

**Universidade Unicesumar**

Gustavo Kenji Sumi

**Plano de gerenciamento de configuração**

Maringá  
2020

O uso do Redmine, por se tratar de uma ferramenta com várias funcionalidades, requer uma pré-configuração para seu funcionamento adequado de acordo com o projeto, diferentemente do Trello, que é uma ferramenta que não necessita de muitas alterações para seu funcionamento.

Deve ser criado um novo projeto com descrição e definir seus modules marcando todas as opções. Após isso, deve ser criados os subprojetos Controle de funcionalidades, Controle de solicitação de mudanças, controle de desenvolvimento e Controle de testes. Esses subprojetos devem ser criados para um melhor funcionamento da ferramenta, deixando o projeto mais organizado, visto que cada uma delas, requer um processo diferente.

Deve ser cadastrado os membros do projeto na página de configuração, e para cada membro, deve ser especificado o seu papel, sendo fundamental, pois isso pode limitar as funcionalidades de cada membro. Com a criação dos subprojetos, devem ser configurados cada um deles, no Controle de funcionalidades deve ser criado novas issues, com o nome da funcionalidade e um prefixo para melhor rastreabilidade, como por exemplo: "FUNC-010 Matricula de alunos" sendo como backlog, as issues podem ser atribuídas a membros da equipe. Para ter controle dessas tarefas, deve ser configurado o workflow, alterando o que cada membro pode realizar nas Trackers, sendo uma máquina de estados. Para o desenvolvedor uma nova tarefa pode atribuir os estados de novo, em progresso, resolvido, fechado ou rejeitado, isso ajuda na descoberta de qual release ela foi implementada, para caso tenha um problema futuro com essa funcionalidade, é de fácil rastreamento, para analisar o problema. Com a criação dessas tarefas, devem ser informados na aba Wiki, um detalhamento da tarefa/ funcionalidade para entendimento de toda a equipe, a criação de fóruns permite comunicação constante entre seus membros.

Com as issues criadas, a aba Kanban view disponibiliza uma visão detalhada do andamento do projeto, como a criação de novas tasks, tasks em desenvolvimento, backlog e prontas. Outra funcionalidade para acompanhamento do desenvolvimento é a aba Gantt que mostra um diagrama/ gráfico, mostrando o avanço das etapas do projeto, intervalos de tempo de início ao fim. Nas configurações, ao selecionar version, deve ser criada as Sprints e Releases com data que irá ocorrer e inserir quais issues estarão incluídas nessa Sprint ou Release, para ter uma rastreabilidade em caso de problemas futuros, não informando a data de entrega das issues.

Os Releases ocorrerão em média a cada 6 meses, porém é previamente combinado com o cliente, para não afetar o seu funcionamento de trabalho. Por se tratar de um cliente único, a versão deve ser atualizada para o consumo da API. Voltando para os subprojetos, em Controle de desenvolvimento uma nova Tracker deve ser criada, com o nome Task incluindo esse subprojeto como utilizador, para ser utilizado e criar versions de Sprint para serem incluídas nas tarefas.

Caso uma Task criada seja muito genérico, deve ser criado uma nova task dividindo essa tarefa, informando a finalidade, Parent Task e a target version, porém a Task principal é onde o apontamento de horas e a pontuação é contabilizada.

Para o subprojeto Controle de solicitação de mudanças deve ser criado issues para cada mudança, que será analisado a fim de analisar a viabilidade dessas mudanças até chegar ao processo de Desenvolvimento. Dentre essas etapas inclui o custo orçamentário, análise das mudanças de funcionalidades e tempo para desenvolvimento.

A integração do Redmine com o Github é feita de forma livre, sendo que o repositório deve ser público, porém autenticado por meio de chaves SSH. É necessário gerar a chave no Redmine para vincular ao repositório e assim tendo acesso via SSH, para fazer o clone do repositório. Para atualizar e controlar o repositório é necessário criar um cron no Linux atualizando, limpando e fazendo um pull de cinco em cinco minutos buscando os novos commits. Assim como no Redmine deve ser utilizado o mesmo padrão para as mensagens e títulos para referenciar o commit com a funcionalidade descrita no Redmine, ajudando no processo de rastreabilidade.

A utilização do Google Docs para repositório de artefatos da especificação e produz artefatos como contêineres do Docker, deve ser de modo que o documento específico cuida da questão de rastreabilidade, de acordo com um determinado Release e um problema apresentado, é necessário saber quando esse Release com problema foi liberado, quais as mudanças, suas funcionalidades, componentes, quem

desenvolveu, seus componentes, testes automatizados utilizados, verificar a utilidade de cada um informando se necessário seu uso ou não, solicitações que deram origem a esse desenvolvimento dos artefatos, quem testou, homologou, aprovou. Ou seja, um histórico completo de cada passo realizado, quem foi responsável por cada detalhe. Por isso é necessário a especificação dos artefatos esteja como especificada nos requisitos que foram levantados, a fim de garantir que não haja duplo sentido e evitar problemas com a rastreabilidade.