**Engenharia de Software**

6 Semestre - Noturno

Gustavo Kenji

1813702-2

**Gerenciamento de configuração em projetos Open Source**

MARINGÁ

2020

**Python Open Sorce**

**INTRODUÇÃO**

Um projeto open source se trata do código de um projeto que é disponibilizado de forma livre e gratuita para a comunidade contribuir, sendo assim é possível realizar modificações no código-fonte para aprimorar o projeto, seja do código ou da documentação. Como as tecnologias estão em constante evolução, o software é necessário evoluir junto, seja com novas funcionalidades, correção de bugs, ou melhorias.

**DESENVOLVIMENTO**

O Python é um desses projetos open source que está em constante aprimoramento, sendo que opera de acordo com a evolução da tecnologia, escrito em C se integrando com ecossistemas C e C++. Para a comunidade contribuir com ele, é necessário configurar e contribuir com um patch, possuir o GitHub para fazer o clone do código-fonte e criar uma nova branch local, para iniciar deve ser feito o fork do repositório CPython utilizando sua contado GitHub.

Após realizar as alterações no código e envia-las para o fork no git, deve ser feita uma pull request para o repository oficial do CPython. O merge request é uma forma para a equipe e membros tenham acesso para criação ou alterações do código no repositório, contendo descrição e widgets, permitindo uma interação entre os membros do projeto, melhorias de arquitetura, identificar problemas, bugs e padrões e posteriormente as alterações vão para o ramo principal, como são realizadas alterações, as vezes são incompatíveis com o restante do projeto, sendo necessário localizar quais são verificando o status.

Entretanto, deve ser tomado um certo cuidado ao realizar o merge request na branch master, seguindo padrões do Python, pois ao lançar a nova release o merge request pode ter algum problema/bug e todo o trabalho feito a partir disso pode ser perdido, para a identificação do erro. Para diminuir esses problemas, devem ser adotadas regras como por exemplo, deve ser implementada apenas uma funcionalidade. Isso faz com que a revisão do código seja feita de maneira rápida e prática, diminui os conflitos, menos correções geram menos trabalho, menor probabilidade de revisões identificar bugs ou melhorias a serem feitas.

Para os testes a maneira mais simples é executando o comando “./python -m test no diretório raiz do checkout, porém nem todos, aqueles que possuem recursos especiais são excluídos da execução. Para testes unitários é necessário especificar o nome do arquivo que será testado como argumento e o comando unittest, exemplo: “ ./python -m unittest -v test.test\_abc.TestABC”. Os testes são direcionados ao comportamento normal e erros, residem no Lib/test diretório, onde possuem um prefixo test\_nome.

Um ponto a se destacar é a integração contínua, possuindo máquinas específicas para a integração, abrangendo hardware e software, hospedando construtores por meio de uma branch ativa.

Para enviar as alterações para a branch é necessário realizar um push origin na branch e um pull request do fork. Para realizar um pull request do fork de um problema pré-existente é necessário fazer referência ao número do problema, caso seja um problema não relatado é necessário fazer referência no pull request, com detalhamento relevante a fim de evitar esforço.

Quando o pull request atingir um nível aceitável, ela será mesclada ou rejeitada, caso for aceita é enviado para o VCS e feito o release na próxima versão do Python, feito o backport nas versões mais antigas como correção de bug caso seja garantido. Quando uma branch possuir mais de 5 anos ela é tratada como branch de segurança, não sendo de manutenção, as alterações são realizadas apenas em casos especiais.

Por fim possuem benchmarks que está relacionado com o desempenho da aplicação diante das diferentes implementações, avaliando se o desempenho está de acordo com o esperado, é chamado de python benchmark suíte.

A documentação também é aberta a mudanças e revisões, caso detecte algum problema seja de digitação, coerência ou partes do código sem fazer parte do documento. O Python fornece um ícone para visualizar se já criaram uma pull request para o problema, campo para comentário sendo necessário informar o tempo para realizar essa tarefa e por fim enviar uma pull request.

A contribuição com a documentação é baseada da mesma maneira que o código, realizando um fork do repositório para sua conta e clonando. Sendo lançadas novas versões para ambos apenas em situações quando há incompatibilidade ou planejadas com antecedência, para correção dos bugs são micro versões, lançadas em média a cada 2 meses. Cada versão é marcada no repositor com uma tag do formulário vX.Y.ZTN, onde X corresponde a versão principal, Y a secundária e Z a micro versão.

**REFERÊNCIAS*.***

Python Developer´s Guide<<https://devguide.python.org/>> Acesso em 17.Set.2020.