

**ENGENHARIA DE SOFTWARE**

6º Semestre - Noturno

GUSTAVO MALHEIRO BOROSKI

1936719-2

**GERENCIAMENTO DE CONFIGURAÇÃO DO FIREFOX**

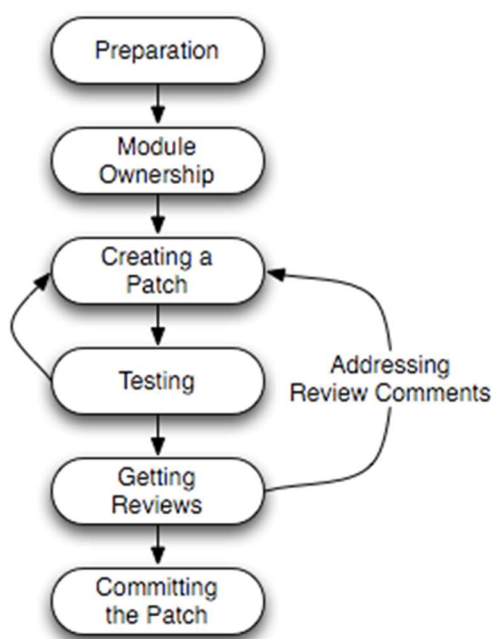
MARINGÁ

2020

Esse trabalho possui como principal ideia mostrar todo o processo de gerenciamento de configuração do FireFox.

Iniciando com o fluxo das versões do projeto, temos o seguinte diagrama retirado do site MDN web docs

Caminho utilizado nas entregas de atividades:



[https://developer.mozilla.org/en-US/docs/Mozilla/Developer\\_guide/How\\_to\\_Submit\\_a\\_Patch](https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide/How_to_Submit_a_Patch)

**Preparação:** deve ser realizado uma a criação de uma espécie de ticket, ou seja, sem essa “documentação”, o seu código não será testado.

**Propriedade do modulo:** Definir um “proprietário” que irá realizar a análise do seu código. Isso deve ser definido antes de envia-lo, com o objetivo de verificar se o mesmo tem disponibilidade de realizar essa atividade.

**Criando um patch:** Cada bug encontrado ou atividade que será realizada deve ter um patch individual, ou seja, quando for entregar essa atividade, deve prestar atenção na divisão das funções das mesmas. Se estiver muito grande e complexa, recomenda-se que divida em patch menores, assim aumentando a efetividade e rapidez nas análises.

**Testando:** Quando realizar uma mudança deve-se fazer novamente a execução dos testes, tanto os automatizados quanto os manuais.

**Enviando o Patch:** Ao enviar um patch é necessário verificar se possui a versão mais recente para evitar conflitos de mesclagens.

Nesse momento o Phabricator é utilizado para realizar as revisões de códigos.

**Endereçando comentários de revisão:** Após passar por todas essas etapas pode ocorrer do revisor pedir modificações em seu código. Os mesmos devem ser corrigidos e mandados para a análise novamente.

Agora temos os testes automatizados, que são mais utilizados em produção. Geralmente são encontrados na parte central do mozilla e são executados quando um certo número de commits são acumulados.

Todos os testes, independente do seu tipo, são classificados em uma tabela com os seguintes campos:

**Símbolo:** Definido em duas letras, qual equipamento será utilizado, e qual conjunto de testes.

**Nome:** Nome padrão para se referir ao teste.

**Tipo do arquivo:** Após criar o teste deve-se referenciar qual o tipo utilizado nesse teste.

**Plataforma:** Definir qual a plataforma que será utilizada, Windows, Android, IOS e etc...

**Processo:** Podendo ser Pai/Criança/Permitir. Significa que cada um será executado na sua ordem familiar, porém o permitir, possui o privilégio de ser executado em qualquer ordem desse processo.

**Meio Ambiente:** Pode ser JSShell/XPCShell/Java/Conteúdo/Navegador/Perfil Navegado.

**Privilégios:** Permite definir se o teste possui alto ou baixo nível de privilégios.

Posteriormente, entramos no desenvolvimento das documentações. Quando realizar a aquisição de um bug ou uma nova atividade, e estiver quase terminando de resolve-la, deve-se marcar no título da mesma a seguinte tag (dev-doc-needed).

Esse termo representa que um bug está pronto ou quase terminado, dessa forma os responsáveis pela documentação, vão saber com antecedência que aquelas atividades estão completas e no desenvolvimento da próxima documentação a pessoa responsável irá trocar a tag de dev-doc-needed para dev-doc-complete e colocando nos comentários links encaminhando para as páginas que representavam o bug.

Outro ponto importante são as tags que podemos inserir junto aos bugs. Isso irá facilitar na hora que o “analista” pegar esse bug para verificar a resolução do mesmo.

Colocar o que foi feito para resolver o problema pode ser uma boa pratica também.

### **Referências Bibliográficas**

MDN contributors. Developer guide, Disponível em: <[https://developer.mozilla.org/en-US/docs/Mozilla/Developer\\_guide](https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide)> Acesso em: 18/09/2020

MDN contributors. Contribuindo para o banco de códigos da Mozilla, Disponível em: <[https://developer.mozilla.org/pt-BR/docs/Mozilla/Developer\\_guide/Introduction](https://developer.mozilla.org/pt-BR/docs/Mozilla/Developer_guide/Introduction)> Acesso em: 18/09/2020

MDN contributors. How to submit a patch, Disponível em: <[https://developer.mozilla.org/en-US/docs/Mozilla/Developer\\_guide/How\\_to\\_Submit\\_a\\_Patch](https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide/How_to_Submit_a_Patch)> Acesso em: 18/09/2020

MDN contributors. Mozilla automated testing, Disponível em: <[https://developer.mozilla.org/en-US/docs/Mozilla/QA/Automated\\_testing](https://developer.mozilla.org/en-US/docs/Mozilla/QA/Automated_testing)> Acesso em: 18/09/2020

MDN contributors. Getting documentation updated, Disponível em: <[https://developer.mozilla.org/en-US/docs/Mozilla/Developer\\_guide/Getting\\_documentation\\_updated](https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide/Getting_documentation_updated)> Acesso em: 18/09/2020