

---

# **Revisão de Software através de Inspeção em Código Fonte**

---

Demonstrar o uso de revisões de software através de inspeção de código fonte.

André Terceiro, Rodrigo Raminelli e Thiago Pelizoni

---

# Introdução

---

- Revisões são como um filtro
  - Eficiência da Revisão Estática
    - Fagan 60%;
    - Millis 90% (processo mais formal).
  - Revisar código fonte
    - Padronização de codificação para projetos;
    - Localizar erros que os testes automatizados não detectam;
    - Compartilhar o conhecimento.
-

# Técnicas

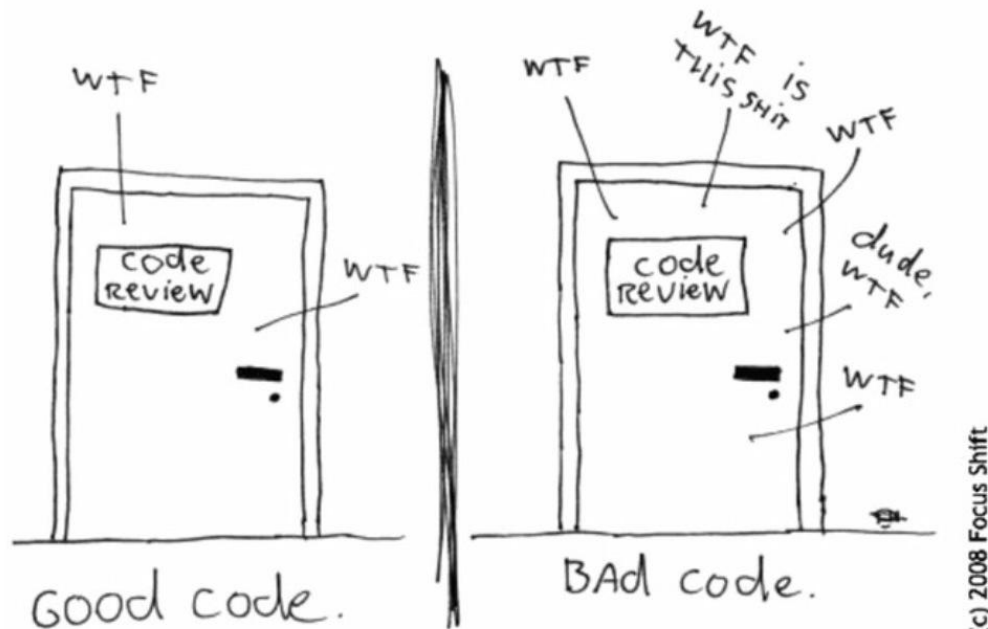
---

- Inspeção: Avaliação formal em que o código é examinado detalhadamente.
  - Inspeção técnica: O autor do código conduz um ou mais revisores pelo código.
  - Leitura de código: A leitura de código é recomendada para verificar pequenas modificações e como uma forma de "verificação de sanidade".
-

# Por que inspecionar código fonte?

---

The ONLY VALID MEASUREMENT  
OF CODE QUALITY: WTFs/MINUTE



# Porque inspecionar código fonte? (continuação)

---

**// Isso é um if ternário direto do inferno completamente ilegível**

```
($this->sql->numRows>0&&(($_POST['form_action']=='editar'&&$_POST['id_usuario']!= $rs_temp['ID_USUARIO'])||$_POST['form_action']=='cadastrar'))?$cpf_duplicado=true:$cpf_duplicado=false;
```

**// Código do mal que não faço a menor idéia para que serve**

```
if(!ereg($this->fileLimit['regex'],$this->fileInfo['mime'])&&!empty($this->fileLimit['regex'])) {  
    $item[0] = false;  
} else {  
    $item[0] = true;  
}  
if(!empty($this->fileLimit['w'])) {  
    if($this->fileLimit['tp']=='==') {  
        if($this->fileInfo['w']!= $this->fileLimit['w']) {  
            $item[1] = false;  
        } else {  
            $item[1] = true;  
        }  
    } else if($this->fileLimit['tp']=='>=') {  
        if($this->fileInfo['w']< $this->fileLimit['w']) {  
            $item[1] = false;  
        }  
    }  
}
```

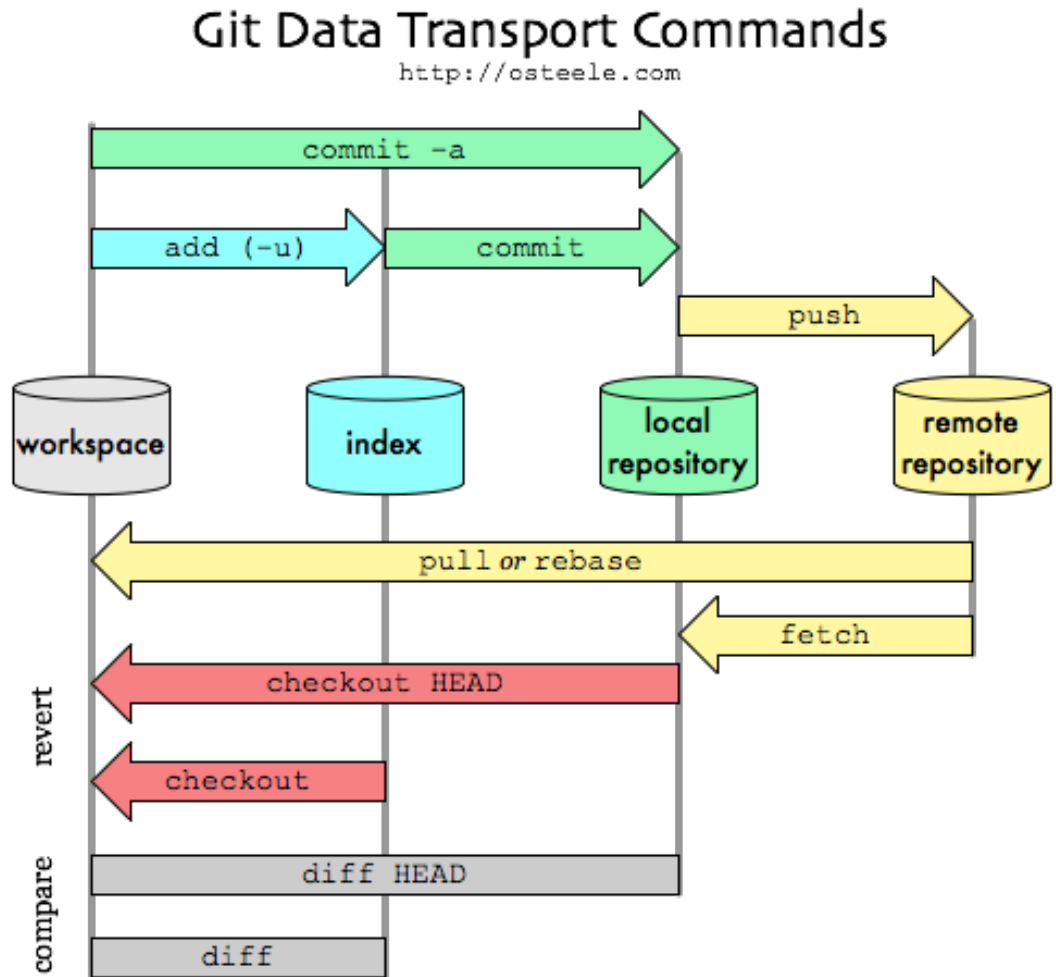
# Controle de versões

---

- Git
    - Cópia de trabalho
    - Repositório
    - Branches
    - Commit
    - Push
    - Pull Request
    - Merge
-

# Git

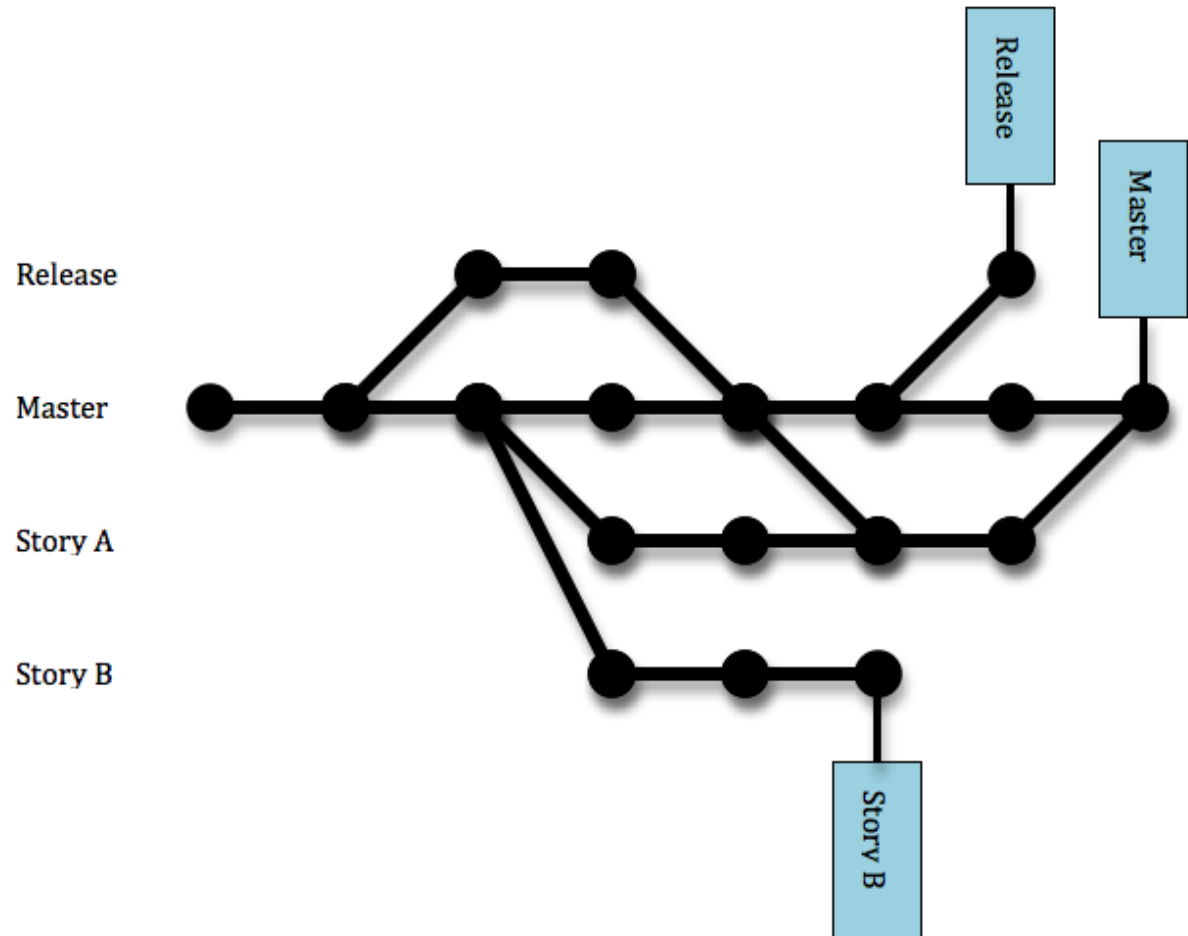
- Cópia de trabalho
- Repositórios
- Commit
- Push
- Pull



# Git

---

- Branches

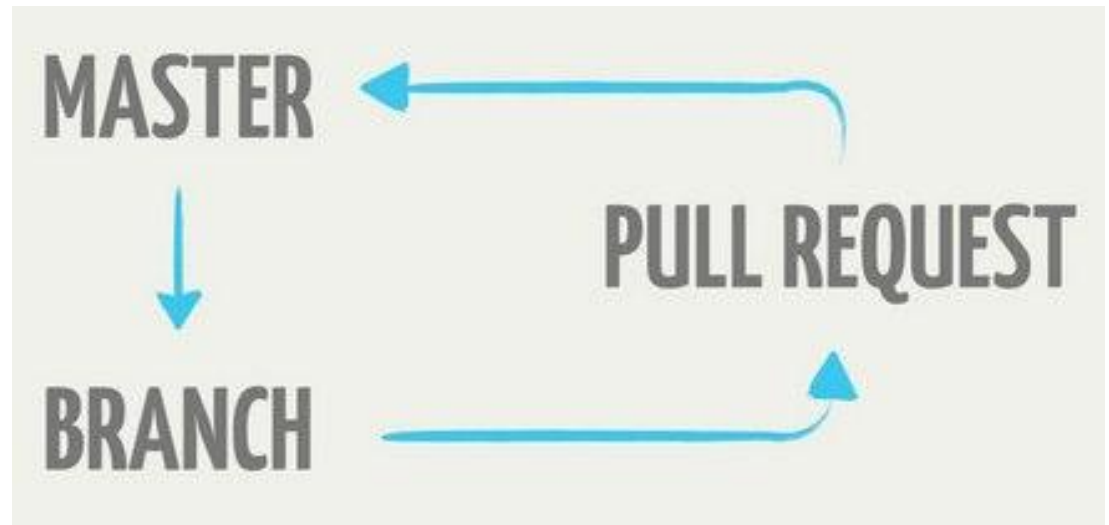




# Git

---

- Pull request
- Merge



# Materiais e métodos

---

A técnica de inspeção que este artigo aplicou foi a de Fagan, que propõe um processo estruturado de tentar encontrar defeitos em documentos de desenvolvimento, tais como o código de programação.

- Planejamento
  - Reunião de visão geral
  - Preparação
  - Reunião
  - Retrabalho
  - Acompanhamento
  - Análise causal
-

# Planejamento

---

- O processo de revisão através de inspeção de código fonte foi institucionalizado dentro da equipe de desenvolvimento;
  - A cada commit todos os membros da equipe recebem um e-mail e podem contribuir com a inspeção.
-

# Reunião de visão geral

---

Não há uma reunião de visão geral no processo da empresa, pois os participantes, neste caso os desenvolvedores, já estão familiarizados com a atividade de revisão devido a esta já ter sido institucionalizada.

---

# Preparação (Checklist)

---

- Erros de sintaxes;
  - Fora de padrão de codificação: não seguir o padrão de normas de codificação estipulados pela PHP Framework Interoperability Group, a PSR
  - Ajuste de lógica: Um método ou implementação, apesar de estar funcionando, muitas vezes contém uma alta complexidade ciclomática
  - Erro de arquitetura: Implementar regras de negócio em controladores ou alta complexidade na camada de visualização.
  - Sugestão de melhoria: Quando uma implementação está concisa, porém pode ser melhorada de algum modo.
-

# Reunião (assíncrona)

---

Card #357

---

**You can merge this Pull Request by running:**

```
git pull https://github.com/casperlibero/sao-silvestre-2012 lp-feature-relatorio-participante-pf
```

Or view, comment on, or merge it at:

<https://github.com/casperlibero/sao-silvestre-2012/pull/85>

## Commit Summary

- Refatoração relatorio de motivos
- Merge branch 'develop-local' into lp-feature-relatorio-motivo
- Refatoração da query de busca do relatorio de motivo e montagem da pa...
- Merge branch 'develop' of [https://github.com/casperlibero/sao-silvest...](https://github.com/casperlibero/sao-silvestre-2012)
- Refatoração da query de busca para o relatorio de participante pf e m...

## File Changes

- **M** application/models/RelatorioMotivoA4Paisagem.php (55)
- **M** application/models/RelatorioMotivoMapper.php (44)
- **M** application/models/RelatorioParticipantePfA4Paisagem.php (122)
- **M** application/models/RelatorioParticipantePfMapper.php (47)

## Patch Links

- <https://github.com/casperlibero/sao-silvestre-2012/pull/85.patch>
- <https://github.com/casperlibero/sao-silvestre-2012/pull/85.diff>

---

Reply to this email directly or [view it on GitHub](#).

# Retrabalho

---




Após o recebimento do comentário da inspeção por e-mail, o responsável pela implementação efetua as devidas alterações/correções e efetua o commit novamente para que o código seja analisado. Caso ainda sejam encontradas irregularidades, o merge deste branch não é efetuado até que todas que foram notificadas sejam corrigidas.

---




# Acompanhamento

```
... @@ -0,0 +1,21 @@
1 +<?php
2 +
3 +namespace Application\Model;
4 +
5 +class Departamento
6 +{
7 +    //-----
8 +
9 +    public $id;
10 +    public $nome;
11 +
12 +    //-----
13 +
14 +    public function exchangeArray($data)
15 +    {
16 +        $this->id = (isset($data['id'])) ? $data['id'] : null;
```

4




 **castilhor** 20 days ago  

Aqui talvez seja melhor utilizar empty (se estiver definida, mas vazia, associa null também).




 **thiagopelizoni** 20 days ago  

Se a classe, por exemplo, tiver 10 atributos, haverá um if ternário para cada um? Não faz sentido, isto pode ser automatizado desta maneira:

<http://pastebin.com/ZP7pzRWz>

 **thiagopelizoni** 20 days ago  

Outro ponto, se todas as classes terão o mesmo comportamento, porque não se faz uma classe abstrata contendo o método deste modo e as outras herdam este comportamento?

 **garbini** 20 days ago  

Boa dica @thiagopelizoni. Irei utiliza-la.



# Análise Causal

---

- Este processo não tem como objetivo expor o desenvolvedor que porventura tenha cometido algum erro de codificação e sim auxiliá-lo através da experiência dos demais como o seu código poderia ficar melhor.
  - Em contrapartida, um desenvolvedor menos experiente inspecionando o código de um desenvolvedor mais experiente, tende a adquirir mais conhecimento e aplicá-lo quando estiver implementando alguma funcionalidade.
-

# Resultados

---

- 2 meses;
  - Entre 3 a 5 commits de cada desenvolvedor por semana;
  - Projeto de Inscrição dos participantes da Corrida de São Silvestre;
  - Total de linhas de código do projeto em 19/12/2012: 123.003 linhas
    - `wc -l $(find . -type f | egrep -v jquery\\|img\\|jpg\\|png\\|gif\\|.git\\|library/Zend\\|library/Twitter)`
  - A técnica não foi aplicada desde o início do projeto, mas, quando este já tinha em torno de 58.000 linhas de código.
-

# O que pode ser observado?

---

- Através da inspeção de código fonte foi possível identificar em média entre 3 e 5 irregularidades de desenvolvedores sêniores e entre 8 e 12 irregularidades de desenvolvedores plenos a cada commit.
  - O fator técnico do desenvolvedor influi intrinsecamente na inspeção do código. Códigos submetidos a revisão por desenvolvedores de nível sênior possuíam um número muito menor de irregularidades em relação a códigos submetidos por desenvolvedores de nível pleno;
-

# O que pode ser observado?

---

- Um commit sendo analisado por desenvolvedores de diferentes níveis mostraram que um desenvolvedor senior encontra mais erros relacionados a arquitetura de sistema, padrões de projeto e boas práticas do que um desenvolvedor pleno;
-

# O que pode ser observado?

---

- Quanto a um commit sobre uma anotação, o desenvolvedor responsável pela implementação e pela correção das irregularidades adquire a experiência com seu erro. Foi observado que em commits posteriores estes erros não foram recorrentes;
-

# O que pode ser observado?

---

- Desenvolvedores plenos tendem a cometer muito mais erros de arquitetura do que desenvolvedores sêniores. Com isso, o custo do retrabalho de, por exemplo, refatorar a regra de negócio de um controlador para um modelo é muito maior, fazendo com que o tempo para o merge deste commit seja maior;
-

# O que pode ser observado?

---

- O ego do desenvolvedor pode ser prejudicial quando seu código está sendo inspecionado. Algumas vezes, por problemas levados a questões pessoais, a anotação para correção pode não ser vista como algo construtivo;
-

# O que pode ser observado?

---

- Quanto mais imaturo profissionalmente o desenvolvedor, menos suscetível a correções este é.



# Conclusão

---

- Aumento de conhecimento dos integrantes da equipe;
  - Estimulo a debates e padronizações;
  - Facilidade em manutenção de código de projetos aos quais você não fez parte;
  - Facilidade de substituição de desenvolvedor.
-