



UNICESUMAR – UNIVERSIDADE CESUMAR
CENTRO DE CIÊNCIAS EXATAS TECNOLÓGICAS E AGRÁRIAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

Gerenciamento de Configuração

Rodrigo Kambara

MARINGÁ – PR
2020

www.unicesumar.edu.br

Avenida Guedner, 1610 - CEP 87050-390 - **Maringá** - Paraná • Avenida Santa Mônica, 450 - CEP 86027-610 - **Londrina** - Paraná
Rua Itajubá, 673 - CEP 81070-190 - **Curitiba** - Paraná • Rua Desembargador Westphalen, 60 - CEP 84036-350 - **Ponta Grossa** - Paraná

Python

Pull Request: no python, o fluxo de trabalho é baseado no pull request. É criado uma branch no Git, após a alteração fazer o push das alterações para o seu fork no origin, então é feito o pull request no repositório.

As boas maneiras no pull request: Para isso, siga diretrizes de estilo do Pythonm seguindo o código PEP 8. Caso tenha alguma discrepância, pode ser resolvido por um core developer que faz o merge das pull requests. Além disso, implemente testes para verificar se o pull request está funcionando como esperado. O pull request não será aceito sem testes adequados.

Estratégia de branching: O branching é feito de forma padrão, git fetch, checkout, add, commit e push. Existem várias versões de branches para cada feature. Branch em desenvolvimento: é o branch master que será o próximo a ser lançado, estando em desenvolvimento de qualquer tipo de alteração, como novas features, mudanças semânticas melhora na performance, correção de bugs. Branch de manutenção: a única mudança que é feita neste branch sem o debate são correções de bugs e a compatibilidade não deve ser quebrado em nenhum momento nas releases. Branch de segurança: uma branch que tem menos de 5 anos, mas não está mais em modo de manutenção, é considerado branch de segurança. As únicas mudanças que podem ser feitas são as correções nos problemas exploráveis por hackers como travamento, mudanças no privilégio, entre outros.

Merging: É feito após a aprovação do pull request.

Resolução de conflitos de merge: primeiramente checar o arquivo que está tendo conflito com o comando git status.

Após isso, editar o arquivo afetado e trazê-lo ao estado final pretendido. Por fim, fazer o commit destes arquivos, com o comando git add <nomedoarquivo> e git merge --continue

Code review: um dos problemas no processo de desenvolvimento do python é o code review. Quando busca o rastreador de bugs, é encontrado inúmeros problemas que possuem correções, mas não é capaz de fazer merge no repositório principal, pois ninguém fez a revisão da solução do problema.

Para fazer o code review, tem 6 passos:

- 1 - Pegar uma cópia do repositório do CPython, montando e fazendo testes.
- 2 - Checar o rastreador de bugs para ver os passos necessários para reproduzir o erro e confirmar se pode reproduzir o erro na versão do Python REPL.

3 - Fazer o checkout e aplicar pull request.

4 - Se as mudanças afetarem o arquivo C, executar a build novamente.

5 - Executar o Python REPL e verificar se consegue reproduzir o problema. Agora que o pull request foi aplicado, o problema deveria ser corrigido.

6 - Caso tenha tempo, execute os testes. Caso não tenha, execute testes só nos módulos em que as mudanças foram aplicadas. Deve-se ter um cuidado, já que está recomendando um pull request como merge ready.

Controle de versão: as duas escolhas principais para controle de versão são: git e mercurial.

Release: existem estágios de releases pre-alpha, alpha, beta, release candidate e final

- Pre-alpha: uma branch está neste estágio quando não foi feito nenhum release oficial. Não há nenhuma restrição nos commits.

- Alpha: serve como um lembrete dos desenvolvedores principais que precisam começar as alterações na semântica ou adicionar algo no Python que não deve ser adicionados na fase seguinte (Beta).

- Beta: depois da primeira release beta, os novos releases não são aceitos. Somente correções de bugs podem ser commitados.

- Release Candidate: só podem ser feitos correções que forem revisados por outros desenvolvedores principais. Geralmente, esses problemas devem ser graves o suficiente.

- Final: Somente o gerenciador de release podem fazer alterações na branch.

Rastreamento de requisitos: existe um plugin em python que pode ser usado como uma ferramenta de gerenciamento de requisitos, o Sphinx Traceability. O plugin pode gerar versões renderizados de objetos de documentações definidos, uma lista de objetos de documentação, matrizes rastreáveis entre objetos.