

Code Review



TODOS OS DIREITOS RESERVADOS

● Maio
2019



O que é Code Review?

TOTVS



// O que é Code Review?

3



Code Review



Uma **revisão de código** que visa contribuir com a qualidade e os padrões de projeto da empresa, realizada por alguém que não seja o próprio autor.

Code review is **systematic examination** (sometimes referred to as peer review) of computer source code. It is intended to find mistakes **overlooked in the initial development phase**, improving the overall quality of software. Reviews are done in various forms such as pair programming, informal walkthroughs, and formal inspections

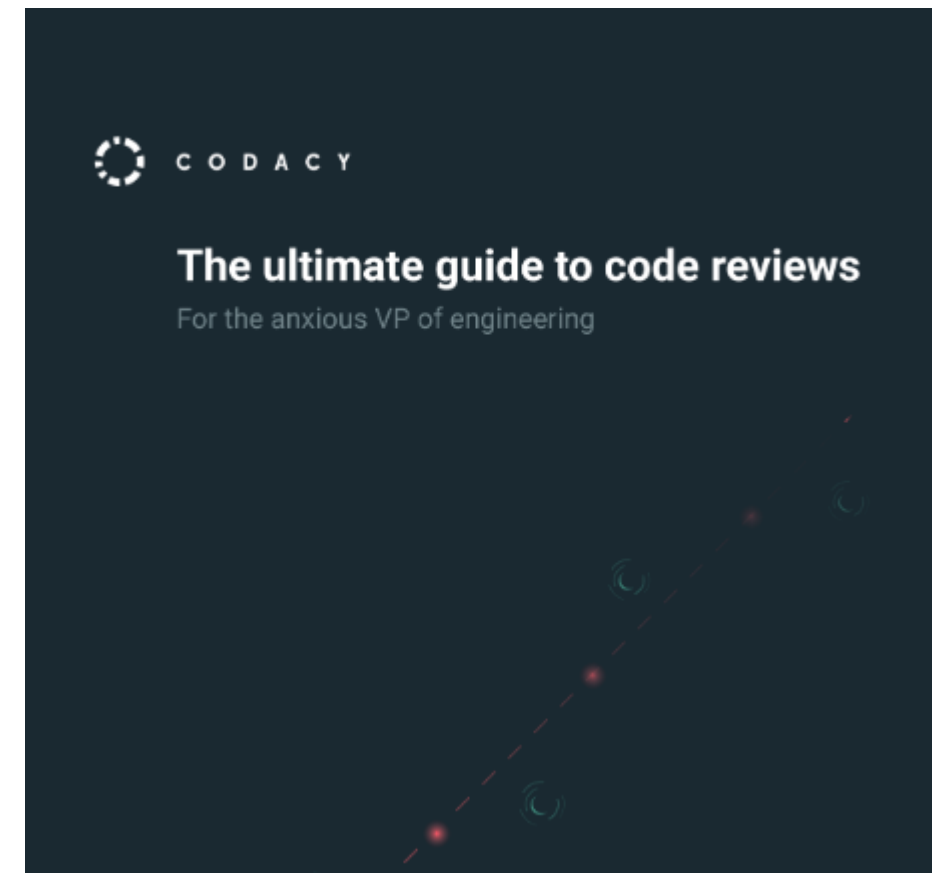
WIKIPEDIA

// O que é Code Review? - Referências

4



<https://leanpub.com/whattolookforinacodereview>



<https://www.codacy.com/ebooks/guide-to-code-reviews>



O que analisar na revisão?



- **Design**
 - Como o código se encaixa na **arquitetura**? Está no lugar certo?
 - O código segue os princípios do **SOLID**, **DDD** ou outros paradigmas utilizados pela empresa?
 - Quais os **Design Patterns** utilizados? Eles são apropriados?
- **Reusabilidade**
 - O autor poderia ter utilizado algo já existente?
 - A alteração induz a duplicação de código? Deveria ser refatorado?
- **Legibilidade e Manutenibilidade**
 - A nomeação de classes, parâmetros, campos e métodos refletem o que realmente representam?
 - Eu consigo entender o que o código faz lendo ele?
 - As mensagens de exceção são claras?
- **Funcionalidade e Performance**
 - Há pontos de potenciais falhas? NullPointerException, Undefined
 - Há possibilidade de melhoria na performance? Acesso a banco, abertura de conexões.
 - As estruturas de dados utilizadas são as melhores?

// Tipos de Code Review – Quando?



| | Pros | Contras |
|--------------------|--|---|
| Pre-Commit Review | <ul style="list-style-type: none">✓ Evita que o código não revisado se junte a branch principal.✓ Evita que a revisão possa ser adiada ou omitida. | <ul style="list-style-type: none">✓ Diminui a produtividade✓ Depois de ter seu código revisado o Dev pode fazer um commit equivocado. |
| Post-Commit Review | <ul style="list-style-type: none">✓ Fluxo de trabalho mais fluído, pois o código não precisa de revisão pra ser commitado.✓ As alterações são disponibilizadas para o time rapidamente. | <ul style="list-style-type: none">✓ Aumenta as chances de um código ruim se juntar a branch afetando o resto do time.✓ Se algum defeito for encontrado, levará um tempo extra para correção do mesmo. |



Tipos de Code Review – Como?

7



| | Pros | Contras |
|------------------------|---|---|
| Code Review em Grupo | <ul style="list-style-type: none">✓ Aumenta a chance de descobrir problemas em potencial.✓ Maior pulverização das informações e mais insights✓ Exclui a figura do revisor individual. | <ul style="list-style-type: none">✓ Diminui a produtividade |
| Code Review Individual | <ul style="list-style-type: none">✓ Mais rápido e mais produtivo. | <ul style="list-style-type: none">✓ Reforço da figura do revisador.✓ Informações são menos compartilhadas. |

Com que frequência fazer Code Review?



- ✓ Em partes críticas do sistema.
- ✓ Rotinas que envolvem acesso a banco.
- ✓ Issues de alta complexidade.
- ✓ Somente em novas features ou novos produtos.

**Realizar as definições
durante a Sprint Planning**

**+ Frequência = + Qualidade
de Software**



Por que fazer Code Review?



- Compartilhamento de informação(regra de negócio e técnicas de programação)
 - Geração de ideias
 - Aprendizado com os pares
- Melhoria na Qualidade de Código(Clean Code)
 - Legibilidade
 - Performance / Eficiência e Eficácia
 - Padronização
 - Reusabilidade
- Reduz o número de manutenções
- Mindset da equipe mais aguçado(The Ego Effect)



02

Como fazer Code Review?

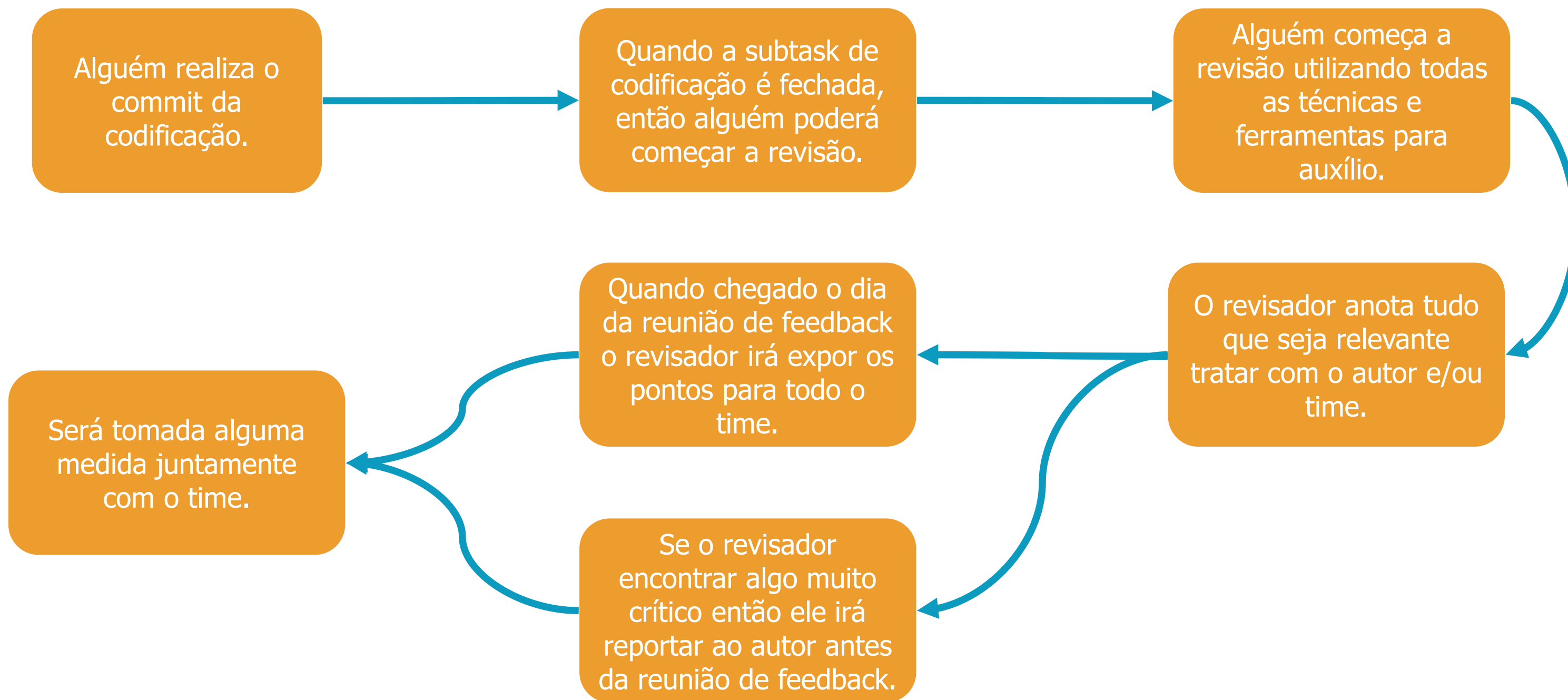
TOTVS





Workflow da Code Review no DevTeam ATIREC

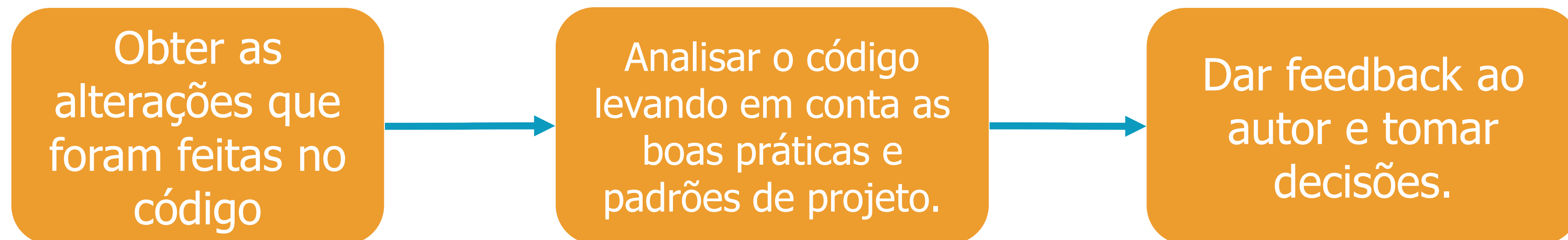
11





Basicamente nós precisamos...

12





03

Obtendo as alterações

TOTVS





- Forma Automatizada
- Forma Manual



Control panel > CD-JV > PIMS-Cana > PIMS-Cana Team

Overview Security Alerts Version Control Service Hooks Services Test

My Alerts (1)

- Work Item Alerts
- Checkin Alerts (1)
- Build Alerts
- Code review Alerts

PIMS-Cana Team alerts

CREATE NEW ALERT WHEN

- A build fails
- My build completes
- A file is checked in under a specified path
- A code review I am working on changes
- A change is made to a work item that I created
- A work item is assigned to me
- Other...

My Checkin alerts

Showing 1 of 1 alerts for Fernando Da Silva Cares_

New... | [Icons]

| Name | Send To |
|---|-------------------------|
| Any checkin occurs in this team project | [Default alert address] |

Name: Any checkin occurs in this team project

Send to: fernando.cares@totvs.com.br [default]

Alert filters



03

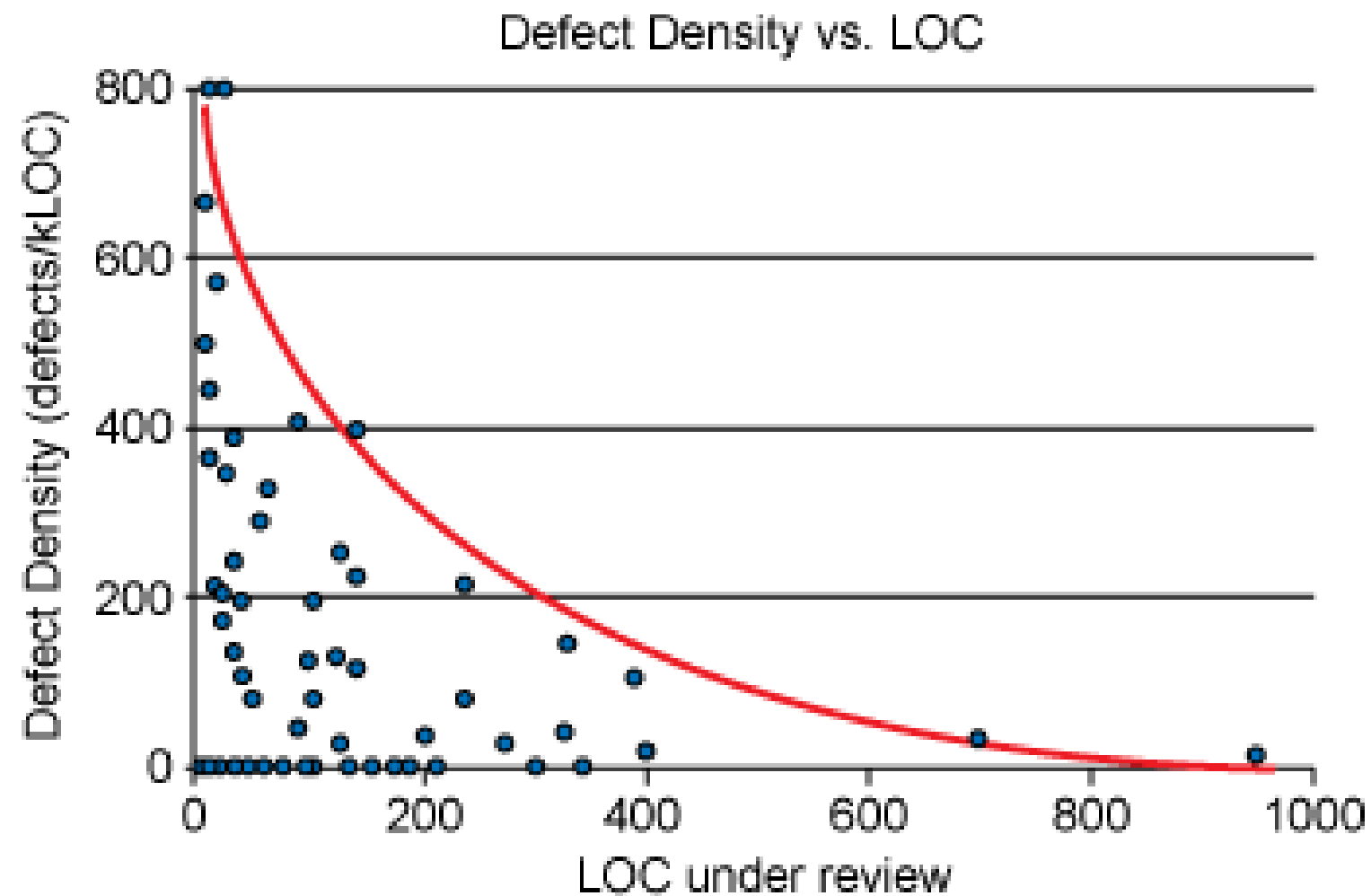
Analizando o Código

TOTVS



// Boas práticas da Code Review

- Revisar no máximo 400 linhas por Code Review.
- Fazer uma pausa quando a revisão passar de 1 hora.
- Usar um checklist para elencar o que será analisado(Objetivos)
- Utilizar a mentalidade crítica no momento da análise.
- Utilizar ferramentas para análise de código.



SmartBear study of a Cisco Systems programming team.





About PMD

PMD is a source code analyzer. It finds common programming flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth. It supports Java, JavaScript, Salesforce.com Apex and Visualforce, PLSQL, Apache Velocity, XML, XSL.

Additionally it includes CPD, the copy-paste-detector. CPD finds duplicated code in Java, C, C++, C#, Groovy, PHP, Ruby, Fortran, JavaScript, PLSQL, Apache Velocity, Scala, Objective C, Matlab, Python, Go, Swift and Salesforce.com Apex and Visualforce.

<https://pmd.github.io/>

SpotBugs

SpotBugs is a program which uses static analysis to look for bugs in Java code. It is free software, distributed under the terms of the [GNU Lesser General Public License](#).

SpotBugs is the spiritual successor of [FindBugs](#), carrying on from the point where it left off with support of its community. Please check [official manual site](#) for details.

SpotBugs requires JRE (or JDK) 1.8.0 or later to run. However, it can analyze programs compiled for any version of Java, from 1.0 to 1.9.

<https://spotbugs.github.io/>

sonarlint

Get the power
to write better code

```
var updateGraph = function ( dom, height, color ) {  
  
    var child = dom.appendChild( dom.firstChild );  
    child.style.height = height + 'px';  
    if( color ) child.style.backgroundColor = color;  
  
}  
  
var perf = window.performance || {};  
// polyfill usedJSHeapSize  
if ( ! perf && !perf.memory){  
    perf.m  
}  
// suppo (javascript:S2583)  
if( perf  
    console.warn( totalJSHeapSize --- 0.1% performance.memory is only 8  
}
```

sonarlint

Change this condition so that it does not always evaluate to "false"; some subsequent code is never executed.

Open description of rule javascript:S2583

Bug detection

Benefit from thousands of rules [🔗] ; which detect common mistakes, tricky bugs and known vulnerabilities.

Instant feedback

On-the-fly! Issues are detected and reported as you code, just like a spell-checker.

Know what to do

SonarLint precisely pinpoints where the problem is, and gives you recommendations on how to fix it.

<https://www.sonarlint.org/features/>



codelyzer

A set of tslint rules for static code analysis of Angular TypeScript projects.

You can run the static code analyzer over web apps, NativeScript, Ionic, etc.

<https://www.npmjs.com/package/codelyzer>



04

Fazendo o feedback.



Como fazer o feedback com o autor?

- Anotar todos os insights e pontos de atenção antes de realizar o feedback, para não ter double spend de tempo.
- Ao sinalizar um ponto de melhoria, de alternativas e utilize argumentos. Se preciso busque fundamentação.
- Torne o feedback algo construtivo e não desmoralizador.

"Quality is the Best Business Plan"

John Lasseter





<https://searchsoftwarequality.techtarget.com/definition/code-review>

<https://medium.com/trainingcenter/qual-o-real-valor-do-code-review-para-uma-equipe-de-desenvolvimento-f43f894c0a04>

<https://github.com/viniciusalonso/code-review-good-practices>

<https://medium.com/@tadasant/theres-a-human-on-the-other-side-of-your-code-review-9732cc15bfee>

<https://medium.com/equals-lab/a-importancia-do-code-review-para-a-equipe-de-desenvolvimento-de-software-a47b70cb1560>

<http://www.methodsandtools.com/archive/whycodereviews.php>

<https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/>

<https://smartbear.com/resources/case-studies/cisco-systems-collaborator/>

<http://www.anarsolutions.com/code-review-vs-code-walkthrough/>

<https://smartbear.com/learn/code-review/agile-code-review-process/>

OBRIGADO



Fernando Cares

Software Developer

18 99759 4542

fernando.cares@totvs.com.br

LinkedIn: Fernando Cares

 totvs.com

 company/totvs

 @totvs

 fluig.com

Tecnologia + Conhecimento são nosso DNA.

O sucesso do cliente é o nosso sucesso.

Valorizamos gente boa que é boa gente.

#SOMOSTOTVERS