

Este guia tem como objetivo elucidar o workflow de desenvolvimento visando desde o início de uma feature até o momento de conclusão, passando pelas etapas de levantamento de requisitos até a conclusão do desenvolvimento e assim a entrega destas.

A demanda se origina com os stakeholders, onde o product owner (PO) fica responsável em levantar os interesses dos stakeholders e então definir os fundamentos das demandas, objetivos e planos de release (pacote de versão que a demanda fará parte, na nossa equipe nós chamamos de *delivery*). Após a etapa de planejamento o PO cria a lista de requisitos (*backlog*) que irá conter as demandas por ordem de prioridade onde fica visível na ordem ascendentes, sendo as primeiras mais prioritárias em relação as últimas. Após a etapa de backlog o time trabalha em duas etapas de refinamento (*refining*), as duas geralmente são trabalhadas em paralelo, visando uma maior agilidade. As duas etapas são *prototipação* e *quebra técnica* a etapa de prototipação é a que irá definir como será apresentados os dados ou organizado visualmente, já a etapa quebra técnica é a organização técnica do que deve ser desenvolvido e a possibilidade técnica de atender a demanda, após essas duas etapas concluídas a demanda vai é passado para coluna (*refined*) a qual caracteriza que a demanda está pronta para estimativa da equipe de desenvolvimento e teste no início da planning.

A planning é realizada sempre no primeiro dia útil da sprint, a sprint é o período de desenvolvimento do pacote a ser entregue, no nosso projeto a sprint é de 15 dias, sendo o primeiro dia segunda-feira.

Durante a planning que será realizada com todos os colaboradores do time também é definido qual ambiente será desenvolvido a demanda e quais tarefas necessárias para que seja paralelizado o desenvolvimento com intuito de atingir a conclusão desta no pacote, a cerimônia da planning também é responsável por definir o quais demanda o time poderá entregar, podem ter alteração na release de entrega, dependendo bugs, issues ou demandas emergências que podem aparecer, assim como falha da estimativa, visto que é uma previsão de capacidade para desenvolver as tarefas.

A hierarquia das demandas é da seguinte forma:

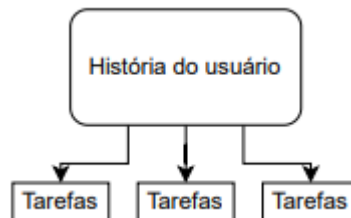
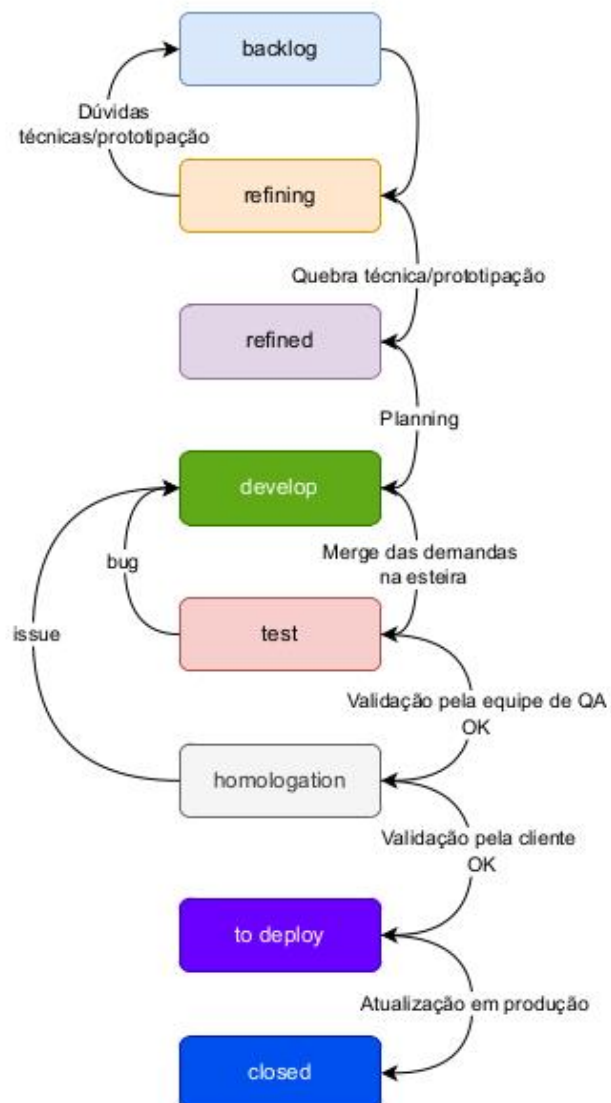
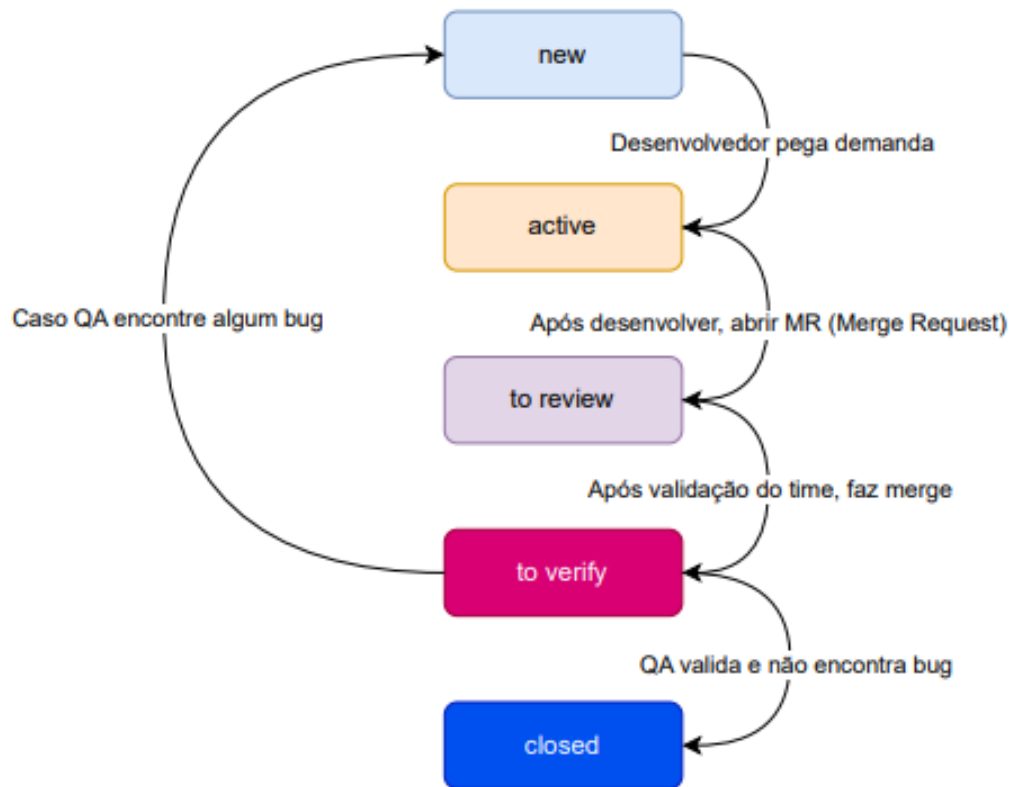


Diagrama do fluxo de uma história:



Após a planning e com todas as histórias de usuários estimadas, o time de desenvolvedor está apto a iniciar, as tarefas nascem na coluna “new” que representa que a tarefa não tem nenhum responsável é então quando o desenvolvedor pega essa tarefa ela deve ser passada para a coluna “active” e preenchido o responsável, indicando qual é o desenvolvedor que está trabalhando nela, após o desenvolvimento, deverá ser criada uma nova branch a partir da develop, o nome da branch deverá conter o número da tarefa (ex: 123) e então o desenvolvedor irá abrir um merge request (MR) das alterações, apontando para a branch develop, nesta equipe há uma necessidade de pelo menos dois desenvolvedores revisarem o MR e darem OK para então poder ser realizado o merge, essa revisão se torna necessária para garantir que esta demanda foi desenvolvida seguindo as melhores práticas de desenvolvimento e garantindo que não há nenhum erro grave no código, é então após esse double check que é realizado o merge, após o merge a tarefa passa para a coluna “to verify”, onde demonstra que a equipe de QA está apta a trabalhar nessa demanda e verificar possíveis falhas ou bugs no fluxo alterado/criado. Caso a equipe de teste encontre alguma falha será criada uma nova tarefa de bug relatando quais os passos para reproduzir e qual cenário esperado, essa tarefa de bug nascerá na coluna “new” e terá prioridade em relação a outras tarefas da história.

Diagrama contendo o fluxo de uma tarefa durante o desenvolvimento:



Há três branches principais no projeto, develop, homolog e master, a branch develop é a branch que os desenvolvedores trabalham, realizando o merge para equipe de qualidade validar. Já a branch homolog é a que o cliente valida antes de entrar em produção (branch master), e como mencionado a branch master é a branch de produção, ou seja, quando realizado o merge das coisas de homolog em master.

