seja, ele espera todos os processos em paralelo finalizarem para seguir para o próximo processo, neste caso, o Processo “Realizar Ateste Técnico”.



8.3.5 Processo: “Solicitar Serviço de Medição”

Processo: “Solicitar Serviço de Medição”	
	Objetivo: Acionar o serviço técnico de medição de sistemas.
Solicitar Serviço de Medição	
PARTICIPANTES	⇒ Líder Técnico
ENTRADAS	⇒ Decisão “Será preciso medir?” SIM
SAÍDAS	⇒ Solicitação de Serviço de Medição de Sistema
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Decisão “Será preciso medir?”	Processo “Executar Serviço de Medição de Sistemas”

8.3.6 Processo: “Executar Serviço de Medição de Sistemas”

Processo: “Executar Serviço de Medição de Sistemas”	
	Objetivo: Executar o serviço técnico de medição de sistemas.
Executar Serviço de Medição de Sistemas	
PARTICIPANTES	⇒ Fábrica de Métricas
ENTRADAS	⇒ Solicitação de Serviço de Medição de Sistemas
SAÍDAS	⇒ Pacote de Produtos de Medição de Sistemas
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Processo “Solicitar Serviço de Medição”	<i>Gateway</i> de convergência ⇒ Processo “Realizar Ateste Técnico”

Outras informações:

Segundo o modelo atual, o serviço técnico de medição de sistemas será realizado por fornecedor externo contratado especificamente para tal finalidade (Fábrica de Métricas), segundo padrões de



qualidade pré-definidos.

8.3.7 Decisão: “Será preciso testar?”

Decisão: “Será preciso testar?”	
◆	Objetivo: Verificar necessidade de realizar testes e outros serviços de avaliação de qualidade relacionados às entregas da fase.
Será preciso testar?	
PARTICIPANTES	⇒ Líder Técnico
ENTRADAS	⇒ Entregáveis
SAÍDAS	SIM ⇒ Processo “Solicitar Serviço de Qualidade” NÃO ⇒ <i>Gateway</i> de convergência ⇒ Processo “Realizar Ateste Técnico”
↔ ATIVIDADE ANTERIOR PRÓXIMA ATIVIDADE =>	
Decisão “Todos os produtos entregues?” ⇒ SIM	VER SAÍDAS

Outras informações:

Da mesma forma, caso a decisão de saída seja negativa, o processo entra em um *gateway* de ativação incondicional de convergência aguardando a finalização de todos os demais processos que estiverem sendo realizados em paralelo para só então seguir para o Processo “Realizar Ateste Técnico”.



8.3.8 Processo: “Solicitar Serviço de Qualidade”

Processo: “Solicitar Serviço de Qualidade”	
Solicitar Serviço de Qualidade	Objetivo: Acionar o serviço técnico de testes de sistemas e controle de qualidade.
PARTICIPANTES	⇒ Líder Técnico
ENTRADAS	⇒ Decisão “Será preciso testar?” SIM
SAÍDAS	⇒ Solicitação de Serviço de Controle de Qualidade
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Decisão “Será preciso testar?”	Processo “Executar Serviço de Controle de Qualidade”

8.3.9 Processo: “Executar Serviço de Controle de Qualidade”

Processo: “Executar Serviço de Controle de Qualidade”	
Executar Serviço de Controle de Qualidade	Objetivo: Executar o serviço técnico de testes de sistemas e controle de qualidade;
PARTICIPANTES	⇒ Fábrica de Qualidade
ENTRADAS	⇒ Solicitação de Serviço de Controle de Qualidade
SAÍDAS	⇒ Pacote de Produtos de Testes de Sistemas e Controle de Qualidade
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Processo “Solicitar Serviço de Qualidade”	<i>Gateway</i> de convergência ⇒ Processo “Realizar Ateste Técnico”

Segundo o modelo atual, o serviço técnico de testes de sistemas e controle de qualidade será realizado por fornecedor externo contratado especificamente para tal finalidade (Fábrica de Qualidade), segundo padrões de qualidade pré-definidos.



8.3.10 Processo: “Realizar Ateste Técnico”

Processo: “Realizar Ateste Técnico”	
	Objetivo: Realizar o ateste técnico propriamente dito ao analisar o conjunto de produtos recebidos.
Realizar Ateste Técnico	
PARTICIPANTES	⇒ Líder Técnico
ENTRADAS	⇒ Entregáveis ⇒ Pacotes de Produtos ⇒ Atas de Reunião
SAÍDAS	⇒ Sinal de Fim do Processo
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Gateway de convergência	Sinal de Fim

Outras informações:

Os produtos recebidos no Processo “Receber Produtos” são avaliados e validados utilizando, quando disponível, as saídas da Fábrica de Métricas e Fábrica de Qualidade. Essa avaliação e validação consiste no ateste técnico propriamente dito. A Decisão “Validado?” do macro processo principal é respondida com base nesse ateste e o fluxo prosseguirá conforme o PDAS.

8.3.11 Evento: “Sinal de Fim”

Evento: “Sinal de Fim”	
	Objetivo: Sinal de finalização do processo.
Sinal de Fim	
PARTICIPANTES	—
ENTRADAS	—
SAÍDAS	—
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Processo “Realizar Ateste Técnico”	—



8.4 Subprocesso Realizar Reuniões de Planejamento

8.4.1 Evento: “Sinal de Início”

Evento: “Sinal de Início”	
	Objetivo: Sinal de início do processo “Realizar Reuniões de Planejamento”.
Sinal de Início	—
PARTICIPANTES	—
ENTRADAS	—
SAÍDAS	—
== ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ==
Evento “Recebimento de Demanda”	Processo “Realizar Reuniões TI e Negócios”



8.4.2 Processo: “Realizar Reuniões TI e Negócios”

Processo: “Realizar Reuniões TI e Negócios”	
	Objetivo: Reunir os <i>stakeholders</i> da Área de TI e da Área de Negócios para definir quem fará o papel de Líder Técnico e o papel de Líder de Negócios. Além disso, já começar a elaborar os primeiros itens do <i>Product Backlog</i> .
PARTICIPANTES	⇒ <i>Stakeholders</i> Área de TI ⇒ <i>Stakeholders</i> Área de Negócios
ENTRADAS	⇒ PEI ⇒ PDTI ⇒ Demanda Interna ⇒ Contratos
SAÍDAS	⇒ Documento de Definição de Líder Técnico e Líder de Negócios ⇒ <i>Product Backlog</i> Inicial ⇒ Atas de Reunião
⇐ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ⇒
Evento “Sinal de Início”	Processo “Realizar Reuniões entre CLDF e Fábricas”



8.4.3 Processo: “Realizar Reuniões CLDF e Fábricas (FS e FQ)”

Processo: “Realizar Reuniões CLDF e Fábricas (FS e FQ)”	
Realizar Reuniões CLDF e Fábricas (FS e FQ)	<p>Objetivo: Reunir os recém definidos Líderes CLDF (Técnico e de Negócio) com os representantes das Fábricas de Software e Qualidade para definir que outros artefatos serão necessários em cada fase do projeto (planejamento, desenvolvimento e encerramento); elaborar e entregar quaisquer artefatos de planejamento que forem julgados necessários; revisar e atualizar o <i>Product Backlog</i> inicial.</p>
PARTICIPANTES	<ul style="list-style-type: none">⇒ Líder Técnico⇒ Líder de Negócios⇒ Representante FS⇒ Representante FQ
ENTRADAS	<ul style="list-style-type: none">⇒ PEI⇒ PDTI⇒ Demanda Interna⇒ Contratos⇒ <i>Product Backlog</i> inicial
SAÍDAS	<ul style="list-style-type: none">⇒ Documento de Definição de Artefatos⇒ Artefatos de Planejamento⇒ <i>Product Backlog</i>⇒ Atas de Reunião
↔ ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ↔
Processo “Realizar Reuniões entre TI e Negócios”	Evento “Sinal de Fim”



8.4.4 Evento: “Sinal de Fim”

Evento: “Sinal de Fim”	
	Objetivo: Sinal de finalização do processo.
Sinal de Fim	—
PARTICIPANTES	—
ENTRADAS	—
SAÍDAS	—
==> ATIVIDADE ANTERIOR	PRÓXIMA ATIVIDADE ==>
Processo “Realizar Reuniões CLDF e Fábricas (FS e FQ)”	—



9 Normas Complementares

9.1 Normas Complementares para Contagem de Pontos de Função

As disposições a seguir detalham a execução prática do processo Executar Serviço de Medição de Sistemas (seção 8.3.6) sem alteração de seu fluxo formal no PDAS.

9.1.1 Reunião de Mediação de Divergência de Aferição de Pontos de Função

Nos casos em que houver divergência técnica quanto à aferição de contagem de pontos de função entre a Fábrica de Software e a Fábrica de Métricas, poderá ser convocada reunião técnica de mediação.

Participantes obrigatórios:

- Especialista em Métricas da Fábrica de Software;
- Especialista em Métricas da Fábrica de Métricas;
- Especialista em Negócio da CLDF.

Participação excepcional:

A participação de outros profissionais das partes somente poderá ocorrer:

- mediante justificativa técnica formal;
- em caráter excepcional;
- com autorização prévia da CLDF;
- desde que demonstrada relevância técnica para análise da contagem ou conhecimento do negócio envolvido.

10 Considerações Finais

A Metodologia de Desenvolvimento de Sistemas e o Processo de Desenvolvimento Ágil de Software apresentados nesse documento poderão ser atualizados em função de evolução tecnológica ou metodológica.

Os casos omissos e situações não previstas serão avaliados pela Área de TI da Câmara Legislativa do Distrito Federal.



DOCUMENTOS DE DIRETRIZES

Os documentos de diretrizes a seguir consistem em um conjunto de normas técnicas que estabelecem padrões obrigatórios para o desenvolvimento, implantação e manutenção de sistemas no âmbito da Câmara Legislativa do Distrito Federal, abrangendo arquitetura de software, modelagem e governança de bases de dados relacionais, gestão de configuração (controle de versão, repositórios, versionamento, branches, CI/CD e ambientes Kubernetes) e processos de testes de software (unitários, integração, funcionais e de desempenho), com o objetivo de assegurar padronização tecnológica, qualidade, segurança, rastreabilidade e conformidade dos sistemas institucionais.

Disposição Geral sobre Vigência e Publicação

Aplicam-se, para todos os fins, as versões mais atualizadas dos documentos de diretrizes mencionados nesta seção.

Os referidos documentos encontram-se disponibilizados no Portal da Câmara Legislativa do Distrito Federal (CLDF), em área específica destinada à governança e normatização de Tecnologia da Informação.

Eventual atualização ou revisão desses documentos produzirá efeitos imediatos no âmbito do Processo de Desenvolvimento Ágil de Software (PDAS), independentemente de alteração formal deste instrumento, desde que regularmente publicada no portal institucional.

Diretrizes de Arquitetura de Sistemas

Estabelecem o padrão tecnológico e arquitetural para o desenvolvimento de sistemas no âmbito da CLDF, definindo tecnologias homologadas para frontend, backend e infraestrutura, padrões de autenticação e autorização, armazenamento, observabilidade e esteira de integração e entrega contínua. As diretrizes asseguram padronização do stack tecnológico, interoperabilidade entre soluções, governança da evolução arquitetural e aderência ao ambiente institucional baseado em contêineres e orquestração.

Diretrizes de Arquitetura de Bases de Dados Relacionais

Definem normas para modelagem lógica e física de dados, padronização de nomenclatura de objetos, regras de integridade, versionamento de scripts e boas práticas de desempenho e governança. Estabelecem critérios como normalização, uso de chaves primárias sequenciais, constraints, padronização de prefixos e sufixos, além de requisitos de documentação e rastreabilidade, visando garantir qualidade, consistência, segurança e integridade dos dados institucionais.

Diretrizes de Gestão de Configuração

Dispõem sobre o controle de versão, organização de repositórios, estratégia de branches, versionamento semântico, uso de tags, pull requests, rastreabilidade de commits e padronização da esteira de CI/CD. Regulamentam também a implantação em ambientes segregados (desenvolvimento, homologação, produção, etc.) e a utilização de helm charts para deploy em Kubernetes, assegurando governança, rastreabilidade e controle do ciclo de vida do software.



Diretrizes de Testes

Estabelecem os tipos de testes obrigatórios e recomendados (unitários, integração, funcionais, desempenho, carga e stress), critérios mínimos de cobertura, ferramentas homologadas e padrões de automação. Definem regras para elaboração de plano de testes, casos de teste, registro de defeitos e evidências, garantindo qualidade, confiabilidade, conformidade normativa e mitigação de riscos antes da promoção de versões para ambientes superiores.

Diretrizes de Projetos e Backlog

Definem práticas de organização e rastreabilidade de requisitos, histórias de usuário, itens de backlog técnico, registro de defeitos e vinculação entre artefatos de documentação, código-fonte e testes. Estabelecem a obrigatoriedade de vinculação entre entregas, versões e demandas registradas em ferramenta de gestão, assegurando controle de escopo, transparência, governança e alinhamento entre requisitos funcionais e implementação técnica.



Referências

- Brasileiro, R. (2019). Veja agora 08 dicas para criar excelentes histórias de usuário, <http://www.metodoagil.com/historias-de-usuario/>. Acessado: 18/09/2019.
- Fowler, M. (2018). *Refactoring: improving the design of existing code*, Addison-Wesley Professional.
- IPHAN (2013). *Metodologia Iphan de Gestão de Demandas de Desenvolvimento Ágil de Softwares MIDAS*, 1.0 edn, Instituto do Patrimônio Histórico e Artístico Nacional.
- MP (2015). *Guia de Projetos de Software com práticas de métodos ágeis para o SISP: versão 1.0*, 1.0 edn, Ministério do Planejamento, Orçamento e Gestão, Secretaria de Logística e Tecnologia da Informação, Brasília. 1. Software 2. Tecnologia da Informação 3. Administração Pública.
- MP (2016). *Roteiro de Métricas de Software do SISP Versão 2.2*, 2.2 edn, Ministério do Planejamento, Desenvolvimento e Gestão, Brasília.
- Prikladnicki, R., Willi, R. e Milani, F. (2014). *Métodos ágeis para desenvolvimento de software*, Bookman Editora.
- Sutherland, J. e Schwaber, K. (2017). Guia do scrum, *Um guia definitivo para o Scrum: As regras do Jogo*. Scrum.org .
- TCU (2013). Ata nº 33, de 28 de agosto de 2013 - sessao ordinaria - plenario, *Ata*, Tribunal de Contas da União. Acórdãos Nºs 2261 a 2289 e 2290 a 2332.
- TCU (2015). Acordao 2362/2015 - plenario, *Relatorio de auditoria (ra)*, Tribunal de Contas da União. Relator: AUGUSTO NARDES, Processo: 002.116/2015-4, Número da ata: 38/2015 - Plenário.

As figuras utilizadas para ilustrar artefatos e reuniões são creditadas ao Freepik.com