

Nome: Lucas Matheus Nakasima Soares

RA: 18319432

Elaboração de um plano de gerenciamento de configuração

O uso do Redmine, por se tratar de uma ferramenta com várias funcionalidades, requer uma pré configuração para seu funcionamento adequado para um projeto, e definir os modules de acordo com as necessidades do projeto, como files, issue tracking, repository, entre outros. Sendo assim dentro do projeto principal serão criados subprojetos, com o intuito de manter um controle melhor para gerenciá-lo, assim torna-se mais fácil controlar as mudanças realizadas no mesmo. Nos subprojetos criados serão acrescentados uma nova issue, ou seja, novas tarefas que podem ser atribuídos ao membros da equipe, é importante mencionar que grandes tarefas devem ser divididas em tarefas secundárias, a fim de que o responsável não fique sobrecarregado em uma única tarefa, no qual a tarefa “pai” terá o apontamento de horas e pontuação. Por padrão na aba Wiki todas as funcionalidades devem ser claramente escritas, isto é, cada tarefa deverá conter um prefixo já definido (FN-0001) e descrição do que deve ser realizado, assim, se caso algum membro da equipe tenha alguma dúvida o mesmo apenas precisa ler as informações contidas nessa aba. Na aba Kanban view, será possível ter um visão detalhada do andamento do projeto, o qual conterá as etapas backlog, a fazer, fazendo, feito e dependência (atividades que dependem de outras para serem finalizadas).

Na integração do Redmine com o repositório no Github, a comunicação é feita de forma livre em repositórios que sejam públicos, mas no caso de ser privado a autenticação é feita com a utilização de chaves SSH. Para isso é gerado e configurado uma chave SSH no Redmine que fica vinculada ao repositório, sendo assim a partir dessa chave o servidor começa a ter acesso via o SSH possibilitando a verificação do repositório no Github, permitindo o Redmine fazer um clone do repositório em questão. Para a atualização e controle do repositório é criado um cron no Linux que atualiza o nosso repositório fazendo um clean e um pull de cinco em cinco minutos buscando todos os novos commits feitos pelos desenvolvedores. Para melhor rastreabilidade desses commits é mantido um padrão de mensagem onde é configurado no Redmine um meio de referenciar uma tag padrão estabelecida, ao qual ela esteja vinculada ao commit e a funcionalidade juntos, facilitando sua rastreabilidade, manutenção e levantamento dos dados.

Por fim a utilização do Google Docs, como repositório dos artefatos de especificação ocorrerá da seguinte forma: o documento em específico será responsável por “cuidar” da questão da rastreabilidade, ou seja, dado um release de software e um problema apresentado é preciso saber quando esse release foi liberado, quais são as funcionalidade que fazem parte dessa release, o que foi implementado a mais nessa release em relação a release anterior, quais são os componentes que a formam, quem trabalhou em cada componente desses, quais foram os testes automatizados e quais não foram e deveriam ter sido, onde estão as solicitações que deram origem as implementações dos artefatos que fizeram parte desse release, por quem foram testados, homologados, e aprovados. Por isso é importante mencionar como ter um grau de rastreabilidade ok, se não apresenta um controle de features e de aprovação de mudanças, além de associação de solicitação da feature que foi afetada ou do código com a solicitação que foi realizada. Dessa forma neste documento é importante que os artefatos do projeto estejam como especificados nos requisitos levantados, garantindo assim que o projeto esteja sob controle, gerando evidências que o projeto está ocorrendo em uma metodologia controlada.