

PUC-Rio
Data: 04/04/2019
Aluno: Wellington Bezerra Chaves
Matrícula: 1413383

Teste 6

1. Explique os termos "pull-request" e "commit" e o papel de cada um no controle de mudanças (GitHub).

Git:

Git é um sistema de controle de versão de arquivos. Através dele podemos desenvolver projetos na qual diversas pessoas podem contribuir simultaneamente no mesmo, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas.

Pull Request:

É basicamente uma forma de você dizer: acabei o meu trabalho, preciso integrá-lo no ramo principal submetendo todas as alterações de código feitas. Ao utilizar essa funcionalidade, é gerado um registro no servidor (por exemplo, o *GitHub*, que é bem popular) que destaca as diferenças de código entre o ramo no qual você trabalhou e o ramo alvo da entrega. Assim fica fácil para o administrador do repositório revisar o seu trabalho.

Também dá para iniciar discussões sobre trechos específicos das alterações. Por fim, o próprio git facilita a mesclagem do código. Isso incentiva a revisão em pares e o trabalho ordenado em equipe, e no caso de projetos open source incentiva também a contribuição de por parte da comunidade.

Commit:

Conforme o programador desenvolve um projeto, é recomendável que ele faça um commit podendo ser local(na própria máquina) e também no próprio servidor paara salvar todas as criações/alterações de código feitas por ele. Para um commit ser feito, é necessário que os arquivos alterados sejam incluídos e também uma descrição para facilitar a identificação da alteração. Caso um commit seja feito de forma errada, ele poderá ser desfeito.

2. Qual a importância da distinção entre verificação e validação para o construtor de software?

O processo de Validação e Verificação, não são processos separados e independentes.

Na verificação, o objetivo é analisar o software para certificar de que ele atende aos requisitos funcionais e não funcionais especificados. Nessa fase, realiza-se teste para encontrar erros. A pergunta principal que deve ser feita é a seguinte: Estamos construindo o produto corretamente? O humano pode ser um mecanismo da verificação.

A validação procura assegurar que o sistema atenda as expectativas e necessidades do cliente. A pergunta principal que deve ser feita é a seguinte: Estamos construindo o produto correto?

Durante o processo, podem ser utilizadas outras técnicas em conjunto, como a Inspeção. A Inspeção é o ato de analisar as representações do sistema, analisar os documentos de requisitos, diagramas e código-fonte. É recomendável a utilização da Inspeção em todas as etapas do processo.

Coerência X Consistência: A coerência diz respeito ao reflexo da realidade. Necessita de validação. A consistência diz respeito ao relacionamento de integridade das informações. Pode ser verificada

3. Na aula 9 vimos três pedaços de código (22, 23, e 24), na aula 5 um pedaço (8) e um na aula 7 (6). Dentre esses cinco pedaços de código, escolha 3 e descubra pelo menos uma pergunta de inspeção das elencadas por Sommerville no Capítulo 24 ("slides" 38 e 39 - ver elo no pdf da aula) que se aplicaria nos códigos escolhidos. Justifique cada uma.

Para essa questão, serão escolhidos os códigos dos slides 22, 23 e 24 da aula 9.

Código 1)

```
-- Recebe uma string e passa uma cópia sua onde
-- todos os caracteres não-alfanuméricos foram substituídos por espaços para a função recebida
-- PRE: str_data é uma string não nula (Verificação: existe uma assertiva garantindo isto)
-- POS: foi criada uma cópia de str_data, onde os caracteres não-alfanuméricos
-- foram substituídos por espaços, e as letras maiúsculas foram substituídas por letras minúsculas, e esta
-- cópia foi passada para a função recebida caso a função recebida não seja nula (Verificação: gsub("%W",'')
-- transforma os caracteres não-numéricos
-- em espaços em branco, e lower() transforma os caracteres maiúsculos em minúsculos)

function filter_chars_and_normalize(str_data, func)
    assert(str_data ~= nil, "A string que deveria ser filtrada esta nula no comeco da funcao")
    filter_chars_and_normalize(str_data, func)

    if(func ~= nil) then
        func(str_data:gsup("%W','");lower(), remove_stop_words)
    end
end
```

<https://github.com/nitanilla/PUC-Rio-INF1629-PES2006-2/blob/master/kick-forward/tf-kick-forward.lua>

Pergunta de inspeção:

- Todas as variáveis de entrada são utilizadas?
- todas as condições de erro possíveis foram levadas em conta?

Podemos ver que a função `filter_charts_and_normalize` recebe dois parâmetros e todos eles são utilizados. Além disso, antes de iniciar a função, uma verificação é feita na variável "data" para evitar que possíveis erros possam ocorrer.

Código 2)

```
-- Recebe uma string e procura por palavras,
-- e passa um vetor de palavras (usando espaço em branco como separador) para a função recebida
-- PRE: str_data é uma string não nula (Verificação: existe uma assertiva garantindo isto)
-- POS: foi criado um vetor com as palavras da string str_data, e este foi passado para a função recebida
-- caso esta não seja nula (Verificação: o for presente na função contém a função table.insert, que realiza
-- isso)
function scan(str_data, func)
    assert(str_data ~= nil, "A string que deveria ser usada esta nula no comeco da funcao scan")
    local iterator = str_data:gmatch("%S+")
    local word_list = {}

    for element in iterator do
        table.insert(word_list, element)
    end
    if(func ~= nil) then
        func(word_list, frequencies)
    end
end
```

Puc-Rio Dep. de Informática
PES-19-1 --
Usando, faça a citação -- 04/19

Pergunta de inspeção:

- Cada loop está certo de terminar?
- Todas as variáveis de programa são inicializadas antes de seus valores serem usados?

Como podemos ver no código acima, antes das variáveis serem utilizadas, elas são inicializadas. Além disso, o loop está certo de terminar, já que seu valor final encontra-se dentro da variável `iterator`.

Código 3)

```
(define (crie-uma-conta saldo)
  (define (retirar quantia)
    (if (>= saldo quantia)
        (begin (set! saldo (- saldo quantia))
                saldo)
        "Uh, estais sem dinheiro ..."))
  (define (depositar quantia)
    (set! saldo (+ saldo quantia))
    saldo)
  (define (processe pedido)
    (cond ((eq? pedido 'retirar) retirar)
          ((eq? pedido 'depositar) depositar)
          (else (error "pedido desconhecido - CRIE-UMA-CONTA"
                        pedido))))
  processe)
```

O código acima foi copiado do livro "[Structure and Interpretation of Computer Programs](#)", mas com mudanças dos nomes para Português.

Pergunta de inspeção:

- **Todas as chamadas de função e método têm o número correto de parâmetros?**
- **Todas as variáveis de entrada são utilizadas?**

A função "retirar quantia" tem como responsabilidade debitar um determinado valor da conta de uma pessoa e para que essa operação seja feita, ela recebe o número de parâmetros da forma correta e todos são utilizados. Essas afirmações podem ser aplicadas também para as funções "depositar quantia" e "processe pedido"