

METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE - MDS

Processos de TIC

Data: 28/06/2019

Versão 1.2

Presidente do Tribunal de Justiça do Estado do Paraná**Desembargador Adalberto Jorge Xisto Pereira****1º VICE-PRESIDENTE****Desembargador Wellington Emanuel Coimbra de Moura****2º VICE-PRESIDENTE****Desembargador José Laurindo de Souza Netto****CORREGEDOR-GERAL****Desembargador José Augusto Gomes Aniceto****CORREGEDOR****Desembargador Mario Helton Jorge****SECRETÁRIA DO TRIBUNAL DE JUSTIÇA****Maria Alice de Carvalho Panizzi****SUBSECRETÁRIA DO TRIBUNAL DE JUSTIÇA****Juliana Moreno Dias Paredes****SUPERVISOR DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO****Desembargador Marcelo Gobbo Dalla Dea****DIRETOR DO DEPTO. DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO****Nelson Joaquim Santos****EQUIPE TÉCNICA DE ELABORAÇÃO****Alexandre Sypniewski Sbalqueiro
Alessio Roman Junior
Aluizio Carlos Wanderley Grochocki
Daniel Targa Dias Anastacio
Johnatan Daniel Fromholz Lima
Pablo Tavares
Tatiane Luiz Gomes da Silva****COMITÊ DE GESTÃO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO (CGETIC)****Danilo Kovalechyn
Fábio de Araújo
Fábio Luís Bruch
Jean Paul Bonneville
Luiz Fernando Moletta Alves
Magno Mario Bayer Filho
Márcio Mortensen Wanderley
Maria Esther Aguirra de Moraes
Nelson Joaquim Santos
Rafael Coninck Teigão
Rolf Mertens Junior**

Brasil. Tribunal de Justiça do Estado do Paraná. Departamento de Tecnologia da Informação e Comunicação.

Documentação de Processos de TIC / Tribunal de Justiça do Estado do Paraná, Departamento de Tecnologia da Informação e Comunicação. - Curitiba: TJPR, 2017. 30 p.: il.

1. Tecnologia da informação. 2. Gestão pública. 3. Administração Pública. 4. Gestão e Governança. 5. Processos de TI.

HISTÓRICO DE ALTERAÇÕES

DOCUMENTO			
Descrição	Documento de Processo de TIC para o fluxo da Metodologia de Desenvolvimento de Software no DTIC.		
Objetivo	Este documento descreve as atividades e procedimentos adotados para padronização da Metodologia de Desenvolvimento de Software no DTIC do TJPR.		
Responsável	Cleverton Mayer	Divisão	DIS
Criado em	15/02/2017	Revisão	Anual

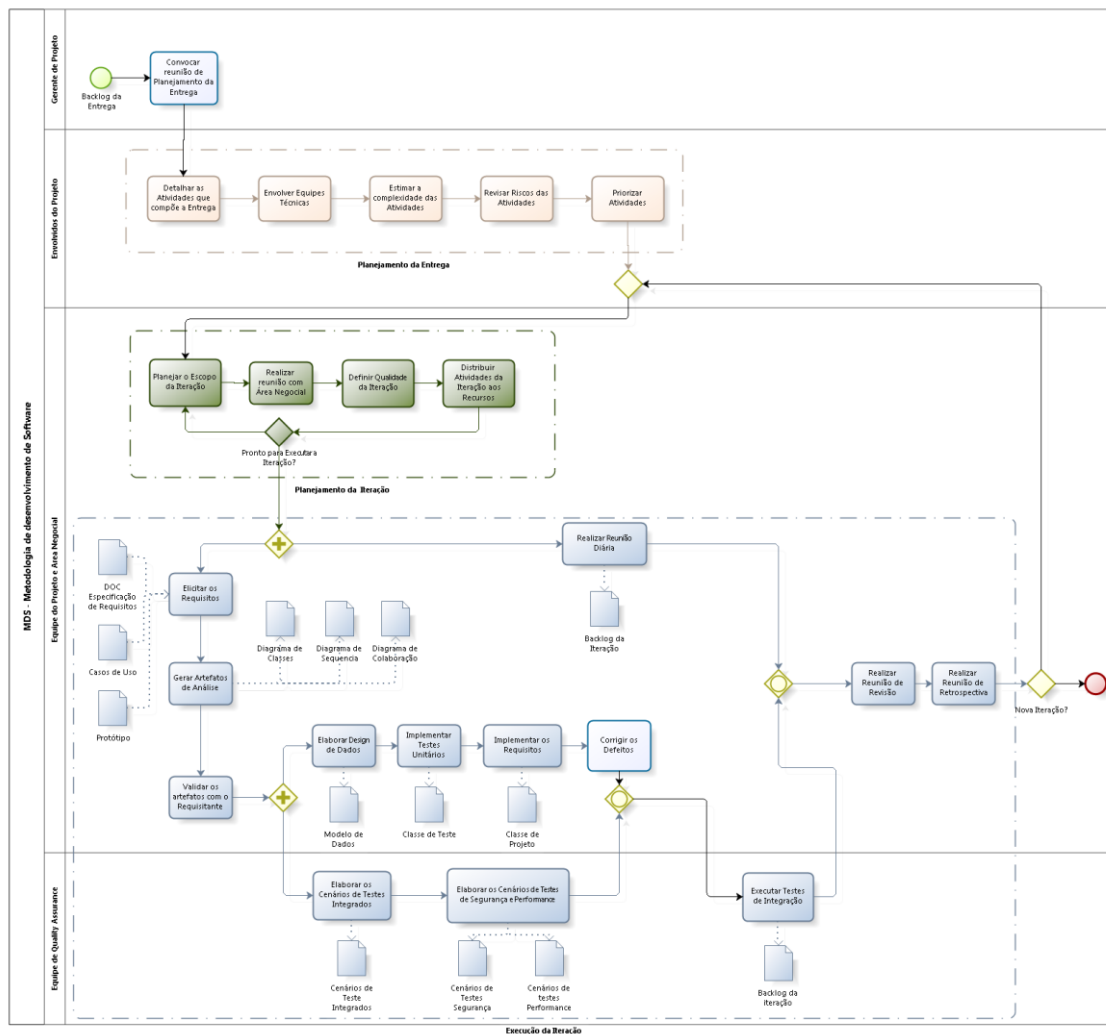
VERSIONAMENTOS			
Versão	Data	Autor	Descrição
1.0	15/02/2017	Equipe Governança de TIC	Criação e Revisão do Documento
1.1	07/02/2019	Equipe Governança de TIC	Atualização da Cúpula Diretiva
1.2	28/06/2019	Equipe Governança de TIC	Atualizado Logotipo TJPR conforme Resolução TJPR 227/2019

Sumário

1	METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE – MDS.....	5
1.1	Diagrama do Processo.....	5
1.2	Descrição das Atividades.....	6
1.2.1	Convocar reunião de Planejamento da Entrega	6
1.2.2	Detalhar as Atividades que compõe a Entrega	7
1.2.3	Envolver Equipes Técnicas	8
1.2.4	Estimar a complexidade das Atividades	9
1.2.5	Revisar Riscos das Atividades	10
1.2.6	Priorizar Atividades	11
1.2.7	Planejar o Escopo da Iteração.....	12
1.2.8	Realizar reunião com Área Negocial	13
1.2.9	Definir Qualidade da Iteração	14
1.2.10	Distribuir Atividades da Iteração aos Recursos.....	15
1.2.11	Elicitar os Requisitos	16
1.2.12	Realizar Reunião Diária.....	17
1.2.13	Gerar Artefatos de Análise	19
1.2.14	Validar os artefatos com o Requisitante	21
1.2.15	Elaborar Design de Dados.....	22
1.2.16	Implementar Testes Unitários	23
1.2.17	Implementar os Requisitos	24
1.2.18	Corrigir os Defeitos.....	25
1.2.19	Elaborar os Cenários de Testes Integrados.....	26
1.2.20	Elaborar os Cenários de Testes de Segurança e Performance	27
1.2.21	Executar Testes de Integração	28
1.2.22	Realizar Reunião de Revisão	29
1.2.23	Realizar Reunião de Retrospectiva	30

1 METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE – MDS

1.1 Diagrama do Processo



1.2 Descrição das Atividades

1.2.1 Convocar reunião de Planejamento da Entrega

Objetivo:

- Envolver todos os participantes do projeto para alinhar as expectativas das entregas do projeto, com o foco na entrega mais próxima de ser realizada. Realizar no mínimo uma reunião para cada entrega.

Entradas:

- Backlog das Entregas;
- Plano de Projeto e Estudo de Viabilidade;
- Cronograma do Projeto.

Descrição das Atividades:

- Criar uma lista de participantes do projeto (ver a relação de envolvidos no Plano de Projeto, os stakeholders, e a equipe interna do projeto);
- Elaborar uma agenda e pauta para apresentação, utilizando as técnicas de reuniões para aumentar a produtividade;
- Agendar uma sala com os recursos e mídias necessários, e enviar a convocação pelo MS-Exchange do TJPR, com no mínimo 7 dias de antecedência;
- Elaborar a ata da reunião e publica-la para todos os participantes e partes interessadas do projeto.

Saídas:

- Reunião convocada.

1.2.2 Detalhar as Atividades que compõe a Entrega

Objetivo:

- Identificar todas as atividades e recursos necessários para executar a entrega proposta.

Entradas:

- Plano de Projeto;
- Cronograma.

Descrição das Atividades:

- Ler novamente o conteúdo do escopo e não escopo do projeto, identificando o que faz parte e o que não faz parte do produto desta entrega, com relação a características e funcionalidades;
- Validar, junto aos envolvidos, se as informações levantadas até o momento estão corretas;
- Apresentar ou discutir sobre a solução adotada para o produto da entrega.

Saídas:

- Especificação Técnica da Entrega criado.

1.2.3 Envolver Equipes Técnicas

Objetivo:

- Comunicar e Antecipar às demais equipes do DTIC sobre os recursos que serão necessários.

Entradas:

- Plano de Projeto;
- Cronograma;
- Especificação Técnica da Entrega.

Descrição das Atividades:

- Solicitar as equipes técnicas (desenvolvimento, infraestrutura, governança, homologação, produção e atendimento aos usuários) um parecer sobre a entrega em questão;
- Deve ser levando em conta os assuntos:
 - Objetivos e requisitos de segurança;
 - Possíveis ataques e ameaças da solução adotada;
 - Itens de infraestrutura: hardware, software, banco de dados, relatórios, redes, telecomunicações, infraestrutura física (quando for o caso);
 - Itens de suporte e atendimento ao usuário;

Saídas:

- Especificação Técnica da Entrega atualizado.

1.2.4 Estimar a complexidade das Atividades

Objetivo:

- Estimativa das atividades para validar o dimensionamento da entrega.

Entradas:

- Plano de Projeto;
- Cronograma;
- Especificação Técnica da Entrega.

Descrição das Atividades:

- Realizar ou revisar a estimativa das atividades, utilizando a métrica adotada pelo órgão.

Saídas:

- Estimativa realizada;
- Cronograma atualizado.

1.2.5 Revisar Riscos das Atividades

Objetivo:

- Revisar os riscos e validar se existem mais riscos não previstos.

Entradas:

- Plano de Projeto;
- Cronograma;
- Especificação Técnica da Entrega.

Descrição das Atividades:

- Revisar os riscos identificados ao projeto;
- Revisar os riscos das atividades técnicas;
- Revisar os riscos das atividades da equipe e ambiente.

Saídas:

- Riscos do Projeto Atualizado.

1.2.6 Priorizar Atividades

Objetivo:

- Identificar aquelas atividades de maior prioridade, criticidade e benefício para a entrega e aos usuários.

Entradas:

- Plano de Projeto;
- Cronograma;
- Especificação Técnica da Entrega.

Descrição das Atividades:

- Identificar aquelas atividades que geram mais benefício ao usuário, considerando as necessidades técnicas da solução.

Saídas:

- Plano de Projeto atualizado com a lista de atividades priorizadas;
- Cronograma atualizado.

1.2.7 Planejar o Escopo da Iteração

Objetivo:

- Planejar as atividades que serão executadas nas iterações previstas.

Entradas:

- Plano de Projeto;
- Cronograma
- Especificação Técnica da Entrega.

Descrição das Atividades:

- Dentre as atividades identificadas e detalhadas em cada entrega, identificar aquelas que fazem parte da iteração em questão.

Saídas:

- Proposta da iteração criada.

1.2.8 Realizar reunião com Área Negocial

Objetivo:

- Revisar com a Área Negocial a proposta da iteração.

Entradas:

- Plano de Projeto;
- Cronograma;
- Proposta da Iteração.

Descrição das Atividades:

- Organizar a Pauta e discutir com os representantes da área negocial explicando como será realizado as iterações, principalmente envolvendo-o sobre o conteúdo das iterações, já alinhando as expectativas;
- Elaborar a ata da reunião e publica-la para todos os participantes e partes interessadas do projeto.

Saídas:

- Ata da reunião publicada.

1.2.9 Definir Qualidade da Iteração

Objetivo:

- Identificar os critérios de aceitação de cada iteração.

Entradas:

- Plano de Projeto;
- Cronograma;
- Proposta da Iteração.

Descrição das Atividades:

- Identificar juntamente com os representantes da área comercial, quais serão os critérios de aceitação de cada parte da iteração, ou após um conjunto de iterações. Pode ser por funcionalidade, benefício ao usuário, itens de menus, etc;
- Distinguir o que é essencial, aceitável e desejável.

Saídas:

- Lista Critérios de Qualidade identificada.

1.2.10 Distribuir Atividades da Iteração aos Recursos

Objetivo:

- Mostrar os componentes que precisam ser executados para a equipe de trabalho.

Entradas:

- Plano de Projeto;
- Cronograma;
- Proposta da Iteração.

Descrição das Atividades:

- Atividades Alocadas pela equipe;
- Cronograma atualizado.

Saídas:

- Lista Critérios de Qualidade identificada.

1.2.11 Elicitar os Requisitos

Objetivo:

- Trabalhar com a Área Requisitante para aprender sobre o domínio da aplicação, quais as funcionalidades ele deverá oferecer, além das questões de desempenho, qualidade, restrições, entre outros.

Entradas:

- Plano de Projeto;
- Cronograma;
- Proposta da Iteração.

Descrição das Atividades:

- Realizar entrevista com o requisitante para identificar os requisitos funcionais e não funcionais ou criar fóruns específicos (caso necessário);
- Elaborar Lista de Requisitos (DER);
- Elaborar os Diagramas de Casos de Uso (DCU) e/ou Protótipos de Tela (PT);
- Analisar as fronteiras do software;
- Realizar verificações de validade, consistência, completeza, realismo e facilidade dos requisitos preliminares;
- Descrever e detalhar os requisitos;
- Classificar e priorizar requisitos;
- Elaborar Documento de Especificação de Requisitos (DER);
- Solicitar aceite preliminar dos artefatos gerados.

Saídas:

- Documento Especificação de Requisitos (ER);
- Diagrama de Casos de Uso (DCU);
- Protótipos de Tela.

1.2.12 Realizar Reunião Diária

Objetivo:

- Também chamada de Daily ou Standup meeting, comunicar a equipe sobre as atividades em execução. Duração máxima 30 min.

Entradas:

- Agenda a ser definida pela equipe

Descrição das Atividades:

- A ideia é que esta reunião ocorra sempre no mesmo horário e no mesmo local durante as Iterações, cada membro da equipe provê respostas para cada uma destas três perguntas:
 - O que você fez ontem, ou desde a última reunião diária?
 - O que você fará hoje, ou até a próxima reunião diária?
 - Quais obstáculos estão no seu caminho?
- Concentrando-se no que cada pessoa fez ontem e no que ela irá fazer hoje, a equipe ganha uma excelente compreensão sobre que trabalho foi feito e que trabalho ainda precisa ser feito;
- O foco das perguntas e de suas respostas não deve ser um status dos itens, e a reunião diária não deve ser uma reunião de status, mas sim uma reunião para um alinhamento do que foi realizado e do que será realizado que poderá agregar valor aos trabalhos dos outros membros;
- Além de compartilhar as produções, os impedimentos e obstáculos devem ser levados muito a sério durante as reuniões diárias, pois estes podem interferir seriamente no objetivo da Iteração e devem ser removidos o mais rápido possível;
- Sendo assim, o mais importante da reunião diária é a comunicação, a eliminação de outras reuniões e a identificação de impedimentos para o

desenvolvimento fluir sem barreiras. Estas ações promovem a tomada rápida de decisões e melhoram o nível de conhecimento de todos acerca do projeto;

- A reunião diária também é uma inspeção do progresso na direção da meta da Iteração, mas não pode ser realizada para resolver problemas, o que ela pode é provocar outras reuniões subsequentes para realizar adaptações ao trabalho que está por vir na Iteração, otimizando a probabilidade de que a equipe alcance a sua Meta;
- Dica: Qualquer pessoa pode participar da reunião diária, porém apenas a equipe pode falar e interferir na reunião.

Saídas:

- Lista de ações para a equipe.

1.2.13 Gerar Artefatos de Análise

Objetivo:

- Classificar e priorizar requisitos, detectar e resolver conflitos entre requisitos; confirmar e validar os requisitos com o cliente.

Entradas:

- Documento Especificação de Requisitos (ER);
- Diagrama de Casos de Uso (DCU);
- Protótipos de Tela.

Descrição das Atividades:

- Identificar Casos de Uso (UC);
- Descrever Casos de Uso (UC);
- Atualizar o documento de Especificação de Requisitos (DER);
- Controlar as mudanças nos requisitos através de ferramentas de rastreabilidade;
- Analisar os casos de uso buscando mapear os elementos estruturais e comportamentais que influenciam na definição da arquitetura final do sistema;
- Especificar o diagrama de classes do sistema;
- Especificar o diagrama de sequência dos casos de uso;
- Especificar o diagrama de colaboração dos casos de uso;

- Especificar a arquitetura do sistema a partir dos seus elementos estruturais e comportamentais.

Saídas:

- Documento Especificação de Requisitos (ER);
- Diagrama de Casos de Uso (DCU);
- Especificação de Caso de Uso (ECU);
- Protótipos de Tela;
- Diagrama de Classes;
- Diagrama de Sequência;
- Diagrama de Colaboração;
- Documento de Arquitetura do Software (DAS).

1.2.14 Validar os artefatos com o Requisitante

Objetivo:

- Detectar os erros contidos nos documentos da especificação para evitar alterações corretivas com custos excessivos.

Entradas:

- Especificação de Requisitos (ER);
- Especificação de Caso de Uso (ECU).

Descrição das Atividades:

- Enviar especificações para a Área Requisitante e, se for o caso, para outros envolvidos no processo;
- Atualizar especificações de acordo com as observações do requisitante ou de acordo com os demais órgãos envolvidos;
- Receber e registrar especificações validadas pelo requisitante;
- Validar os casos de uso críticos gerando as evidências dessa avaliação;
- Apresentar e revisar a arquitetura do sistema.

Saídas:

- Especificação de Requisitos (ER – validado);
- Especificação de Caso de Uso (ECU – validado).

1.2.15 Elaborar Design de Dados

Objetivo:

- Analisar os casos de uso do sistema;
- Analisar os casos de uso do sistema para identificar os elementos estruturais (diagrama de classes) e comportamentais (diagramas de colaboração e de sequência) como insumos para a elaboração da especificação da arquitetura final do sistema.

Entradas:

- Especificação de Requisitos (ER);
- Especificação de Caso de Uso (ECU).

Descrição das Atividades:

- Analisar os casos de uso buscando mapear o modelo de dados necessário, relatórios futuros e necessidades na infraestrutura;
- Mapear classes do diagrama de classes para o modelo de dados;
- Avaliar o desempenho do modelo de dados e o acesso aos dados;
- Definir tabelas de referência;
- Definir regras para a integridade referencial e de dados;
- Definir, se necessário, rotinas de automação para o monitoramento dos dados;
- Também contemplar requisitos de performance, segurança e auditoria;
- Revisar o Documento de Arquitetura do Software (DAS);
- Criar Solicitação de Mudança (se necessário).

Saídas:

- Documento de Modelo de dados criado;
- Documento de Arquitetura do Software (DAS - atualizado);
- Solicitação de Mudança (se necessário).

1.2.16 Implementar Testes Unitários

Objetivo:

- Implementação dos Cenários de testes unitários.

Entradas:

- Especificação de Requisitos (ER);
- Especificação de Caso de Uso (ECU);
- Documento de Arquitetura do Software (DAS - atualizado).

Descrição das Atividades:

- Criação dos procedimentos de testes para validar as rotinas que serão implementadas.

Saídas:

- Cenários de Testes Unitários (CTU) implementados.

1.2.17 Implementar os Requisitos

Objetivo:

- Codificação dos componentes e arquitetura planejada;

Entradas:

- Especificação de Requisitos (ER);
- Especificação de Caso de Uso (ECU);
- Documento de Arquitetura do Software (DAS - atualizado).

Descrição das Atividades:

- Produzir código-fonte dos casos de uso da iteração, implementando operações, estados, associações e atributos das classes de implementação;
- Avaliar o código, verificando se está adequado à sua finalidade, e realizar testes nas unidades, executando os casos de testes unitários;
- Geração de Scripts para banco de dados;
- Versionamento das fontes;
- Realizar testes nas unidades buscando identificar defeitos no software;
- Caso ocorram falhas, deve ser iniciada a atividade de correção de defeitos.

Saídas:

- Código fonte e bibliotecas versionados.

1.2.18 Corrigir os Defeitos

Objetivo:

- Corrigir os defeitos encontrados na execução dos testes.

Entradas:

- Componentes implementados com defeitos.

Descrição das Atividades:

- Revisar o código fonte de forma a corrigir os defeitos que provocaram falhas na execução dos testes.

Saídas:

- Componentes sem os defeitos.

1.2.19 Elaborar os Cenários de Testes Integrados

Objetivo:

- Criar os casos de testes, bem como os procedimentos necessários à sua aplicação, de acordo com os requisitos de software.

Entradas:

- Especificação de Requisitos (ER);
- Especificação de Caso de Uso (ECU);
- Documento de Arquitetura do Software (DAS - atualizado);
- Diagrama de Sequência;
- Diagrama de Colaboração.

Descrição das Atividades:

- Identificar e priorizar cenários de teste;
- Projetar e priorizar casos de teste;
- Documentar e organizar casos de teste.

Saídas:

- Casos de Testes Integrados (CTI).

1.2.20 Elaborar os Cenários de Testes de Segurança e Performance

Objetivo:

- Criar os casos de testes relativos a garantia da segurança e performance na aplicação perante a arquitetura adotada.

Entradas:

- Especificação de Requisitos (ER);
- Especificação de Caso de Uso (ECU);
- Documento de Arquitetura do Software (DAS - atualizado);
- Diagrama de Sequência;
- Diagrama de Colaboração.

Descrição das Atividades:

- Identificar e priorizar cenários de teste de segurança e performance;
- Projetar e priorizar casos de teste de segurança e performance;
- Documentar e organizar casos de teste de segurança e performance;
- Elaborar procedimentos de teste de segurança e performance;
- Definir base de dados para execução dos testes de segurança e performance.

Saídas:

- Casos de Testes de Segurança e Performance (CTSP).

1.2.21 Executar Testes de Integração

Objetivo:

- Executar os casos de testes de integração de componentes e módulos do software em desenvolvimento.

Entradas:

- Casos de Testes Integrados (CTI).

Descrição das Atividades:

- Preparar o ambiente de teste;
- Preparar o banco de dados de teste;
- Executar os casos de testes de integração de acordo com os procedimentos definidos no projeto do teste;
- Avaliar e registrar os resultados obtidos;
- Revisar os casos de teste quando for identificada necessidade de aperfeiçoamento ou ajuste;
- Reportar a ocorrência e a gravidade das falhas ao Gerente de Projetos para monitoramento e controle.

Saídas:

- Builds testados;
- Registro de Testes (RT) de Integração.

1.2.22 Realizar Reunião de Revisão

Objetivo:

- Cerimônia destinada para demonstrar ao cliente os itens concluídos. Geralmente, tem no máximo 4 horas de duração.

Entradas:

- Meta da Iteração

Descrição das Atividades:

- A melhor maneira de fazer isso é justamente realizar uma apresentação, ao estilo demonstração, de todos os itens concluídos ao Cliente/Área negocial. Com isso ele poderá conferir e avaliar o que está sendo considerado pronto, levando em conta o que está sendo entregue versus o que deveria ser entregue;
- Dica: Um membro da equipe pode realizar a demonstração dos itens de Backlog prontos ao Cliente/Área negocial, ou o próprio Cliente/Área negocial pode navegar no sistema e fazer por si só a avaliação dos itens entregues;
- Dica: A Revisão não é um momento para testar os itens entregues, mas sim de apresentação, conferência e avaliação dos itens de acordo com a indicação de pronto de cada um dos itens;
- Dica: Não gaste tempo montando uma apresentação do sistema ou das funcionalidades prontas, utilize o próprio sistema, mesmo que em um ambiente de desenvolvimento, para apresentar os itens concluídos;
- Dica: Ao final da Revisão, pode-se ter novos itens não planejados para a próxima Iteração, gerando com isso entradas muito importantes para as futuras reuniões de planejamento da Iteração.

Saídas:

- Backlog atualizado.

1.2.23 Realizar Reunião de Retrospectiva

Objetivo:

- Cerimônia destinada para a equipe voltar no tempo e discutir os trabalhos da última iteração.

Entradas:

- Documento com Ações de Melhoria das Iterações Anteriores;
- Desvios e Impedimentos, Dificuldades ou Problemas Encontrados.

Descrição das Atividades:

- Inspeccionar como a última iteração foi em relação às pessoas, aos relacionamentos, aos processos e às ferramentas;
- Identificar e ordenar os principais itens que foram bem e as potenciais melhorias;
- Criar um plano para implementar melhorias no modo que o dono do produto, gerente de projetos e a equipe de desenvolvimento fazem seu trabalho;
- Finalmente, a melhoria contínua do processo é importante para aumentar a qualidade da execução e, conseqüentemente, a garantia da qualidade do produto entregue;
- Recomendação:
 - Para facilitar a gestão de tempo e evitar reuniões com assuntos repetitivos as reuniões de demonstração e retrospectiva poderão ser agrupadas em uma única reunião.

Saídas:

- Lições Aprendidas;
- Ações de Melhoria.