



METODOLOGIA DE DESENVOLVIMENTO E MANUTENÇÃO DE SISTEMAS - MDMS

Histórico de Alterações

Versão	Data	Descrição	Responsável
1.1	24/04/2020	Atualização da MDMS. Adicionar e detalhar prazos e indicadores.	Alexandre e Vinicius

Sumário

1. Introdução	3
1.1. Objetivo	3
1.2. Bibliografia.....	3
2. Princípios Gerais.....	4
2.1. Agregar valor	4
2.2. Simplicidade	4
2.3. Foco	4
2.4. Homologação Frequente.....	4
2.5. Prazo Fixo.....	4
2.6. Inspeção e Adaptação	5
3. Papéis e responsabilidades	5
4. Modelo de Referência	6
4.1. Modelo.....	6
4.2. Atividades de planejamento	7
4.2.1. Atividade: Construir a visão do produto	7
4.2.2. Atividade: Planejar o Roadmap das Releases.....	9
4.3. Atividade de construção	11
4.3.1. Atividade: Planejar/Executar a Sprint	11
4.4. Atividade de Transição.....	14
4.4.1. Atividade: Disponibilizar a Release.....	14
4.4.2. Atividade: Preparar e Realizar Treinamentos.....	16
4.4.3. Atividade: Reunião de Encerramento da Release ou Projeto.....	17
Conclusão	19
Referências	20

1. Introdução

1.1. Objetivo

Definir a Metodologia de Desenvolvimento e Manutenção de Sistemas (MDMS) da Escola Superior do Ministério Público da União - ESMPU com base nos seguintes fundamentos:

- **Práticas ágeis:** A MDSM utilizará práticas ágeis, conforme tendência na Administração Pública Federal em virtude da maior tempestividade na entrega de resultados que agregam valor ao negócio e considerando o cenário da ESMPU.
- **Enxuta:** as Metodologias Ágeis já definem os princípios e práticas. Dessa forma, pretende-se definir os pontos centrais específicos do desenvolvimento de sistemas da ESMPU, para tornar o processo eficiente e eficaz.
- **Evolutiva e empírica:** a metodologia tem o objetivo de ser simples e objetiva e, a partir dos ciclos de desenvolvimento de sistemas, pode agregar outros elementos, de acordo com a prática e a maturidade da equipe.
- **Referências técnicas:** a utilização de referências a manuais técnicos, ferramentas e *templates* no decorrer das fases da metodologia, para que possa agregar o conhecimento técnico necessário.

1.2. Bibliografia

- Guia do Scrum, versão 2017 (GSCRUM)
<http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>
- Guia de Projetos de Software Com Práticas de Métodos Ágeis Para o SISP, versão 1.0, 2015 (GPSMA-SISP):
http://www.sisp.gov.br/guiaagil/wiki/download/file/Guia_de_Projetos_Ageis
- Roteiro de Métricas de Software do SISP, versão 2.2, 2016 (RMS-SISP):
https://www.governoeletronico.gov.br/documentos-e-arquivos/Roteiro_de_Metricas_de_Software_do_SISP_-_v2.0.pdf

2. Princípios Gerais

2.1. Agregar valor

O valor agregado é a percepção pelo Gestor de Sistema/ Dono do produto que o objetivo específico do negócio foi alcançado. Deverá ser certificado que as expectativas estão alinhadas e os produtos entregues agregam valor ao negócio.

2.2. Simplicidade

Retire tudo que for desnecessário, pois onera e sobrecarrega o processo, as pessoas e o produto, bem como as partes interessadas.

2.3. Foco

Estabeleça um foco nos itens de solução, nas necessidades e nos problemas, para que a energia de trabalho se concentre apenas nos objetivos e acordos firmados.

2.4. Homologação Frequente

As partes interessadas devem ser envolvidas desde o início do projeto, para que as validações aconteçam de forma parcial na entrega do produto. As entregas realizadas dessa forma, possibilitam um ciclo de feedback mais curto e um maior envolvimento e contribuição das partes interessadas e equipe.

As sprints não são um desenvolvimento em cascata com um prazo definido, pois se as partes interessadas forem acionadas apenas no início e o produto for entregue somente ao final, pode gerar uma expectativa frustrada e vários problemas na homologação. Portanto, é essencial que os interessados participem ativamente do desenvolvimento do produto, pois a realização de homologações parciais mitiga os riscos e facilita a homologação final do produto.

2.5. Prazo Fixo

Um princípio já bastante pontuado em práticas ágeis é o conceito de o prazo ser definido e fixo antes de seu início. Respeitar este conceito é primordial para o sucesso das Metodologias Ágeis, ou seja, não permitirá exceções de qualquer natureza. É importante salientar que, ao final do prazo, o produto entregue deve estar “pronto”, ou seja, deve satisfazer o conceito de finalizado para todas as partes definidas pelo projeto como responsáveis pela verificação e validação (geralmente o time de desenvolvimento e o gestor de sistema).

Uma confusão comum é achar que, ao final do prazo, o produto ainda será homologado pelo gestor de sistema.

2.6. Inspeção e Adaptação

Verificar se o objetivo desejado está sendo alcançado é importante para acompanhar e medir o sucesso do projeto. Caso os aspectos de um processo sejam desviados dos limites aceitáveis, o resultado do produto pode tornar-se inaceitável, com isso, o processo ou o material produzido deve ser ajustado. O ajuste deve ser realizado o mais rápido possível para diminuir outros desvios.

3. Papéis e responsabilidades

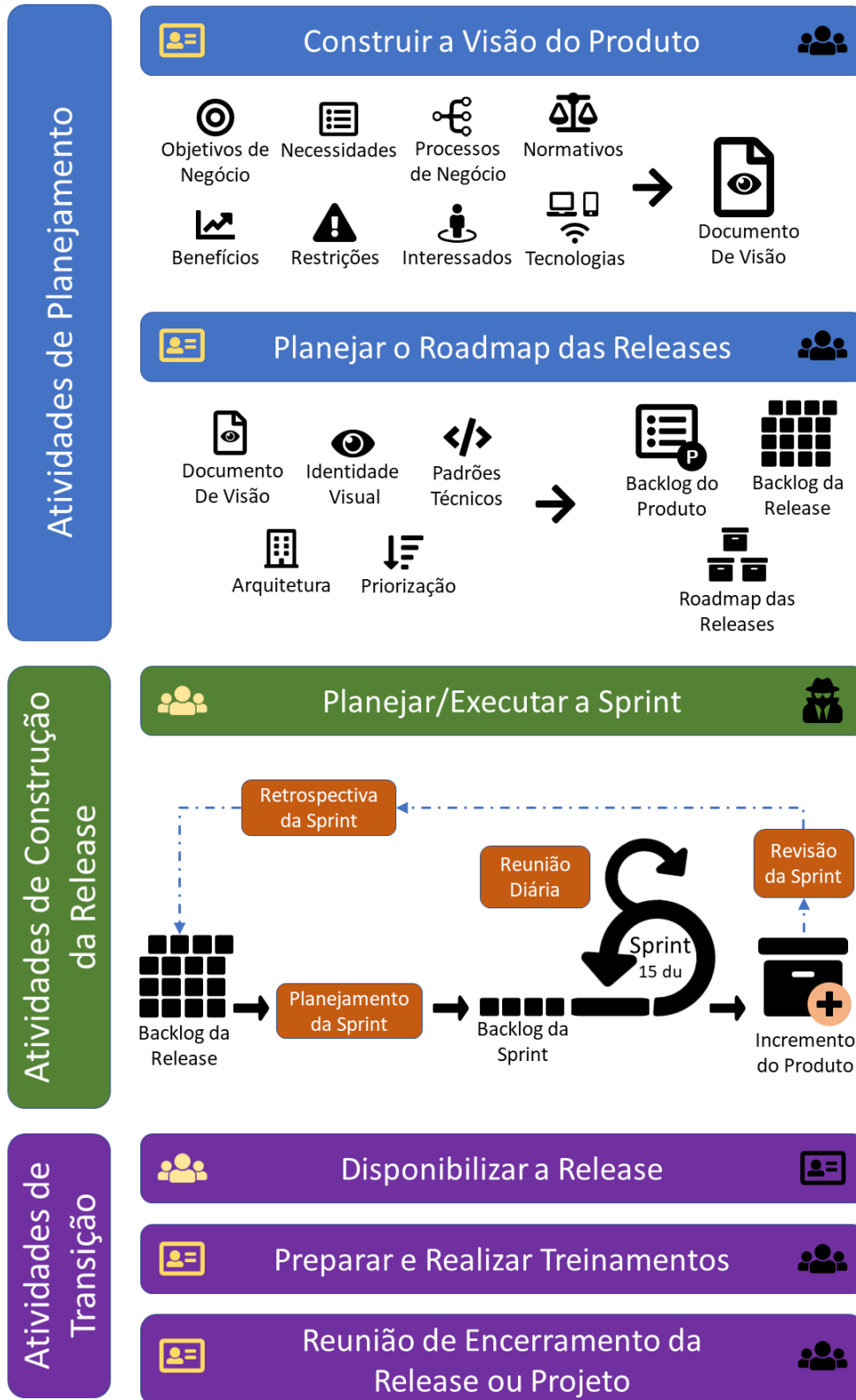
O processo de desenvolvimento de software exige a participação de papéis importantes que desempenham ações definidas, tais como os responsáveis, coparticipantes ou colaboradores.

Os papéis do processo são:

Papel	Área	Descrição
Gestor de Sistema (Product Owner)	Negócio	Representante da área de negócio com conhecimento suficiente para definir e priorizar requisitos do negócio e responder aos questionamentos da equipe de desenvolvimento. É o representante do cliente e sua atuação tem por finalidade garantir a entrega do valor esperado através do produto final do projeto. Existe apenas um gestor de sistema, independentemente do número de pessoas da área de negócio envolvidas no projeto. Possui capacidade de tomada de decisão ou está em contato direto com o chefe da unidade.
Líder do Projeto (Scrum Master)	TI	Responsável por promover e suportar o processo Scrum, ajudando todos a entenderem a teoria, as práticas, as regras e os valores do framework. Tem uma forte participação na remoção de impedimentos para o progresso do Time de Desenvolvimento e na facilitação dos eventos Scrum. Cumpre tipicamente as atividades de gerente de projeto, como monitoramento e controle, e relaciona-se com todos os envolvidos e atividades.
Time de Desenvolvimento	TI	Time multifuncional e auto-organizado que consiste de profissionais que realizam o trabalho de entregar um incremento potencialmente liberável do produto "Pronto" ao final de cada Sprint. Individualmente os integrantes do Time de Desenvolvimento podem ter habilidades especializadas e área de especialização, mas a responsabilidade pertence ao Time de Desenvolvimento como um todo.

4. Modelo de Referência

4.1. Modelo



4.2. Atividades de planejamento

4.2.1. Atividade: Construir a visão do produto

Objetivo: Construir a visão do produto onde são descritas as necessidades, expectativas, objetivos específicos de negócio e proposta de solução para o projeto.

Prazo: 10 dias úteis.

Pré-requisito: um projeto de desenvolvimento de sistemas só pode ter início quando o negócio é bem conhecido e tem-se, com clareza, um processo de negócio que deve ser automatizado ou um problema que pode ser resolvido com uma solução de software.

Portanto, são pré-requisitos:

- Processo de negócio mapeado. Existem casos específicos em que o mapeamento poderá ser opcional devido à experiência e maturidade da equipe da área negocial e à clareza da solução pretendida;
- Disponibilidade e envolvimento de pessoas ligadas à área negocial com conhecimento e experiência para construir a visão do produto.

Descrição: Nessa atividade, entende-se que existe um problema a ser resolvido ou uma oportunidade a ser aproveitada. Para atender a essa demanda, um produto será construído por meio de um projeto.

O documento de visão deve ser construído buscando responder as seguintes questões:

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

As necessidades e expectativas do cliente, relacionadas aos objetivos de negócio, serão utilizadas para definir o conjunto de características-chaves (ou macrofunções) do produto final e estarão fortemente alinhadas. Ela é um tipo de requisito de negócio de alto nível que consiste numa declaração de um objetivo do negócio que a solução deverá trazer ao ambiente do negócio, é um estado ou condição que o negócio deve satisfazer para alcançar a sua visão. (BABOK, 2011).

Os objetivos específicos do negócio devem estar sempre no foco do projeto (agregação de valor para o cliente), pois são a chave para tomar boas decisões sobre custo, escopo e prazos, tanto no início e depois, quando as mudanças acontecem. Os objetivos de negócio devem atender simultaneamente a seis critérios (Chiavenato, 2004):

- Focado em um resultado e não em uma atividade;
- Consistente, ou seja, alinhado coerentemente a outros objetivos e demais metas da organização;
- Específico e bem definido;
- Mensurável, ou seja, quantificável e objetivo;
- Relacionado com um determinado período, como dia, semana, mês e número de anos;
- Alcançável.

Responsável:

- Gestor de Sistema.

Coparticipantes:

- Time de Desenvolvimento.

Entradas:

- PDTI;
- Objetivos e processos de negócio;
- Necessidades;
- Normativos.

Saídas:

- Documento de Visão.

Referências técnicas: a elaborar.

Ferramentas: Microsoft Word.

4.2.2. Atividade: Planejar o Roadmap das Releases.

Objetivo: Construir o planejamento de entregas ou plano cronológico de liberação dos releases (versões do produto). Esse plano cronológico é chamado de roadmap.

Prazo: 5 dias úteis.

Pré-requisito: Atividade “Construir a Visão do Projeto”.

Descrição: Planejar o roadmap significa dividir objetivos de negócio e as características importantes ou macrofunções do produto em partes entregáveis (releases), por ordem de prioridade.

As características-chaves do produto são extraídas do Documento de Visão e devem estar alinhadas com as necessidades, as expectativas do cliente e, por fim, aos objetivos específicos de negócio. É nesse momento em que as partes envolvidas discutem e consolidam as características-chaves no backlog do produto e, a partir do planejamento, definem o roadmap das releases e o primeiro backlog da release.

Orientações para a elaboração do plano de entregas dos diversos releases:

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

Ao final de cada release, o roadmap poderá ser revisado e atualizado.

Além disso, é função dessa atividade definir quais padrões técnicos centrais a solução adotará. Em geral, esses padrões já foram previamente elaborados em projetos internos. São exemplos de padrões técnicos:

- Arquitetura de software;
- Identidade visual;
- Padrões relativos à qualidade de software, como cobertura de testes, análise estática e dinâmica de código, estilo de código;
- Uso de frameworks, componentes e ferramentas;
- Versão utilizada das metodologias, documentos e templates;

- Configuração padrão das máquinas do time de desenvolvimento.

Por fim, o plano de entrega das versões é o registro cronológico e preordenado da entrega de cada parte da solução, devidamente aprovado pelo gestor de sistema e clientes.

Notas:

- Defina em conjunto com a equipe de desenvolvimento e analista de infraestrutura de TI, caso necessário, a arquitetura Inicial do software e sua identidade visual. Neste momento, pode ser realizado um evento de workshop do produto para discussão dessas definições;
- Agregação de valor ao negócio: a fórmula do sucesso para os projetos ágeis (vide os princípios do manifesto ágil) é priorizar pelo valor de negócio, buscando sempre a simplicidade (na especificação, implementação e execução);
- Os releases não têm um prazo fixo, apenas as iterações, as quais devem sempre entregar valor ao cliente. Não se recomenda executar iterações apenas técnicas. Pode até mesclar revisões técnicas, mas sempre incluindo entregas que agreguem valor ao cliente.

Responsável:

- Gestor de Sistema.

Coparticipantes:

- Time de Desenvolvimento.

Entradas:

- Documento de Visão;
- Metodologias, padrões e manuais técnicos.

Saídas:

- Backlog do Produto;
- Roadmap das Releases;
- Backlog da Release.

Referências técnicas:

- Manual Técnico

Ferramentas: Microsoft Word.

4.3. Atividade de construção

4.3.1. Atividade: Planejar/Executar a Sprint

Objetivo: Construir e entregar um incremento do produto em um prazo pré-definido, de acordo com o backlog da sprint, com a meta de completar o backlog da release.

Prazo: 15 dias úteis.

Pré-requisito: Atividade “Planejar o Roadmap das Releases”.

Descrição: O planejamento e execução da sprint, de forma geral, seguirão os passos e eventos descritos no “Guia do Scrum, versão 2017”. Aqui, destacaremos definições específicas do processo de desenvolvimento da ESMPU ou itens do Scrum que merecem reforço. Como visto no modelo de referência, a atividade ocorre com base no backlog da Release e segue as etapas:

1. Planejamento da Sprint;
2. Execução da Sprint;
3. Revisão da Sprint;
4. Retrospectiva da Sprint.

Uma release é um conjunto de funcionalidades que entrega valor ao negócio, e pode ser constituído de uma ou mais sprints. O backlog da release servirá dois propósitos:

1. Agrupar entregas com efetivo valor ao negócio. Nem sempre uma sprint apenas terá a capacidade de produzir um incremento do produto que agregue valor ou viabilize um processo de negócio de ponta a ponta.
2. Em um modelo de terceirização, uma ordem de serviço contempla uma release. Só existirá pagamento mediante a entrega e validação de todos os itens do backlog da release. Além disso, a contratação definirá um teto percentual de alterações do tipo Refinamento que não serão objeto de faturamento. Refinamentos são quaisquer mudanças ocorridas sobre uma função transacional ou de dados já previamente trabalhadas na release corrente (seja por meio de uma inclusão, alteração ou exclusão), provocadas pelo aprofundamento, detalhamento e complementação de requisitos durante o processo de desenvolvimento (RMS-SISP, p. 51).

Em um modelo com terceirização, é interessante limitar o tamanho máximo de uma release, de forma a evitar que uma ordem de serviço tenha uma duração muito extensa sem o devido pagamento pelo serviço prestado. Por esse motivo, temos a primeira definição:

Definição 01: uma release será constituída de, no máximo, 3 sprints. Totalizando 45 dias úteis.

Em seguida, passaremos às etapas.

Planejamento da Sprint: o planejamento da Sprint responde as seguintes questões:

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

O Time de Desenvolvimento é o único a dimensionar a capacidade produtiva e a definir quais itens do backlog serão atendidos na sprint de 15 dias úteis, baseado no mais recente incremento do produto e no desempenho passado. Se o Time de Desenvolvimento determinar que tem excesso ou falta de trabalho, os itens do backlog poderão ser renegociados com o gestor de sistema.

Execução da Sprint: o Time de Desenvolvimento usa a Reunião Diária para inspecionar o progresso em direção ao objetivo da Sprint e se o progresso tende na direção de completar o trabalho do backlog da sprint. Ela aumenta a probabilidade de se atingir o objetivo da Sprint mediante a resposta às questões:

- O que eu fiz ontem que ajudou o Time de Desenvolvimento a atingir a meta da sprint?
- O que eu farei hoje para ajudar o time de desenvolvimento a atingir a meta da sprint?
- Eu vejo algum obstáculo que impeça a mim ou o time de desenvolvimento no atingimento da meta da sprint?

É importante frisar que, alinhado aos princípios de homologação frequente e prazo fixo a etapa de execução da sprint entrega um incremento do produto “Pronto”, já validado pelos interessados no ambiente de homologação e com a chancela do gestor de sistema.

Caso um item da sprint seja disponibilizado pelo time de desenvolvimento conforme acordado e o gestor de sistema e/ou as partes interessadas não homologuem dentro do prazo estabelecido, o item é marcado como não concluído e retornado ao backlog da release. O prazo máximo para homologação da sprint pelo gestor de sistema é de 10 dias úteis.

Se for constatada a falta de envolvimento de alguns atores no processo, o scrum master deverá notificar o gestor de TI para que possa agir de forma a contornar a situação. Porém, deve estar claro que o prazo e a entrega de incrementos somente no estado “Pronto” devem sempre ser respeitadas.

Revisão da Sprint: é realizada no final da Sprint para inspecionar o incremento e adaptar o backlog da release ou do produto se necessário. Esta é uma reunião

informal, não uma reunião de status, e a apresentação do incremento destina-se a motivar e obter feedback e promover a colaboração. Inclui os seguintes elementos:

- Os participantes incluem o Time Scrum e as partes interessadas convidadas pelo Product Owner;
- O Product Owner esclarece quais itens do Backlog do Produto foram “Prontos” e quais não foram “Prontos”;
- 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 Time de Desenvolvimento demonstra o trabalho que está “Pronto” e responde as questões sobre o incremento;
- O Product Owner discute o Backlog da Release ou Produto tal como está. Ele projeta os prováveis alvos e datas de entrega baseado no progresso até a data (se necessário);
- 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;
- Revisão de como o mercado ou o uso potencial do produto pode ter mudado e o que é a coisa mais importante a se fazer a seguir; e,
- Revisão da linha do tempo, orçamento, potenciais capacidades, e mercado para a próxima versão esperada de funcionalidade ou de capacidade do produto.

O resultado da revisão da sprint é um Backlog da Release ou Produto revisado que define os prováveis itens para a próxima Sprint. O Backlog da Release ou Produto pode também ser ajustado completamente para atender novas oportunidades.

Retrospectiva da Sprint: é uma oportunidade para o Time Scrum inspecionar a si próprio e criar um plano para melhorias a serem aplicadas na próxima Sprint. Tem como propósito:

- Inspecionar como a última Sprint foi em relação às pessoas, aos relacionamentos, aos processos e às ferramentas;
- Identificar e ordenar os principais itens que foram bem e as potenciais melhorias; e,
- Criar um plano para implementar melhorias no modo que o Time Scrum faz seu trabalho;

Responsável:

- Time de Desenvolvimento.

Coparticipantes:

- Scrum Master;
- Gestor de Sistema

Entradas:

- Backlog da Release;
- Metodologias, padrões e manuais técnicos.

Saídas:

- Incremento do Produto;
- Termo de aceite comercial;
- Backlog da Release e/ou Produto atualizado;
- Lições aprendidas;
- Ações de melhoria.

Referências técnicas:

- Padronização do Repositório com Git
- Oracle Apex
- Build e Deploy Automático com Jenkins e Artifactory

Ferramentas:

- OpenProject ou outra que venha a substituí-la

4.4. Atividade de Transição

4.4.1. Atividade: Disponibilizar a Release

Objetivo: Implantar a release no ambiente de produção, incluindo a disponibilização de documentações e configurações de acesso.

Prazo: 5 dias úteis.

Pré-requisito: Atividade “Planejar/Executar a Sprint”.

Descrição: após o aceite técnico e comercial da release, o Product Owner indica o momento ideal para disponibilizar a release no ambiente de produção. Ela pode ser disponibilizada imediatamente ou, por questões normativas ou estratégicas, por exemplo, uma release pode ser agendada para estar disponível para o público-alvo em uma data certa.

A disponibilização não envolve, somente, a implantação no ambiente de produção. Ela marca o início da operação da release e pode contemplar as seguintes tarefas:

- Publicação de documentações, como manuais técnicos, de treinamento e códigos-fonte;
- Configuração de acessos;
- Carga de dados;
- Configuração de rotinas automáticas de e-mail, banco de dados e *jobs*;

- Publicação de notícias sobre a release.

Responsável:

- Gestor de Sistema

Coparticipantes:

- Time de Desenvolvimento
- Analista de Infraestrutura

Entradas:

- Release homologada;
- Definições de “Pronto”

Saídas:

- Release em operação.

Ferramentas:

- Git
- Jenkins
- Oracle Apex
- Portal Intranet / Internet
- E-mail
- Scripts

4.4.2. Atividade: Preparar e Realizar Treinamentos.

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

Prazo: Não há um prazo formal estabelecido, no entanto, haverá um cronograma de treinamento.

Pré-requisito: Atividade “Planejar/Executar a Sprint”.

Descrição: A atividade de preparar e realizar treinamentos é realizada em paralelo com a próxima sprint, de modo a não comprometer o prazo global do projeto. Porém, existem casos em que o treinamento mobiliza muitas pessoas do projeto e pode ser incluído como uma entrega de uma sprint.

Orientações para o treinamento e repasses:

- O gestor de sistema deve planejar a estratégia de transferência de conhecimento para gestores de negócio e usuários do produto de software;
- O gestor de sistema deve, com o apoio da equipe de desenvolvimento, produzir o manual de usuário e o material de treinamento, quando necessário;
- O gestor de sistema deve produzir o material específico para treinamento dos gestores de negócio ou usuários;
- O líder do projeto, junto com a equipe de desenvolvimento, deve planejar a estratégia de transferência de conhecimento para equipes de suporte e sustentação;
- As equipes de suporte e sustentação devem ser treinadas pela equipe de desenvolvimento para receber e tratar os suportes de 1º, 2º e 3º nível da solução. Esse treinamento é gerenciado pelo líder de projetos;
- Devem ser repassados, pela equipe de desenvolvimento, todos os documentos necessários ao suporte e manutenção da solução;
- Deve ser dado apoio, pelo líder do projeto, para atualização ou inclusão de informações da solução no catálogo de serviços de TI;
- Adotar formas e recursos necessários para a transferência efetiva de conhecimento entre as equipes em alinhamento às normas da instituição e contratos envolvidos.

Notas:

- A transferência de conhecimento pode ser realizada por sala de aula com instrutor, por vídeos disponibilizados na intranet ou canal oficial da instituição ou ainda por cursos EAD;
- Há alguns casos em que o treinamento pode ser fornecido mediante contratação específica.

Responsável:

- Gestor de Sistema.

Coparticipantes:

- Líder do Projeto;
- Time de Desenvolvimento.

Entradas:

- Release homologada;
- Documentações da release.

Saídas:

- Manuais de usuário;
- Manuais técnicos;
- Roteiros de atendimento;
- Vídeos e PDFs tutoriais;
- Cursos presenciais e a distância.

Ferramentas:

- Microsoft PowerPoint;
- Microsoft Word.
- Ambiente virtual de aprendizagem
- Sistema de videoconferência

4.4.3. Atividade: Reunião de Encerramento da Release ou Projeto

Objetivo: Realizar verificações do processo e do produto para confirmar que a release e/ou projeto podem ser formalmente encerrados.

Prazo: 2 horas.

Pré-requisito: Atividades de Planejamento, Construção da Release e Transição.

Descrição: O líder do projeto, juntamente com o gestor de sistema e o time de desenvolvimento, deve verificar se o processo de desenvolvimento foi cumprido.

Orientações para esta atividade:

- Confirmar que a solução está operacional e foi transferida para a equipe de suporte e sustentação, verificando que:
 - Foram desenvolvidos todo o material de suporte (manuais, vídeos e scripts de atendimento);
 - Equipes de suporte (1º, 2º e 3º nível) estão preparadas para dar o suporte necessário para a continuidade da solução;
- Avaliar os critérios de aceitação da qualidade do produto;
- Analisar e definir ações para o tratamento dos defeitos residuais do release/projeto;
- Confirmar se os usuários foram treinados e estão aptos a utilizar a solução;

- Aprovar os resultados do release / projeto;
- Caso seja o último release do projeto, aprovar o encerramento do projeto.

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

- Atualizar o status do projeto como encerrado, caso não haja mais releases do projeto;
- Incluir o novo sistema no catálogo de serviços da TI.

Responsável:

- Líder do projeto.

Coparticipantes:

- Gestor de Sistema;
- Time de Desenvolvimento.

Entradas:

- Releases em operação;
- Documentações da release / projeto.

Saídas:

- Termo de Encerramento da release /projeto.

Ferramentas:

- Microsoft Word
- OpenProject
- Sistema Eletrônico de Informações – SEI

Conclusão

O Modelo de Desenvolvimento e Manutenção de Sistemas evidencia as melhores práticas ágeis adotadas na Administração Pública Federal ajustadas a realidade da ESMPU. As entregas constantes, o processo simplificado, os papéis enxutos e bem definidos e o foco no valor a ser entregue aumentam as probabilidades de sucesso.

Projetos de desenvolvimento e manutenção bem sucedidos dependem de um amplo entendimento do Gestor de Sistema acerca dos processos negociais e do problema a ser resolvido. A equipe de desenvolvimento, em complemento, compromete-se com o valor da entrega e autorregula-se para atender o conceito de pronto a cada Sprint. Uma boa visão do produto aliada aos ciclos de desenvolvimento de releases para fornecerem pacotes de soluções.

Para tanto, o scrum master deve garantir que a equipe dispõe dos meios e recursos necessários para o desempenho de suas tarefas, atuando na dissolução dos impedimentos e zelando pela participação de todos os membros da equipe. Os processos, técnicas, ferramentas e *templates* são imprescindíveis para a qualidade do produto final.

Portanto, o MDMS é ferramenta importantíssima para a disciplina de desenvolvimento de soluções de software da ESMPU, devendo ser utilizada tanto para a execução direta quanto indireta destes projetos. Adere-se a realidade da ESMPU por ser um framework capaz de suportar diferentes leiautes cujas melhores peças dependem dos recursos e maturidade de cada instituição.

Referências

BABOK, GUIA. Um guia para o Corpo de Conhecimento de Análise de Negócios. **IIBA-International Institute of Business Analysis Versão**, v. 2, 2011.

CHIAVENATO, Idalberto; SAPIRO, Arão. **Planejamento estratégico**. Elsevier Brasil, 2004.