

Metodologia de Desenvolvimento Ágil de Sistemas

Versão: 1.1
07/08/2023



Coordenação Geral de Tecnologia da Informação

Controle de Revisão

Rev.	Natureza da Revisão	Data	Elaborador	Revisor
1	Emissão Inicial	04/09/2017	Neirivan Júnior	Affonso Júnior Rosângela Scorzello
2	Atualização nos nomes de unidades	07/08/2023	Rafael Guimarães Bisaggio	Neirivan Júnior

Índice

1	Introdução	7
2	Objetivos.....	8
3	Papéis e responsabilidades.....	8
4	Processos organizacionais	9
4.1	Visão geral da entrega de valor	9
4.1.1	Registro de demanda.....	10
4.1.2	Construção da visão da demanda.....	10
4.1.3	Construção do roadmap da demanda	11
4.1.4	O planejamento dos releases e das sprints	11
4.1.5	A execução da demanda.....	12
4.1.6	A forma de pagamento das demandas.....	12
4.2	Fluxo de planejamento da demanda	12
4.2.1	Atividades de planejamento.....	13
4.2.2	Atividades do acompanhamento de projetos	16
4.3	Fluxo de execução da demanda	17
4.3.1	Atividades de planejamento do release	17
4.3.2	Atividades de planejamento da sprint.....	18
4.3.3	Atividades de gestão de ordem de serviço.....	21
4.3.4	Atividades de execução da sprint	21
4.3.5	Atividades da demonstração da sprint.....	23
4.3.6	Atividades de gestão de ordem de serviço.....	25
4.3.7	Atividades da conclusão do release.....	25
4.3.8	Atividades do acompanhamento de projetos	26
4.4	Fluxo de pagamento da demanda	27
5	Processos de apoio	29
5.1	Garantia da qualidade de software.....	29
5.1.1	Avaliação da qualidade de código	29
5.1.2	Testes de aceitação	30
5.1.3	Testes funcionais e de integração	30
5.1.4	Smoke tests	30
5.2	Gerência de mudanças.....	30
5.3	Gerência de ambientes de TI	30
5.4	Gerência de configuração	31
5.4.1	Git Flow.....	31
5.5	Gerência de ordens de serviço de TI.....	32

5.6	Acompanhamento de projetos (escritório de projetos)	33
6	Artefatos	34
7	Guias técnicos	34

Figuras

Figura 1 – Modelo de referência do SISP.....	7
Figura 2 – Fluxo de trabalho do tratamento da demanda	10
Figura 3 – Planejamento do release	11
Figura 4 – Planejamento da sprint.....	11
Figura 5 – Fluxo de planejamento da demanda	13
Figura 6 – Fluxo de execução da demanda.....	17
Figura 7 – Fluxo de entrega contínua	22
Figura 8 – Fluxo de execução da demanda.....	27
Figura 9 – Modelo de ramos do Git Flow	32
Figura 10 – Diagrama de estados de histórias de usuário e de ordens de serviço.....	33

Tabelas

Tabela 1 – Papéis e Responsabilidades exclusivos de servidores da SUSEP.....	8
Tabela 2 – Papéis e Responsabilidades delegáveis para funcionários de empresas terceirizadas.....	9
Tabela 3 – Papéis e Responsabilidades exclusivos em projetos executados por empresas terceirizadas ...	9
Tabela 4 – Lista de artefatos da metodologia	34

1 Introdução

Este documento descreve e formaliza a Metodologia de Desenvolvimento Ágil de Software (MDAS) da SUSEP. A MDAS está inserida em um contexto organizacional de tratamento de demandas para a área de TI da SUSEP e retrata os processos, fluxos de trabalho, papéis, responsabilidades e artefatos gerados pelos projetos da Coordenação Geral de Desenvolvimento de Tecnologia da Informação (CGDTI). Nesse modelo, apresentado ao longo desse documento, a demanda é recebida e tratada de forma uniformizada, seja novo produto ou manutenção em sistema já existente, com participação de empresa contratada ou não.

Para desenvolvimento desse modelo, foi utilizado como referência o processo descrito no “Guia de Projetos de Software com Práticas de Métodos Ágeis para o SISP”, elaborado e disponibilizado pela Secretaria de Tecnologia da Informação do Ministério do Planejamento (STI/MP). Esse guia é utilizado como documento orientador e não prescritivo, de forma a manter a autonomia dos projetos e equipes da SUSEP.

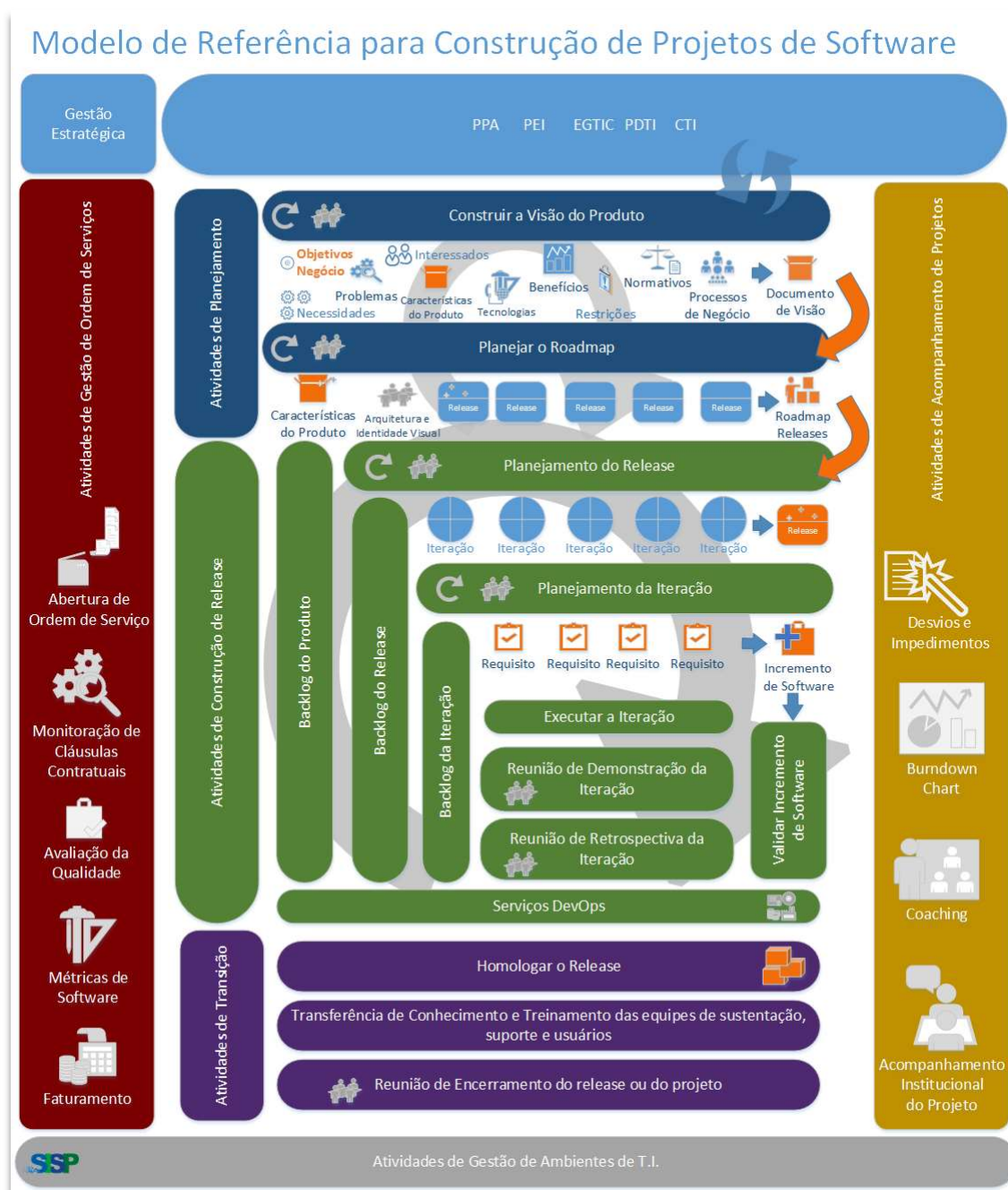


Figura 1 – Modelo de referência do SISP

A metodologia do SISP foi adaptada para a SUSEP, de forma a refletir a sua realidade. A seguir, são apresentadas as especificidades da metodologia de desenvolvimento ágil da SUSEP.

2 Objetivos

Dentre os objetivos da MDAS da SUSEP podemos citar a entrega de valor à sociedade, por meio da padronização da forma de desenvolvimento de software na SUSEP com uma abordagem ágil, iterativa e incremental durante o ciclo de vida do projeto, maior acurácia nos resultados, foco na melhoria da qualidade e na performance.

A MDAS da SUSEP é apoiada sobre quatro princípios básicos:

- **Priorização com base na entrega de valor:** promover práticas que permitam aos participantes do projeto e envolvidos desenvolver uma solução que maximize os benefícios dos envolvidos / cliente final e ao mesmo tempo seja compatível com as restrições do projeto.
- **Colaboração alinhando interesses e compartilhando entendimento:** promover práticas que promovam um ambiente de equipe saudável, permitam a colaboração e desenvolvam uma compreensão comum do projeto.
- **Concentrar na arquitetura no início para minimizar riscos e organizar o desenvolvimento:** promover práticas que garantam foco inicial na arquitetura para minimizar riscos do desenvolvimento.
- **Evoluir continuamente e obter *feedback*:** promover práticas que permitam a equipe obter *feedback* continuamente e o mais cedo possível, demonstrando o valor agregado a cada desenvolvimento.

3 Papéis e responsabilidades

A MDAS contempla alguns papéis que executam, cada um, atividades e responsabilidades específicas. Um papel é responsável por uma ou mais atividades e algumas atividades possuem mais de um responsável.

Tabela 1 – Papéis e Responsabilidades exclusivos de servidores da SUSEP

Papel	Descrição	Atividades que é Responsável ou que participa como um dos Responsáveis pela Execução
Gestor da TI	Liderança dos recursos da área de tecnologia da informação e do planejamento das atividades.	<ul style="list-style-type: none"> ➤ Receber demanda de software ➤ Abrir backlog do produto ➤ Definir equipe do produto
Product Owner (PO)	<p>Servidor da SUSEP com domínio sobre o negócio a ser tratado pelo sistema / produto.</p> <p>O PO será definido a cada demanda, de acordo com o seu domínio. Por isso, poderá ser tanto da área cliente quanto da área de desenvolvimento de sistemas.</p>	<ul style="list-style-type: none"> ➤ Elaborar visão da demanda ➤ Descrever histórias ➤ Elaborar roadmap da demanda ➤ Registrar informações do novo projeto ➤ Atualizar informações de projetos ➤ Estabelecer a priorização do backlog ➤ Atualizar backlog do release ➤ Definir objetivo da sprint ➤ Elaborar backlog preliminar da sprint ➤ Detalhar história ➤ Validar Incremento de Software ➤ Homologar release ➤ Abrir ordem de serviço ➤ Atualizar informações da OS
Gerente do produto	Quando o PO for de fora da TI e a demanda for desenvolvida por uma empresa terceirizada, deverá ser definido um servidor	<ul style="list-style-type: none"> ➤ Participar da reunião de planejamento ➤ Divide responsabilidades com o PO, atuando nas atividades que exigem maior conhecimento técnico.

	da TI para atuar como Gerente do produto e fazer a intermediação de conhecimento entre o PO (cliente) e o Scrum Master (da empresa contratada)	
--	--	--

Tabela 2 – Papéis e Responsabilidades delegáveis para funcionários de empresas terceirizadas

Papel	Descrição	Atividades que é Responsável ou que participa como um dos Responsáveis pela Execução
Scrum Master	Responsável pela coordenação do desenvolvimento da demanda	<ul style="list-style-type: none"> ➤ Estimar tamanho e custo das histórias ➤ Elaborar roadmap da demanda ➤ Estimar custo e prazo para conclusão da demanda ➤ Identificar atividades necessárias para realização das histórias ➤ Estimar esforço para construção das histórias ➤ Adequar o backlog da sprint ➤ Configurar integração contínua ➤ Integrar código ➤ Demonstrar incremento de software ➤ Realizar retrospectiva da sprint
Equipe de desenvolvimento	Responsável pelo efetivo desenvolvimento das demandas	<ul style="list-style-type: none"> ➤ Configurar integração contínua ➤ Analisar, codificar e testar a história ➤ Integrar código

Tabela 3 – Papéis e Responsabilidades exclusivos em projetos executados por empresas terceirizadas

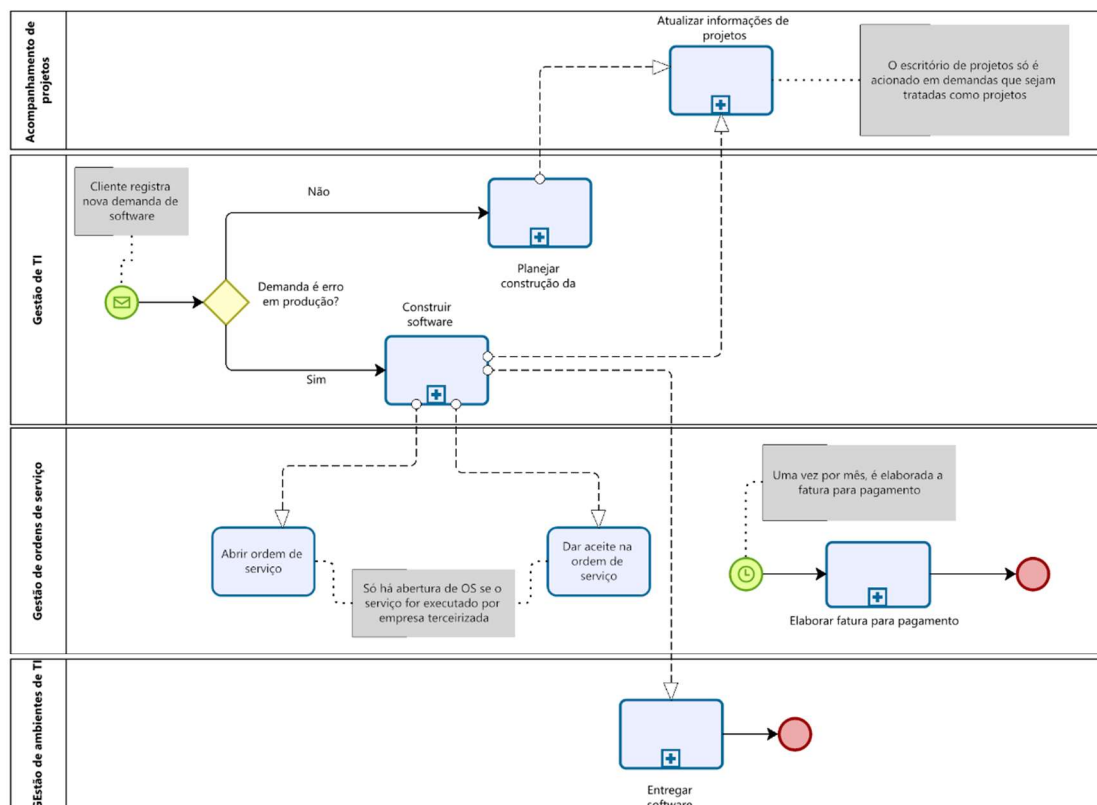
Papel	Descrição	Atividades que é Responsável ou que participa como um dos Responsáveis pela Execução
Gestor do Contrato	Servidor responsável pela gestão do contrato com a empresa terceirizada	<ul style="list-style-type: none"> ➤ Enviar fatura para pagamento
Fiscal Técnico	Servidor da área de desenvolvimento de sistemas responsável pela manutenção do contrato com a empresa terceirizada dentro dos padrões de conformidade	<ul style="list-style-type: none"> ➤ Verificar conclusão das demandas da fatura ➤ Validar tamanho das demandas
Preposto da contratada	Funcionário da contratada responsável pela gestão do contrato e interação com a Susep	<ul style="list-style-type: none"> ➤ Verificar tamanho da demanda ➤ Elaborar fatura para pagamento

4 Processos organizacionais

4.1 Visão geral da entrega de valor

A finalidade da construção de projetos de software é a entrega de um produto (ou versão funcional do produto de software) e a finalidade do produto é a entrega de valor agregado para o cliente. A entrega de valor é efetivada, através do atendimento das necessidades e expectativas do cliente.

A figura abaixo mostra o procedimento organizacional para gestão da demanda de TI na SUSEP.



Powered by
buzagi
Modeler

Figura 2 – Fluxo de trabalho do tratamento da demanda

4.1.1 Registro de demanda

Uma Demanda é uma solicitação feita à área de desenvolvimento de software para adicionar, retirar ou alterar funcionalidade em um produto. Pode representar um novo produto ou incrementos em um sistema já existente. Em termos finalísticos, uma demanda de software deve agregar valor ao cliente por meio de um sistema em funcionamento.

4.1.2 Construção da visão da demanda

A construção da Visão da Demanda consiste na identificação e compreensão das necessidades, problemas, riscos, objetivos de negócio, características-chaves do produto (software features), oportunidades, desafios e expectativas dos clientes.

A compreensão do atual funcionamento dos processos de negócio é importante para capturar as problemáticas e insatisfações do cliente para, finalmente, identificar as macrofunções de uma possível solução. As características-chaves ou macrofunções são atributos de valor ou serviços providos pela solução, derivados dos objetivos de negócio, para satisfazer necessidades de negócio dos clientes. Em outros termos, são os aspectos mais relevantes do produto de software que o cliente valorizará com mais facilidade.

O artefato Visão da Demanda reúne todo este conhecimento adquirido por meio de técnicas de elicitação de requisitos de software e análise de negócio. A partir dele será estabelecido o Backlog do Produto que consiste na lista de objetivos de negócio e características-chave a serem agregados ao produto até o final do atendimento da Demanda, desdobrados em requisitos de software (funcionais e não funcionais) e tarefas técnicas importantes para a construção de produtos de software.

4.1.3 Construção do roadmap da demanda

O modelo de atendimento às demandas de desenvolvimento de software é realizado por meio de ciclos de entregas incrementais. Por isso, após a concepção do documento de visão, inicia-se o planejamento de cada Release.

Dessa forma, um Release consiste no agrupamento de objetivos de negócio e características-chaves do produto por meio de critérios de priorização e de importância. Deve garantir um produto completo, funcional e utilizável ao usuário final.

O Roadmap da Demanda compreende o planejamento e o registro cronológico de releases, assim como objetivos de negócio e características-chaves associadas. Em outras palavras, é um planejamento de liberação de versões.

4.1.4 O planejamento dos releases e das sprints

Nessa etapa, os itens do Backlog do Produto deverão ser subdivididos em Backlogs de Releases. Essa subdivisão é orientada pelo Roadmap e pelos valores que se deseja entregar a cada Release.



Figura 3 – Planejamento do release

Cada Release, por sua vez, pode ser subdividido em uma ou mais Sprints. Cada uma destas inicia com planejamento específico onde é gerado o plano da sprint, com o objetivo de gerar um incremento de software. Os requisitos ou histórias de usuários contidas no Backlog do Release são selecionados conforme o objetivo da sprint e os critérios definidos pela equipe do produto para compor o Backlog da Sprint. Esse backlog é consumido para produzir o incremento de software da sprint. Durante a execução, a critério da equipe, poderão ser realizadas Reuniões Diárias de acompanhamento. Ao final do período definido, deverão ser realizadas a Demonstração e a Retrospectiva da sprint. A Validação do Incremento de Software poderá ocorrer em paralelo com a próxima sprint.



Figura 4 – Planejamento da sprint

Nessa fase, também devem ser definidos os conceitos de pronto e preparado, e qual será a duração das sprints.

Um mesmo release deve sempre produzir um software pronto para ser publicado e disponibilizado ao cliente. Durante as sprints, entretanto, o software deverá estar preparado, mas não necessariamente será disponibilizado ao cliente.

Sugere-se a adoção de sprints com duração entre uma e duas semanas para demandas com estimativa de esforço inferior a três meses de duração e sprints que durem entre duas e quatro semanas caso contrário.

4.1.5 A execução da demanda

Após o planejamento, a demanda segue para ser executada, de forma incremental, internamente ou, caso haja, por uma empresa terceirizada.

Caso a demanda seja para correção de um erro em produção, o planejamento e a execução da demanda devem ocorrer em paralelo para que a correção seja feita rapidamente e, assim, sejam evitados possíveis danos em dados ou nas atividades das pessoas afetadas pelas funcionalidades que não estão a contento.

A construção de cada história de usuário deverá contemplar todas as atividades necessárias para o pleno atendimento do requisito de negócio por ela representado, incluindo os scripts de banco de dados, código, modelos de testes e demais artefatos de construção de software.

A MDAS não obriga que sejam construídos artefatos de documentação, tais como diagramas UML ou documento de arquitetura. A necessidade de escrita desses documentos e de modelagem da solução deverá ser avaliada pela equipe do projeto de acordo com a experiência e tomando por base critérios como criticidade e complexidade do produto de software a ser desenvolvido.

Uma vez dada como concluída pela equipe de desenvolvimento, a demanda deverá estar com o código em pleno funcionamento e pronta para ser posta em produção. Essa publicação seguirá o planejamento feito no roadmap, de forma que o cliente tenha sempre um software funcional e útil. Além disso, cada história só será considerada completa após a validação pelo cliente e desde que todos os testes automatizados funcionem.

4.1.6 A forma de pagamento das demandas

Para as demandas construídas por empresas terceirizadas, serão abertas tantas ordens de serviço quantas forem as histórias a serem construídas pela empresa.

Uma vez por mês, a empresa contratada emitirá relatório com a listagem das ordens de serviço por ela atendidas, com respectivos tamanho e valor, de acordo com o contrato em vigência. O fiscal técnico do contrato validará o tamanho informado pela empresa e enviará ao gestor do contrato para que proceda ao pagamento daqueles valores.

4.2 Fluxo de planejamento da demanda

A figura abaixo apresenta o fluxo de trabalho para a fase de planejamento do atendimento da demanda, conforme a Metodologia de Desenvolvimento Ágil de Sistemas. O fluxo apresenta os processos, papéis, envolvidos e os produtos gerados.

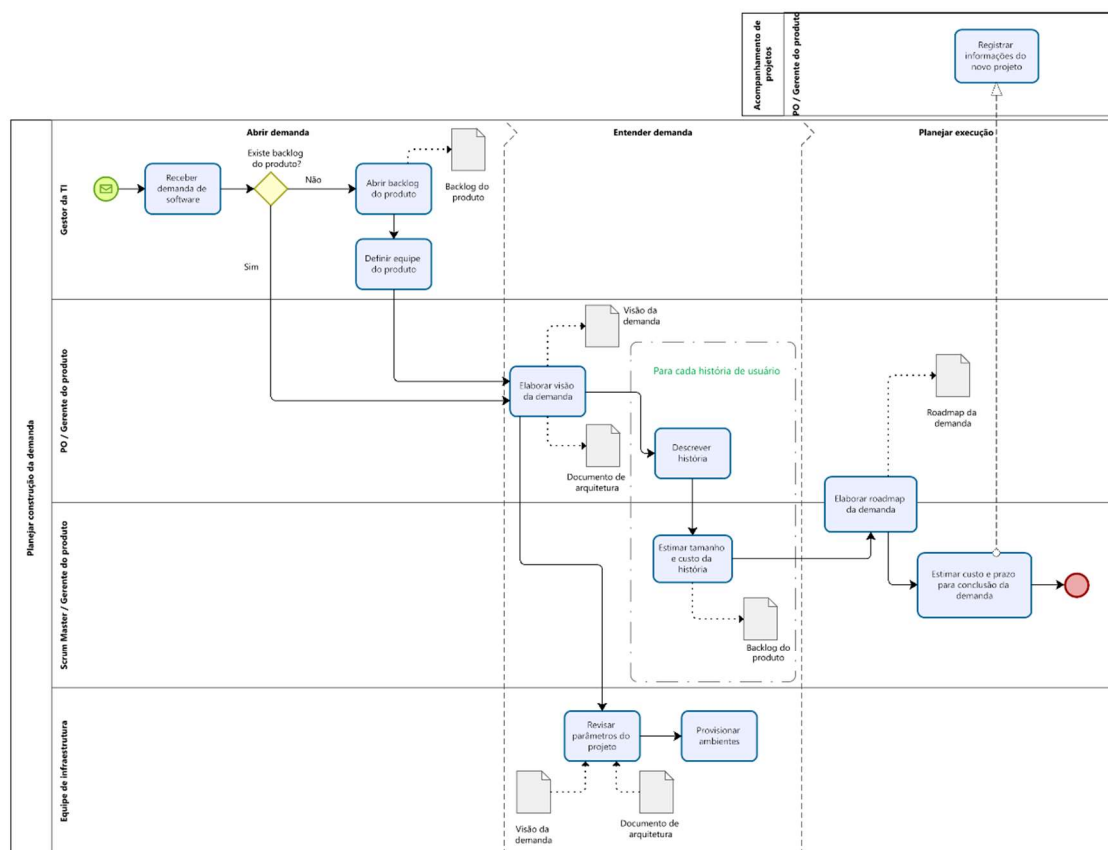


Figura 5 – Fluxo de planejamento da demanda

4.2.1 Atividades de planejamento

Atividade: Receber demanda de software

Nesta atividade, o Cliente identifica uma demanda e abre solicitação de serviço por meio do sistema de controle de demandas. A Central de Serviços da CGETI realiza análise e tratamento.

Tarefas	Descrição
Abrir a demanda	➤ O Cliente encaminha para a Central de Serviços formulário de abertura da demanda
Receber a demanda	➤ O Gestor de TI recebe a solicitação por meio da Central de Serviços
Papéis	➤ Responsável: Gestor de TI ➤ Participantes: ✓ Cliente
Entradas	➤ Necessidade de demanda de TI
Saídas	➤ Solicitação de serviço de TI
Observações	

Atividade: Abrir backlog do produto

Nesta atividade, após verificar que não existe um arcabouço ágil definido para o produto objeto da demanda, o Gestor de TI cria a estrutura do produto no software de gestão de ciclo de vida de aplicações (ALM) e, com isso, fica aberto o backlog do produto.

Tarefas	Descrição
Abrir o backlog	➤ O Gestor de TI cria a área do projeto e seu backlog no ALM
Papéis	➤ Responsável: Gestor de TI
Entradas	➤ Dados do produto
Saídas	➤ Backlog do produto
Observações	

Atividade: Definir equipe do produto

Uma vez criada a estrutura do produto no ALM, cabe ao gestor de TI formalizar a equipe do produto e de que forma será feita a sua execução.

Tarefas	Descrição
Definir equipe do produto	➤ O Gestor de TI formaliza a equipe do produto
Papéis	➤ Responsável: Gestor de TI
Entradas	➤ Dados do produto
Saídas	➤ Equipe definida: <ul style="list-style-type: none"> ✓ PO; ✓ Scrum Master; ✓ Equipe de desenvolvimento. ➤ Forma de execução: <ul style="list-style-type: none"> ✓ Interna; ou ✓ Empresa terceirizada
Observações	

Atividade: Elaborar visão da demanda

Essa atividade consiste na identificação e compreensão das necessidades, problemas, riscos, objetivos de negócio, características-chaves do produto (software features), oportunidades, desafios e expectativas dos clientes.

Tarefas	Descrição
Elaborar visão da demanda	➤ O PO descreve o valor a ser agregado com o atendimento da demanda
Papéis	➤ Responsável: PO
Entradas	➤ Backlog do produto
Saídas	➤ Visão da Demanda
Observações	

Atividade: Descrever história

Essa atividade consiste na descrição em alto nível de cada objetivo de negócio ou característica do produto identificados na visão da demanda e que deverão ser atendidos pela demanda de software.	
Tarefas	Descrição
Descrever história	➤ O PO descreve cada uma das histórias ou características do produto
Papéis	➤ Responsável: PO
Entradas	➤ Visão da demanda
Saídas	➤ Descrição em alto nível das histórias de usuário
Observações	

Atividade: Estimar tamanho e custo da história	
Essa atividade consiste na avaliação de tamanho e complexidade da história, para com isso chegar a valores estimados de prazo e custo para a construção da mesma.	
Tarefas	Descrição
Avaliar tamanho e complexidade	➤ O Scrum Master avalia, com base na experiência adquirida, o tamanho e a complexidade da história de usuário
Estimar tamanho e custo da história	➤ Com base no tamanho e na complexidade identificados, o Scrum Master estima o prazo e o custo para a construção da história
Papéis	➤ Responsável: Scrum Master
Entradas	➤ História de usuário
Saídas	➤ Estimativa de prazo e custo para a construção da história de usuário ➤ Backlog do produto atualizado
Observações	
Após a conclusão dessa atividade, o backlog do produto estará pronto com as descrições em alto nível das histórias a serem construídas para o atendimento da demanda.	

Atividade: Elaborar Roadmap da demanda	
Essa atividade consiste no planejamento cronológico das entregas de forma a fornecer ao cliente software funcional e utilizável durante todo o ciclo de vida do produto.	
Tarefas	Descrição
Priorizar backlog	➤ O PO define a ordem de importância e agregação de valor das histórias a serem desenvolvidas
Definir objetivos dos releases	➤ O PO define o objetivo que deseja alcançar com cada uma das entregas
Estimar sprints dos releases	➤ Para cada release programado, o PO, juntamente com o Scrum Master, identifica quantas sprints serão necessárias e quais as histórias deverão ser construídas em cada uma delas
Sequenciar as entregas	➤ O PO faz o sequenciamento cronológico das entregas
Papéis	➤ Responsável: PO ➤ Participantes: ✓ Scrum Master
Entradas	➤ Backlog do produto

Saídas	<ul style="list-style-type: none"> ➤ Roadmap da demanda ➤ Backlog do produto atualizado
Observações	

Atividade: Estimar custo e prazo para conclusão da demanda	
Com base nas histórias selecionadas para cada uma das sprints e seus respectivos prazos e custos, o Scrum Master estima o custo e prazo de cada uma das sprints. A partir das sprints, chega-se aos valores para os releases e, consequentemente, para a demanda como um todo.	
Tarefas	Descrição
Estimar custo e prazo dos releases	➤ O Scrum Master estima o custo e prazo de cada um dos releases com base nas sprints contidas neles e suas respectivas histórias de usuário
Estimar custo e prazo da demanda	➤ O Scrum Master estima o custo e prazo da demanda com base nos releases contidos nela
Papéis	➤ Responsável: Scrum Master
Entradas	<ul style="list-style-type: none"> ➤ Backlog do produto ➤ Roadmap da demanda
Saídas	<ul style="list-style-type: none"> ➤ Estimativa de custo e de prazo para conclusão dos releases ➤ Estimativa de custo e de prazo para conclusão da demanda
Observações	

4.2.2 Atividades do acompanhamento de projetos

Uma vez definidos os custos e prazos da demanda, o escritório de projetos é acionado para registrar essas informações e programar a execução, conforme o planejamento estratégico e a disponibilidade de recursos.

Atividade: Registrar informações do novo projeto	
Essa atividade consiste no registro da demanda, suas principais características, inclusive agregação de valor esperada, seus prazos e custos.	
Tarefas	Descrição
Registrar a demanda	➤ O PO registra a demanda já como uma proposta de projeto.
Papéis	➤ Responsável: PO
Entradas	<ul style="list-style-type: none"> ➤ Backlog do produto ➤ Roadmap da demanda ➤ Custos e prazos da demanda
Saídas	➤ Proposta de projeto
Observações	
Essa atividade é de responsabilidade da TI. Será desenvolvida pelo Gerente de Produtos quando o PO for da área cliente.	

4.3 Fluxo de execução da demanda

A figura abaixo apresenta o fluxo de trabalho para a fase de execução da demanda, conforme a Metodologia de Desenvolvimento Ágil de Sistemas. O fluxo apresenta os processos, papéis, envolvidos e os produtos gerados.

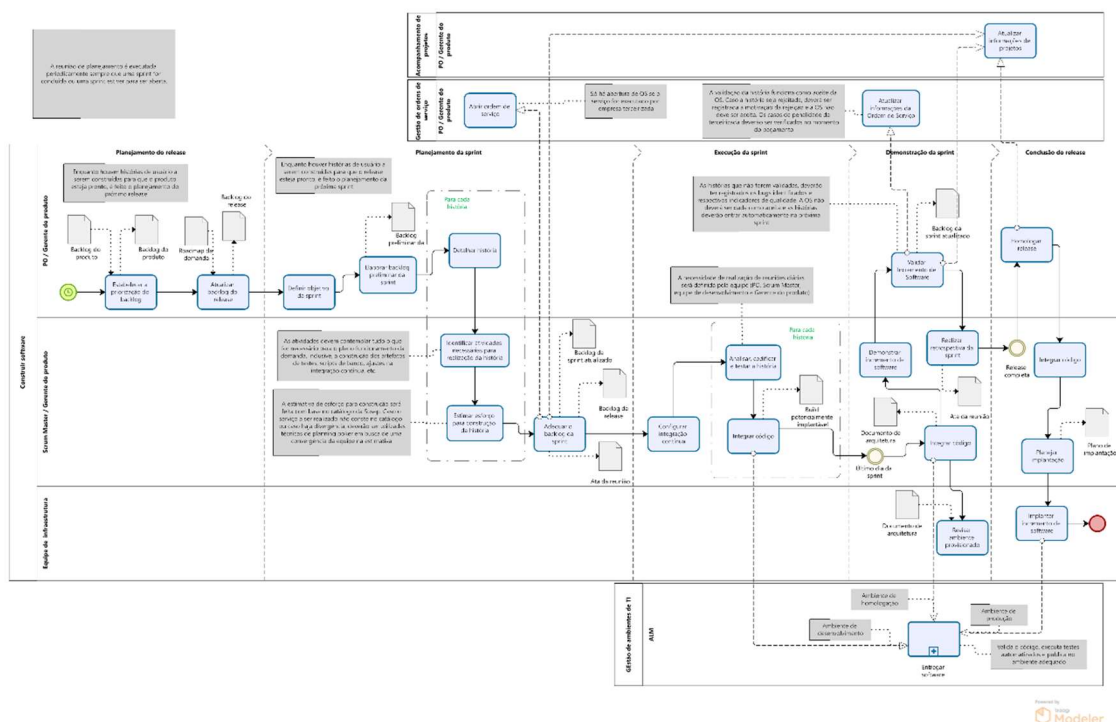


Figura 6 – Fluxo de execução da demanda

4.3.1 Atividades de planejamento do release

O planejamento do release ocorre durante a reunião de planejamento da sprint. Essa reunião acontece sempre que uma sprint for começar ou quando uma sprint for concluída e deve envolver toda a equipe da demanda, incluindo o PO, o Scrum Master, a Equipe de desenvolvimento e o Gerente do produto, quando houver.

Atividade: Estabelecer a priorização do backlog	
Nesta atividade, é estabelecida a ordem de prioridade de execução do backlog do produto, com base na expectativa de agregação de valor e nos objetivos de negócio a serem atendidos pela demanda.	
Tarefas	Descrição
Priorizar backlog	➤ O PO define a ordem de prioridade de cada uma das histórias de usuário contidas no backlog do produto
Papéis	➤ Responsável: PO
Entradas	➤ Backlog do produto
Saídas	➤ Backlog do produto priorizado
Observações	
Essa atividade é necessária pois, durante a execução da demanda, novas histórias podem ser identificadas pelo PO e adicionadas ao backlog do produto, ocasionando a necessidade de revisão do planejamento dos releases e das sprints a cada novo ciclo de desenvolvimento.	

Atividade: Atualizar backlog do release	
Nesta atividade, o backlog do release é atualizado tomando por base a priorização do backlog do produto e os objetivos a serem alcançados em cada release.	
Tarefas	Descrição
Atualizar backlog do release	➤ O PO e o Scrum Master ajustam o cronograma de construção das histórias dentro do release atual, com base no roadmap e no backlog do produto
Papéis	➤ Responsável: PO ➤ Participantes: ✓ Scrum Master
Entradas	➤ Backlog do produto ➤ Roadmap da demanda
Saídas	➤ Backlog do Release
Observações	

4.3.2 Atividades de planejamento da sprint

O planejamento da sprint ocorre durante a reunião de planejamento da sprint. Uma vez realizado o planejamento do release, a equipe passa para o planejamento da sprint. Caso a construção do release já esteja em andamento, a reunião tratará apenas do planejamento da sprint.

Atividade: Definir objetivo da sprint	
Nesta atividade, é definido o objetivo que se deseja alcançar com a execução da sprint.	
Tarefas	Descrição
Definir objetivo da sprint	➤ O PO define o objetivo da sprint
Papéis	➤ Responsável: PO
Entradas	➤ Backlog do release
Saídas	➤ Objetivo da sprint
Observações	

Atividade: Elaborar backlog preliminar da sprint	
Nesta atividade, é montado o backlog preliminar da sprint com a sugestão da lista de histórias que serão construídas na sprint, tomando por base o backlog do release e o objetivo da sprint.	
Tarefas	Descrição
Elaborar backlog preliminar da sprint	➤ O PO, com o auxílio do Scrum Master, monta como sugestão a lista de histórias a serem construídas na sprint
Papéis	➤ Responsável: PO ➤ Participantes: ✓ Scrum Master
Entradas	➤ Backlog do release ➤ Objetivo da sprint
Saídas	➤ Backlog preliminar da sprint

Observações

Atividade: Detalhar história	
Essa atividade consiste no detalhamento da história de usuário com o objetivo de identificar os requisitos funcionais e não funcionais para que seja atendido o objetivo de negócio que representa	
Tarefas	Descrição
Detalhar história	➤ Para cada uma das histórias selecionadas no backlog preliminar da sprint, o PO descreve o objetivo de negócio que a história atende.
Desdobrar histórias	➤ Eventualmente, durante o detalhamento da história, outras necessidades poderão ser identificadas. Essa tarefa consiste no registro das histórias que porventura sejam identificadas durante o referido detalhamento
Identificar cenários de teste	➤ No detalhamento da história, devem ser identificados os cenários de testes e os critérios de aceitação da história. Esses critérios serão os insumos para o desenvolvimento dos casos de testes unitários automatizados.
Papéis	➤ Responsável: PO
Entradas	➤ Backlog preliminar da sprint ➤ História de usuário
Saídas	➤ Detalhamento da história de usuário
Observações	

Atividade: Identificar atividades necessárias para realização da história	
Essa atividade consiste no levantamento de todas as atividades necessárias para a correta realização da história	
Tarefas	Descrição
Identificar atividades necessárias para realização da história	➤ Para que o software que implementa a história seja construído, é necessário que uma série de tarefas seja executada. Deve-se identificar todas essas tarefas. São exemplos: scripts de banco, protótipos, códigos de software, casos de teste automatizados.
Papéis	➤ Responsável: Scrum Master
Entradas	➤ História de usuário
Saídas	➤ Atividades necessárias para a realização da história
Observações	

Atividade: Estimar esforço para construção da história	
Essa atividade consiste na estimativa de esforço para a construção da história de usuário, durante o planejamento da sprint.	
Tarefas	Descrição
Estimar esforço da atividade	➤ Para cada atividade identificada como necessária para a construção da história, é necessário estimar o esforço para a construção. ➤ Deve ser utilizado o catálogo de serviços da Susep.

	<ul style="list-style-type: none"> ➤ Caso o serviço a ser realizado não conste no catálogo ou caso haja divergência, deverão ser utilizadas técnicas adequadas em busca de uma convergência da equipe na estimativa. Entre técnicas possíveis podemos citar: <ul style="list-style-type: none"> ↳ Construção acompanhada: <ul style="list-style-type: none"> ⇒ A equipe faz a construção conforme esperada e o esforço é medido em conjunto pelas equipes da Susep e da contratada ↳ Planning poker: <ul style="list-style-type: none"> ⇒ Cada um dos presentes na reunião de planejamento da sprint informa quanto acredita que seja o esforço necessário para a construção da história ⇒ Caso haja discrepância de valores, cada um dos que estimou um valor acima ou abaixo da média deverá explicar o motivo de ter sugerido valor diferente do estimado pelos demais participantes da reunião ⇒ Repete-se os dois últimos passos até que o valor convirja.
Papéis	<ul style="list-style-type: none"> ➤ Responsável: Scrum Master ➤ Participantes: <ul style="list-style-type: none"> ✓ PO ✓ Equipe de desenvolvimento ✓ Gerente do Produto (caso exista)
Entradas	<ul style="list-style-type: none"> ➤ Atividades necessárias para a conclusão da história de usuário
Saídas	<ul style="list-style-type: none"> ➤ Estimativa de esforço para a conclusão de cada uma das atividades e, consequentemente, da história
Observações	

Atividade: Adequar o backlog da sprint

Essa atividade consiste na definição das histórias que seguirão para construção na sprint a ser iniciada.

Tarefas	Descrição
Definir histórias	<ul style="list-style-type: none"> ➤ Com base nas histórias previamente detalhadas, nas suas atividades e estimativas de esforço, os participantes da reunião de planejamento definem quais atividades serão desenvolvidas durante a sprint.
Elaborar e assinar ata da reunião	<ul style="list-style-type: none"> ➤ Caso a equipe julgue necessário, poderá ser gerada uma ata da reunião em que deverá constar a lista das histórias selecionadas para ser desenvolvidas na próxima iteração e seus respectivos esforços estimados. A ata deverá ser assinada por todos os participantes.
Papéis	<ul style="list-style-type: none"> ➤ Responsável: Scrum Master ➤ Participantes: <ul style="list-style-type: none"> ✓ PO ✓ Equipe de desenvolvimento ✓ Gerente do Produto (caso exista)
Entradas	<ul style="list-style-type: none"> ➤ Backlog preliminar da sprint
Saídas	<ul style="list-style-type: none"> ➤ Backlog da sprint ➤ Backlog do release atualizado ➤ Ata da reunião
Observações	
Durante a seleção das histórias que irão compor a sprint, os participantes da reunião de planejamento deverão levar em consideração os dados do contrato em andamento para que não se estourem recursos orçamentários. Ao mesmo tempo, deverão ser levados em consideração o tempo da sprint e	

a capacidade de desenvolvimento da equipe para que seja selecionado um número suficiente e adequado de histórias para a sprint.

4.3.3 Atividades de gestão de ordem de serviço

Essas atividades são constantes do fluxo de gestão do contrato de TI e somente são executadas quando o desenvolvimento da demanda for feito por empresa terceirizada.

Atividade: Abrir ordem de serviço	
Essa atividade consiste na abertura de ordem de serviço para atendimento pela empresa terceirizada.	
Tarefas	Descrição
Abrir ordem de serviço	➤ Caso o desenvolvimento da demanda seja feito por empresa terceirizada, o PO deverá abrir uma ordem de serviço para cada uma das histórias selecionadas para construção durante a sprint. A ordem de serviço será registrada com a situação ABERTA
Papéis	➤ Responsável: PO
Entradas	➤ Backlog sprint
Saídas	➤ Ordens de serviço
Observações	

4.3.4 Atividades de execução da sprint

Uma vez concluído o planejamento da sprint, passa-se para a execução da mesma.

Caso a execução seja realizada por empresa terceirizada, a equipe da SUSEP deve evitar realizar qualquer tipo de ingerência na forma de execução do serviço. Entretanto, esta autarquia se resguarda do direito de exigir que parâmetros contratuais de qualidade de serviço e de estrutura da equipe de desenvolvimento sejam atendidos.

Atividade: Configurar integração contínua	
Essa atividade consiste na configuração do software ALM para permitir que a entrega contínua (validações de código, testes automatizados e publicação dos sistemas) e, consequentemente, os parâmetros de qualidade sejam alcançados.	
Tarefas	Descrição
Configurar validação de código	➤ O ALM deve ser configurado para somente aceitar que sejam adicionados ao repositório códigos que atendam aos critérios mínimos de qualidade definidos pela CGETI
Configurar geração do build	➤ O ALM deve ser configurado para a cada publicação de código no repositório seja gerado um build (versão executável do sistema), de forma a garantir que o código salvo seja compilável.
Configurar testes unitários	➤ O ALM deve ser configurado para executar os testes unitários de forma a garantir que o build atenda todos os critérios de aceitação definidos para as histórias já desenvolvidas.
Configurar publicação do build	➤ O ALM deve ser configurado para publicar o build gerado no ambiente correto, de acordo com o ramo do repositório de onde tenha sido recuperado o código, conforme sessão Gestão de Configuração dessa MDAS.
Configurar smoke tests	➤ O ALM deve ser configurado para executar testes de navegação no sistema após a publicação, sem, entretanto, salvar dados, para garantir que a versão

	recentemente publicada não apresente erro grosseiro, tais quais erros de configuração, acesso a base de dados, entre outros.
Configurar a comunicação com as partes interessadas	➤ O ALM deve ser configurado para manter um padrão de comunicação com as partes interessadas por meio de mensagens eletrônicas, objetivando informar das publicações realizadas ou de falhas ocorridas durante esse processo.
Configurar o retorno de versão	➤ O ALM deve ser configurado para retornar a versão do sistema para o estado anterior, caso seja identificada alguma falha após a publicação
Configurar outros serviços	➤ Poderá ser necessário, ainda, a depender da natureza da história ou da sprint, configurar o ALM para realizar outros serviços de forma a garantir a correta publicação do sistema nos ambientes de software. ➤ São exemplos desses serviços: a execução de scripts de banco, a configuração de containers de aplicações, entre outros.
Papéis	➤ Responsável: Scrum Master
Entradas	➤ Backlog sprint
Saídas	➤ ALM configurado
Observações	
A associação entre ramo de código do repositório e o ambiente no qual o sistema deverá ser publicado é definida na sessão Gestão de Configuração dessa MDAS.	
A sessão Entrega Contínua dessa MDAS descreve mais detalhadamente os requisitos desse serviço.	

O fluxo de entrega contínua é exemplificado abaixo:

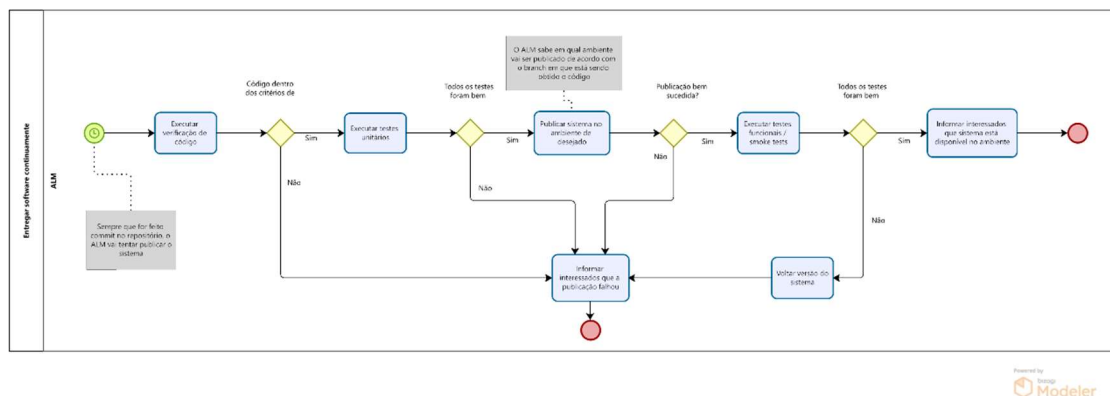


Figura 7 – Fluxo de entrega contínua

Atividade: Analisar, codificar e testar a história	
Essa atividade consiste na construção propriamente dita da história	
Tarefas	Descrição
Analisar e codificar história	➤ O Scrum Master e sua equipe deverão analisar e codificar a história, atendendo a cada uma das atividades identificadas como necessárias para a conclusão satisfatoriamente da história.
Testar a história	➤ O Scrum Master e sua equipe deverão construir casos de testes unitários e, ao mesmo tempo, testar o software resultante da construção da história de

	forma a garantir que atenda ao que foi solicitado pelo cliente e que os critérios mínimos de qualidade sejam atingidos.
Papéis	<ul style="list-style-type: none"> ➤ Responsável: Scrum Master ➤ Participantes: <ul style="list-style-type: none"> ✓ Equipe de desenvolvimento
Entradas	<ul style="list-style-type: none"> ➤ Atividades da história
Saídas	<ul style="list-style-type: none"> ➤ Código fonte ➤ Scripts de banco ➤ Testes automatizados ➤ Outros
Observações	

Atividade: Integrar código

Essa atividade consiste na adição no repositório da SUSEP do código gerado durante a construção da história

Tarefas	Descrição
Integrar código	<ul style="list-style-type: none"> ➤ O Scrum Master deverá, após realizar todos os testes de forma que seja possível garantir os critérios mínimos de qualidade, integrar o código desenvolvido por sua equipe ao código constante no repositório da SUSEP. ➤ O código integrado deverá ser compilável, estável e atender minimamente a todos os requisitos de qualidade estabelecidos pelos roteiros e guias desta autarquia e, eventualmente, os níveis mínimos de serviço previstos em contrato.

Papéis	<ul style="list-style-type: none"> ➤ Responsável: Scrum Master
Entradas	<ul style="list-style-type: none"> ➤ Código fonte ➤ Scripts de banco ➤ Testes automatizados ➤ Outros
Saídas	<ul style="list-style-type: none"> ➤ Repositório atualizado

Observações

Deverão ser seguidas as definições feitas na sessão Gestão de Configuração deste e demais guias de forma a garantir que os ramos de desenvolvimento, homologação e produção estejam íntegros e com código compilável e potencialmente implantável nos respectivos ambientes.

4.3.5 Atividades da demonstração da sprint

Essas atividades ocorrem durante a reunião de demonstração da sprint. Essa reunião acontece sempre que uma sprint for concluída e deve envolver toda a equipe da demanda, incluindo o PO, o Scrum Master, a Equipe de desenvolvimento e o Gerente do produto, quando houver.

Atividade: Demonstrar incremento de software

Essa atividade consiste na demonstração de que as histórias de usuário selecionadas para a sprint foram construídas e que funcionam conforme o definido pelo PO

Tarefas	Descrição
---------	-----------

Demonstrar incremento de software	<ul style="list-style-type: none"> ➤ Para cada história do backlog da sprint, o Scrum Master demonstra que ela foi construída, que os testes funcionam, que os critérios mínimos de qualidade foram atendidos e que a funcionalidade está disponível para ser validada pelo PO
Papéis	<ul style="list-style-type: none"> ➤ Responsável: Scrum Master ➤ Participantes: <ul style="list-style-type: none"> ✓ PO ✓ Equipe de desenvolvimento ✓ Gerente do Produto (caso exista)
Entradas	<ul style="list-style-type: none"> ➤ Backlog da sprint ➤ Software em funcionamento
Saídas	<ul style="list-style-type: none"> ➤ Evidências de atendimento dos critérios de aceitação da história e dos critérios mínimos de qualidade
Observações	

Atividade: Validar Incremento de Software

Essa atividade consiste na validação de que as histórias de usuário selecionadas para a sprint foram construídas e que funcionam conforme o definido

Tarefas	Descrição
Validar incremento de software	<ul style="list-style-type: none"> ➤ Para cada história do backlog da sprint, o PO verifica que ela foi construída e que funciona conforme o definido
Papéis	<ul style="list-style-type: none"> ➤ Responsável: PO
Entradas	<ul style="list-style-type: none"> ➤ Backlog da sprint ➤ Software em funcionamento
Saídas	<ul style="list-style-type: none"> ➤ Aceite das histórias construídas
Observações	

Atividade: Realizar retrospectiva da sprint

Essa atividade consiste na realização da retrospectiva da sprint, registrando os fatores facilitadores e complicadores da sprint, servindo para a melhoria contínua do processo de desenvolvimento de software

Tarefas	Descrição
Realizar retrospectiva da sprint	<ul style="list-style-type: none"> ➤ O Scrum Master relata a execução da sprint, destacando os fatores facilitadores e complicadores do processo
Elaborar e assinar ata da reunião	<ul style="list-style-type: none"> ➤ Caso a equipe julgue necessário, poderá ser gerada e assinada por todos os participantes uma ata da reunião, em que deverá constar: <ul style="list-style-type: none"> ✓ A lista das histórias entregues e aceitas ✓ A lista das histórias entregues e rejeitadas. <ul style="list-style-type: none"> ↳ Deverão ser explicitados os erros identificados ou os critérios de qualidade não atendidos que motivaram a rejeição da entrega. ✓ A lista de histórias não entregues ✓ Fatores facilitadores, complicadores e demais observações feitas durante a retrospectiva da sprint

Papéis	<ul style="list-style-type: none"> ➤ Responsável: Scrum Master ➤ Participantes: <ul style="list-style-type: none"> ✓ PO ✓ Equipe de desenvolvimento ✓ Gerente do Produto (caso exista)
Entradas	<ul style="list-style-type: none"> ➤ Backlog da sprint
Saídas	<ul style="list-style-type: none"> ➤ Estatísticas do desenvolvimento <ul style="list-style-type: none"> ✓ Velocidade da equipe ✓ Indicadores de qualidade ✓ Outros ➤ Lições aprendidas ➤ Pontos de atenção para a melhoria do processo ➤ Ata da reunião
Observações	

4.3.6 Atividades de gestão de ordem de serviço

Essas atividades são constantes do fluxo de gestão do contrato de TI e somente são executadas quando o desenvolvimento da demanda for feito por empresa terceirizada.

Atividade: Atualizar informações da Ordem de Serviço	
Essa atividade consiste na atualização da ordem de serviço, registrando as que foram concluídas corretamente e as que não, identificando possíveis penalidades para a empresa terceirizada.	
Tarefas	Descrição
Fechar ordens de serviço concluídas	<ul style="list-style-type: none"> ➤ Caso o desenvolvimento da demanda seja feito por empresa terceirizada, ao final de cada sprint, o PO, ao aceitar a história, estará dando o aceite da respectiva ordem de serviço. A ordem de serviço passa para a situação RESOLVIDA.
Atualizar ordens de serviço não atendidas	<ul style="list-style-type: none"> ➤ Para as ordens de serviço cujo atendimento não tenha sido plenamente realizado, o PO deve registrar o motivo da rejeição da entrega, identificando erros ou critérios de qualidade que tenham impedido o aceite da história.
Papéis	<ul style="list-style-type: none"> ➤ Responsável: PO
Entradas	<ul style="list-style-type: none"> ➤ Backlog sprint
Saídas	<ul style="list-style-type: none"> ➤ Ordens de serviço atualizadas
Observações	
Os casos de não realização ou de rejeição de entregas deverão ser considerados para efeitos de glosa ou de penalidade da terceirizada no momento do pagamento.	

4.3.7 Atividades da conclusão do release

Essas atividades ocorrem ao fim de todas as sprints planejadas para a construção do release e, quando, consequentemente, todas as histórias do backlog do release foram concluídas. Nesse ponto, deverá haver um produto de software pronto, estável e que deverá ser disponibilizado ao cliente.

Atividade: Homologar release	
Essa atividade consiste na homologação do release lançado como um todo, envolvendo funcionalidades das diferentes sprints executadas durante a construção do release	
Tarefas	Descrição

Homologar release	➤ O PO valida todo o sistema disponibilizado, de forma a garantir que os objetivos de negócio identificados foram atendidos satisfatoriamente.
Papéis	➤ Responsável: PO
Entradas	➤ Backlog do release ➤ Software em funcionamento
Saídas	➤ Aceite do release ➤ Backlog do release atualizado ➤ Backlog do produto atualizado
Observações	

Atividade: Integrar código

Essa atividade consiste na adição no repositório da SUSEP do código gerado durante a construção do release

Tarefas	Descrição
Integrar código	➤ O Scrum Master deverá integrar o código gerado durante o release no ramo de produção.
Papéis	➤ Responsável: Scrum Master
Entradas	➤ Código fonte ➤ Scripts de banco ➤ Testes automatizados ➤ Outros
Saídas	➤ Ramo de produção atualizado
Observações	
Após a integração do código, o ALM deverá publicar o sistema em um ambiente de produção.	

4.3.8 Atividades do acompanhamento de projetos

Uma vez concluída uma sprint ou release, o escritório de projetos é acionado para registrar as informações da execução, atualizar as estatísticas e os cronogramas de execução dos projetos da SUSEP.

Atividade: Atualizar informações de projetos

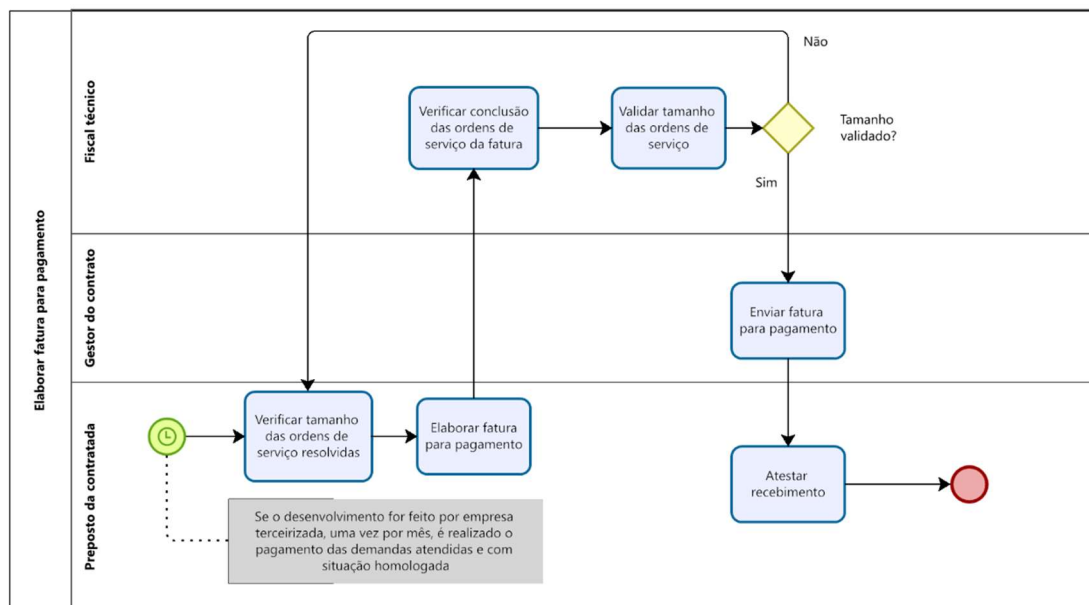
Essa atividade consiste no registro dos dados da execução da sprint ou do release recentemente concluídos.

Tarefas	Descrição
Registrar informações da execução da sprint ou do release	➤ O PO registra as informações do planejamento e da execução da sprint ou do release, tais quais histórias selecionadas para construção, histórias concluídas, desvios de prazos ou custos.
Papéis	➤ Responsável: PO
Entradas	➤ Backlog da sprint ou do release
Saídas	➤ Informações do projeto atualizadas ➤
Observações	

Essa atividade é de responsabilidade da TI. Será desenvolvida pelo Gerente de Produtos quando o PO for da área cliente.

4.4 Fluxo de pagamento da demanda

A figura abaixo apresenta o fluxo de trabalho para a fase de pagamento da demanda, conforme a Metodologia de Desenvolvimento Ágil de Sistemas. O fluxo apresenta os processos, papéis, envolvidos e os produtos gerados. Essas atividades somente são executadas quando o desenvolvimento da demanda for feito por empresa terceirizada.



Powered by
brazo
Modeler

Figura 8 – Fluxo de execução da demanda

Atividade: Verificar tamanho das ordens de serviço RESOLVIDAS	
Nesta atividade, é verificado o tamanho exato das Ordens de Serviço RESOLVIDAS pela empresa no mês anterior.	
Tarefas	Descrição
Verificar tamanho da demanda	➤ O Preposto da contratada calcula o tamanho de cada uma das ordens de serviço registradas e que estejam com a situação RESOLVIDA
Papéis	➤ Responsável: Preposto da contratada
Entradas	➤ Ordens de serviço ➤ História de usuário ➤ Software em funcionamento
Saídas	➤ Tamanho de cada Ordem de Serviço RESOLVIDA (memória de cálculo)
Observações	

Atividade: Elaborar fatura para pagamento

Nesta atividade, são listadas todas as ordens de serviço registradas e que estejam com a situação RESOLVIDA.

Tarefas	Descrição
Elaborar fatura para pagamento	➤ O Preposto da contratada seleciona as ordens de serviço registradas e que estejam com a situação RESOLVIDA e anexa a cada uma delas o tamanho verificado anteriormente

Papéis	➤ Responsável: Preposto da contratada
Entradas	➤ Ordens de serviço
Saídas	➤ Lista de ordens de serviço a serem faturadas

Observações
Atividade: Verificar conclusão das ordens de serviço da fatura

Nesta atividade, é verificada a correta conclusão de cada uma das ordens de serviço listadas.

Tarefas	Descrição
Verificar conclusão das demandas da fatura	➤ O Fiscal técnico verifica se todas as ordens de serviço listadas foram efetivamente construídas pela empresa terceirizada observando se todas as ordens de serviço estão com a situação RESOLVIDA.

Papéis	➤ Responsável: Fiscal técnico
Entradas	➤ Lista de ordens de serviço a serem faturadas
Saídas	➤ Lista de ordens de serviço a serem faturadas validada

Observações
Atividade: Validar tamanho das ordens de serviço

Nesta atividade, é verificado se o tamanho informado de cada uma das ordens de serviço listadas está correto.

Tarefas	Descrição
Validar tamanho da demanda	➤ O Fiscal técnico verifica se o tamanho de todas as ordens de serviço listadas está correto

Papéis	➤ Responsável: Fiscal técnico
Entradas	➤ Lista de ordens de serviço a serem faturadas
Saídas	➤ Lista de ordens de serviço a serem faturadas com tamanhos validados

Observações
Atividade: Enviar fatura para pagamento

Nesta atividade, a lista de ordens de serviços a serem faturadas é enviada para efetivo pagamento pelo setor administrativo da SUSEP.

Tarefas	Descrição
---------	-----------

Enviar fatura para pagamento	➤ O Gestor do contrato envia para pagamento à empresa terceirizada a lista de ordens de serviço efetivamente executada e dentro dos critérios de qualidade, juntamente com os tamanhos validados e valores associados.
Alterar situação das ordens de serviço	➤ O Gestor do contrato altera a situação das ordens de serviço para FECHADA
Papéis	➤ Responsável: Gestor do contrato
Entradas	➤ Lista de ordens de serviço a serem faturadas com tamanhos validados
Saídas	➤ Lista de ordens de serviço para faturamento
Observações	

Atividade: Atestar recebimento	
Nesta atividade, a lista de ordens de serviços enviadas para pagamento tem o registro do pagamento efetivamente realizado.	
Tarefas	Descrição
Atestar recebimento	➤ O Preposto da contratada atesta que as faturas foram efetivamente pagas para Susep.
Papéis	➤ Responsável: Preposto da contratada
Entradas	➤ Lista de ordens de serviço enviadas para pagamento
Saídas	➤ Lista de ordens de serviço pagas
Observações	

5 Processos de apoio

Existem alguns processos que suportam todas as fases da metodologia e que em paralelo são responsáveis por garantir a qualidade dos produtos de trabalho gerados e do seguimento dos processos e das melhores práticas. Abaixo é detalhado o funcionamento de cada um desses processos de apoio.

5.1 Garantia da qualidade de software

5.1.1 Avaliação da qualidade de código

O código gerado será sempre avaliado em critérios de complexidade, aderência aos padrões de nomenclatura e outros fatores.

As avaliações de código serão feitas, preferencialmente, de forma automatizada e de maneira integrada ao ALM. Essas ferramentas deverão impedir que códigos que não atendam aos critérios mínimos de qualidade definidos sejam integradas ao repositório do produto.

Apesar de a avaliação ser feita de forma automatizada, eventualmente, os códigos constantes nos repositórios dos produtos da SUSEP poderão passar por inspeção realizada pela equipe da CGETI. Nessas inspeções serão avaliados aspectos como a separação entre camadas da aplicação, a tradução dos conceitos de negócio para o código, aderência às boas práticas de codificação, entre outros.

As métricas de qualidade exigidas são as listadas no guia técnico de qualidade de software.

Sempre que houver manutenção em código existente, as métricas de código resultante devem estar iguais ou melhores do que antes da manutenção/melhoria.

5.1.2 Testes de aceitação

Sempre que uma história de usuário for detalhada, deverão ser definidos os critérios de aceitação dessa história. Esses critérios nada mais são do que os cenários que poderão ocorrer com a execução do software que representa a respectiva história. Em termos práticos, os critérios de aceitação deverão ser transformados em casos de testes para validar e garantir que o código gerado cobre todas as situações previstas para acontecer.

Quando a história for desenvolvida utilizando tecnologia que permita, a criação dos casos de testes desses cenários deverão ser construídos na forma de testes unitários automatizados. Esses testes deverão ser integrados à ferramenta de ALM e deverão executar sempre que um build seja gerado, garantindo que o software construído esteja aderente às regras definidas para suas histórias de usuário.

Para os sistemas construídos em tecnologias que não permitam a construção de testes unitários automatizados, os casos de testes de cenários deverão ser elencados em um roteiro de casos de testes e sua execução deve ser evidenciada para garantir que o software atenda a contento as regras definidas.

5.1.3 Testes funcionais e de integração

Ao fim de cada sprint ou release, deverão ser executados testes para verificar se as funcionalidades dos sistemas, após a integração do código novo com o código existente, estão funcionando corretamente. Esses testes são divididos entre o Scrum Master e o PO. Esses atores são os responsáveis por garantir que a aplicação disponibilizada nos ambientes de TI da SUSEP estejam funcionais e disponíveis para uso. Essa responsabilidade será de um ou do outro ator, de acordo com a fase da construção do produto.

5.1.4 Smoke tests

Deverão ser construídos, ainda, roteiros automatizados de testes de interface. Esses roteiros devem fazer uma navegação superficial no sistema sem, entretanto, salvar dados. Deverão ser integrados à ferramenta de ALM de forma que, sempre que houver a publicação de versão do sistema em ambiente da TI, seja feita uma passagem nas principais telas do sistema de forma a garantir que não ocorra qualquer erro grosseiro que impeça o uso do produto.

5.2 Gerência de mudanças

Ao longo do processo de desenvolvimento de software, ocorrem alterações nos requisitos devido a vários motivos. À medida que o cliente aumenta seu conhecimento em relação às suas necessidades e a equipe ganha experiência no sistema e em seu contexto, novos entendimentos são identificados e mudanças surgem nos já existentes.

O modelo de desenvolvimento ágil de software entende o surgimento de mudanças como algo natural e inevitável. As mudanças deverão ser detalhadas e sua execução deverá ser planejada durante as reuniões de planejamento das sprints.

Caso uma mudança seja solicitada, sua construção deverá ser tratada como uma história de usuário e, se for desenvolvida por empresa terceirizada, deverá ser aberta uma ordem de serviço específica para o atendimento dessa nova demanda.

5.3 Gerência de ambientes de TI

Uma praxe do desenvolvimento de software é a existência de, pelo menos, três ambientes de TI:

1. Ambiente de desenvolvimento: utilizado pelas equipes de desenvolvimento para fazer testes de integração e provas de conceito nas soluções adotadas;
2. Ambiente de homologação: utilizado para que os clientes validem o software entregue, verificando sua aderência ao que foi especificado;

3. Ambiente de produção: ambiente em que o software validado é disponibilizado aos usuários para que gerem dados reais.

No ambiente de desenvolvimento deverá ser dada à equipe de desenvolvimento permissão para alterar o ambiente, instalar outros softwares, alterar objetos de bancos de dados, entre outras opções. Já os ambientes de homologação e produção deverão ter acesso restrito e sua configuração deverá ser feita, sempre que possível, por meio da ferramenta de ALM. O acesso deve ser restrito e toda instalação de objetos e alteração de objetos de banco deverá passar por aprovação ou verificação da equipe de infraestrutura correspondente. Os scripts de banco de dados e outros procedimentos necessários ao correto funcionamento da aplicação deverão ser configurados no ALM para que sejam executados automaticamente quando for feita a publicação do software por essa ferramenta.

5.4 Gerência de configuração

Um projeto de software é composto por muitos artefatos: planos, códigos executáveis, códigos fontes, modelos, relatórios, manuais e outros documentos, sendo alguns deles resultados oficiais do projeto enquanto outros servem de utilização interna dentro da equipe do projeto. Esses artefatos podem evoluir ao longo do processo de desenvolvimento do projeto e, dessa forma, é importante que sejam armazenados e controlados, uma vez que pode ser necessário recuperá-los.

O processo de Gestão de Configuração do Projeto tem como objetivo, portanto, identificar e documentar as características dos artefatos criados ou utilizados no projeto, controlar as alterações, armazenar e registrar o processamento das modificações e verificar a integridade desses artefatos ao longo do projeto. Com isso, evita problemas como perda de versões de artefatos, reaparecimento de erros, falta de controle de alterações e de visibilidade em relação ao estado do projeto em um dado momento, retrabalho e perda de produtividade.

É importante destacar que a gestão de configuração se aplica tanto ao projeto quanto ao produto de software, uma vez que os projetos podem atuar sobre um novo produto ou sobre um mesmo produto já existente. A gestão da configuração dos produtos e projetos de Tecnologia da Informação deverá estar integrada à ferramenta de ALM, para os mais diversos artefatos.

O detalhamento das diretrizes da gestão de configuração são as listadas no guia técnico de gestão de configuração. Entretanto, independentemente do sistema de controle de versão utilizado, para o correto funcionamento do fluxo de desenvolvimento de software destacado nessa MDAS, os artefatos do software deverão seguir o fluxo conhecido como GitFlow, conforme explicado abaixo.

5.4.1 Git Flow

É um modelo de ramos do repositório que se adequa às situações cotidianas no desenvolvimento de software e oferece resposta organizada e padronizada a essas ocorrências.

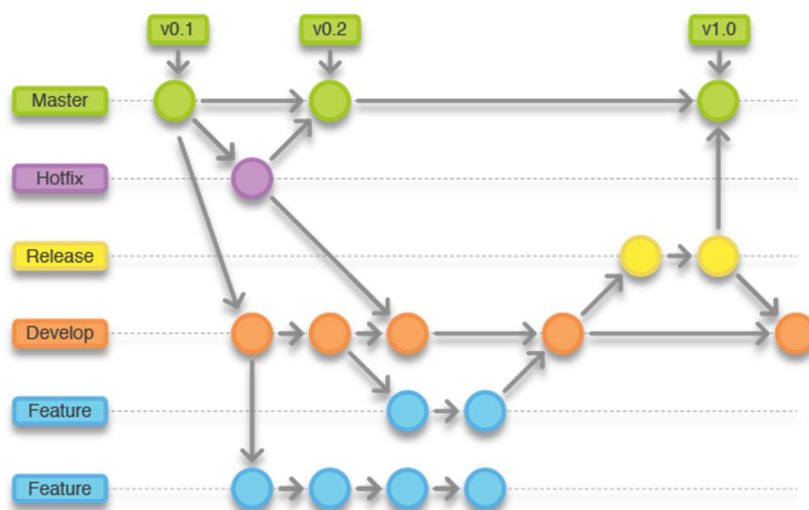


Figura 9 – Modelo de ramos do Git Flow

Os ramos Master e Develop são, respectivamente, os da produção e do desenvolvimento. Esses ramos devem estar íntegros, com código compilável, durante todo o tempo, pois, sempre que houver uma integração de código em qualquer um deles, a ferramenta de ALM irá atuar de forma a publicar o sistema no ambiente correspondente.

Sempre que o desenvolvimento de uma funcionalidade começar, time de desenvolvimento deverá criar um ramo do tipo Feature. Esse ramo deverá ser utilizado para armazenar o código temporário gerado enquanto o trabalho estiver sendo feito. Ao concluir a construção das funcionalidades, os ramos Feature deverão ser encerrados e o código integrado ao ramo de desenvolvimento.

Assim que concluída uma sprint ou release, o Scrum Master deverá criar um ramo do tipo Release para integrar o código constante no ramo de desenvolvimento. Com a criação de ramos desse tipo, a ferramenta de ALM irá entender que deverá publicar uma versão no ambiente de homologação e irá informar aos interessados que existe uma versão disponível para validação ou homologação.

Por fim, uma vez homologada a release, a própria ferramenta de ALM irá atuar para promover o código constante no ramo de Release para o ramo Master e, com isso, o código validado e homologado será publicado na produção e estará disponível para os usuários.

Existe, ainda, um tipo de ramo Hotfix usado para a correção de erros identificados em sistemas na produção. Esse ramo permite que o desenvolvedor atue diretamente na correção do problema e o código seja integrado diretamente no ramo Master, corrigindo, imediatamente, o erro em produção. As alterações feitas para corrigir erros identificados em produção deverão ser integradas a todos os ramos do produto, uma vez que o erro corrigido provavelmente acontece também nas demais versões do software.

5.5 Gerência de ordens de serviço de TI

Sempre que uma demanda for desenvolvida por uma empresa terceirizada, será necessário abrir uma ordem de serviço de forma a oficializar a solicitação da prestação do serviço. No modelo desta MDAS, uma ordem de serviço será aberta para cada história de usuário repassada à empresa terceirizada para construção.

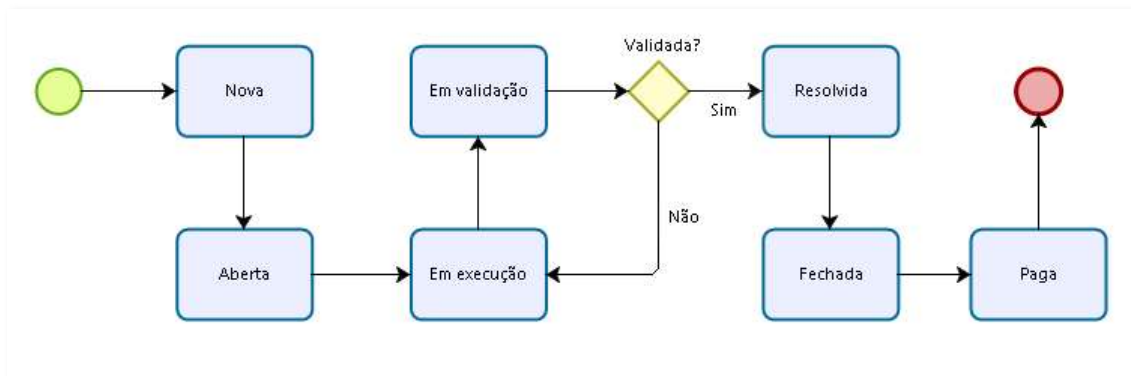


Figura 10 – Diagrama de estados de histórias de usuário e de ordens de serviço

Abaixo, o fluxo de estados das histórias de usuário e das ordens de serviço:

1. A história de usuário, quando adicionada ao backlog do processo, terá o estado NOVA;
2. Uma vez que seja atribuída à equipe de desenvolvimento, o estado da história é alterado para ABERTA;
3. Quando a equipe de desenvolvimento começar a trabalhar na construção da história, deverá alterar o estado desta para EM EXECUÇÃO;
4. Ao fim da construção, o Scrum Master deve enviar a história para ser validada pelo PO, promovendo o estado desta para EM VALIDAÇÃO;
5. Se a história não for validada pelo PO, voltará para o estado EM EXECUÇÃO. Caso contrário, passará ao estado RESOLVIDA.
6. Uma vez verificado que foi construída completamente e que a medição de tamanho for validada, a ordem de serviço é enviada para pagamento e a história tem seu estado alterado para FECHADA.

A alteração de estado deve ser feita de forma automática e tempestiva, diretamente no quadro kanban da ferramenta de ALM.

Para efeitos de uniformização da informação, os estados da história serão os mesmos que os da ordem de serviço. Entretanto, para efeitos de gestão contratual, são relevantes apenas os estados ABERTA, RESOLVIDA e FECHADA para as ordens de serviço. O estado ABERTA representa a oficialização da demanda à empresa terceirizada. O estado RESOLVIDA indica que a ordem de serviço foi concluída. O estado FECHADA indica que a ordem de serviço teve seu tamanho validado e foi enviada para pagamento.

A ferramenta de ALM (com ou sem a ajuda de ferramenta auxiliar) deverá ser capaz de recuperar as ordens de serviço / histórias de usuário de todos os produtos, permitindo a filtragem de resultado por parâmetros como estado e período de tempo.

5.6 Acompanhamento de projetos (escritório de projetos)

O escritório de projetos é uma área que tem como missão facilitar a execução de projetos dentro da organização por meio do planejamento, organização e controle das atividades dos projetos da melhor forma possível. Cabe ao escritório fornecer informações acerca da execução dos projetos, registrando lições aprendidas, gerando estatísticas, promovendo a comunicação entre as partes interessadas, retirando impedimentos e dificuldades, entre outras funções.

Cabe a esse departamento, ainda, ao fim de cada sprint, receber os dados da execução das tarefas de forma que seja possível o acompanhamento e o controle dos projetos, com o objetivo da consolidação de uma base histórica de execução de projetos pela CGETI.

6 Artefatos

A tabela abaixo define a documentação necessária e os produtos que deverão ser gerados durante os processos de atendimento a demandas de software conforme a MDAS da SUSEP.

Os templates e os campos a serem preenchidos em cada um dos artefatos encontram-se anexo a este documento.

A responsabilidade pela elaboração de cada artefato é descrita nos processos da MDAS citados nas seções anteriores. Os artefatos listados podem variar entre projetos, de acordo com as respectivas complexidades destes, e suas obrigatoriedades são definidas em coluna específica na tabela abaixo.

Tabela 4 – Lista de artefatos da metodologia

Artefato	Momento em que é gerado / revisado	Obrigatório
Backlog do produto	Abertura da demanda	Sim
	Entendimento da demanda	
	Encerramento de sprint	
	Encerramento de release	
Visão da demanda	Entendimento da demanda	Não
Roadmap da demanda	Planejamento da execução da demanda	Não
Backlog do release	Abertura de release	Não
	Encerramento de sprint	
	Encerramento de release	
Backlog da sprint	Abertura de sprint	Sim
	Encerramento de sprint	
Ata de reunião	Abertura de sprint	Não
	Encerramento de sprint	

7 Guias técnicos

Os guias técnicos descrevem padrões, orientações e roteiros a serem seguidos em determinados processos da MDAS. Por exemplo, há guias de orientação de como utilizar a arquitetura de desenvolvimento, padrões de nomenclatura dos objetos de banco de dados, roteiro para estimar tamanho de software etc.