

Sensedia Dashboard

José Vahl - jose.vahl@sensedia.com

Rafael Navarro - jose.vahl@sensedia.com

Marcilio Oliveira - jose.vahl@sensedia.com

Sobre a Sensedia

- Especializada em soluções para **Governança SOA** (Arquitetura Orientada a Serviços) e **Reutilização** de Ativos de Software
- **Visionária** no Quadrante Mágico para Governança SOA do **Gartner** (março/2009)



Produtos e Serviços

Sensedia SOA Solution



Assessment e Gap Analysis, Processos de Governança SOA, Infra Estrutura, Roadmap de Adoção e Capacitação

sensedia discoverer



- Identificação automática de ativos existentes
- Gráficos de relacionamentos

sensedia repository



- Gestão dos Serviços e componentes
- Governança do ciclo de vida dos ativos
- Visibilidade do acervo e promoção do reúso
- Métricas de ROI
- Análise de Impacto

sensedia dashboard



- Análise da Qualidade Ativos
- Avaliação da Saúde dos Projetos

Objetivos

🎯 O *Sensedia Dashboard* dá a **visibilidade** que os gerentes precisam para **medir, controlar e melhorar** a qualidade técnica e a performance dos times de desenvolvimento, sejam eles internos, externos ou ambos.



- ✓ Aumenta a visibilidade gerencial
- ✓ Garante a adequação aos padrões de qualidade
- ✓ Melhora a qualidade dos ativos

Qualidade de software

1.

```
void  
usage (char *name)  
{  
    printf ("usage:\n");  
    printf ("%s -a [-c file",  
    name);  
    #ifdef LOFF  
    printf ("[-g] [-G] ");  
    #endif  
    printf ("[-p what] [-r]  
    [-u file [type]]");  
    #ifdef LOFF  
    printf (" [-w love] [-w  
    ewid] [-z size] ");  
    #endif  
}
```



✓ Qualidade do código

2.



✓ Conformidade com requisitos

3.



✓ Acompanhamento do processo

4.



✓ Satisfação do cliente

Motivação



Não há nenhuma definição amplamente reconhecida sobre qualidade de métricas

Mesmo quando sabemos que números usar, os dados não são fáceis de serem obtidos

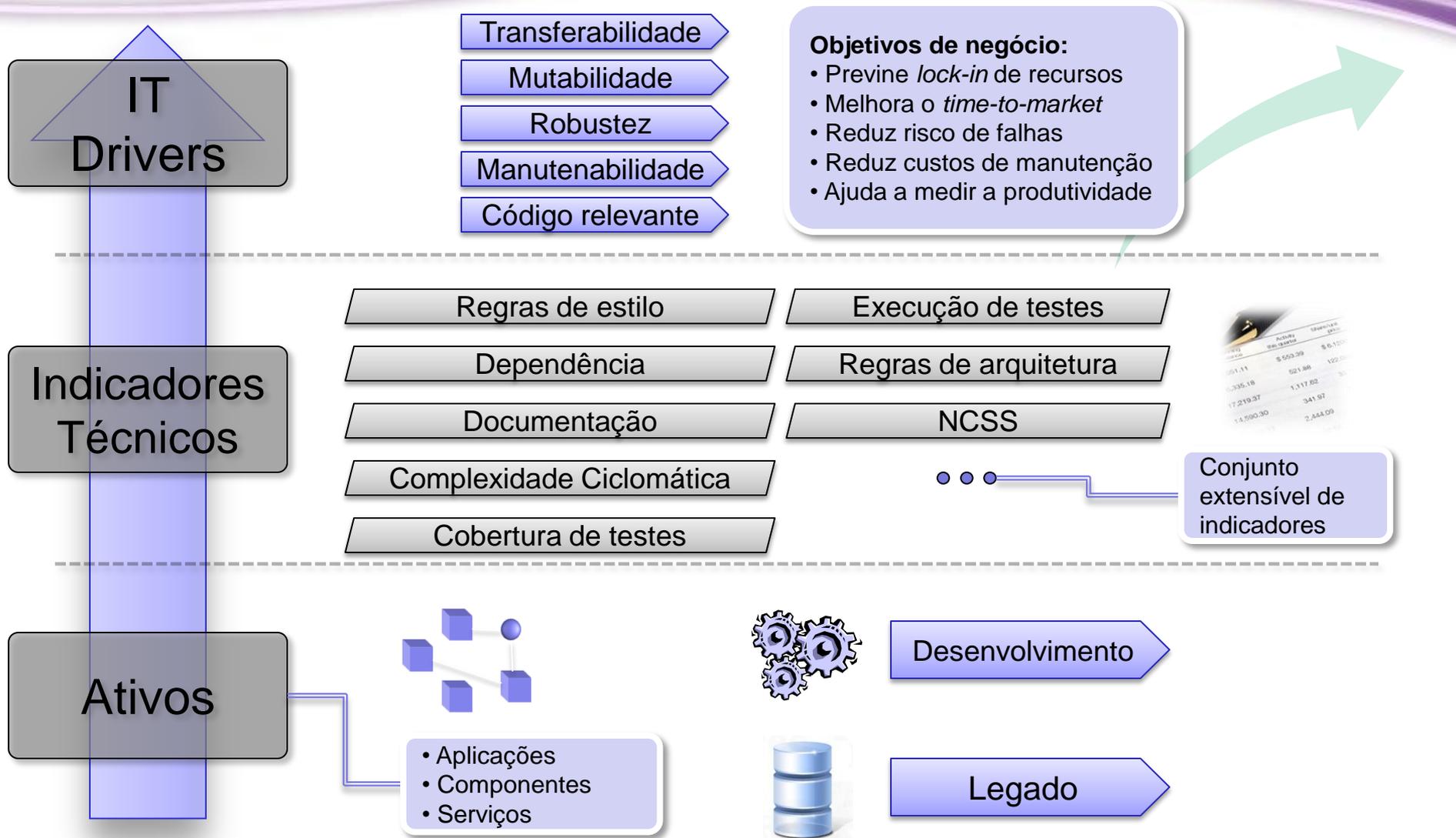
Mesmo com os dados, não é tão óbvio saber interpretar e usar os números

As pessoas são relutantes em medir a qualidade do seu próprio trabalho

- Principais Funcionalidades:
 - ✓ Análise de indicadores de qualidade arquitetural e saúde dos projetos e componentes
 - ✓ Integração com o Sensedia Repository e o ambiente de build da empresa
 - ✓ Integração com ambientes de desenvolvimento
 - ✓ Painel indicativo utilizando semáforos



Saúde de ativos



Fatores de saúde



- Fatores de saúde mapeados pelo Sensedia Dashboard:
 - ✓ Transferabilidade
 - Determina o quão facilmente um time pode tornar-se produtivo após receber um projeto já iniciado por outro time
 - ✓ Mutabilidade
 - Determina o quão rápido e facilmente uma aplicação pode ser modificada
 - ✓ Robustez
 - Determina o risco de falhas ou defeitos que podem acontecer após a modificação de uma aplicação
 - ✓ Manutenibilidade
 - Determina o custo e a facilidade/dificuldade para se manter uma aplicação
 - ✓ Código relevante
 - Determina a quantidade técnica de código baseada em linhas de código efetivas, número de métodos, número de campos, etc.

Metodologia

"Without numbers, quality programs are just talk."
 Watts S. Humphrey
 SEI



Source Assets

- * Applications
- * Components
- * Services

Under Development

Existing Software

Control Panel



- * Alerts
- * Baselines
- * Balance weight projects
- * Schedule Builds

Extensible Quality Metrics



- Cyclomatic Complexity
- Documentation
- Test Coverage
- Test Execution
- Components Dependency
- Naming Convention
- Code Signing
- Custom...

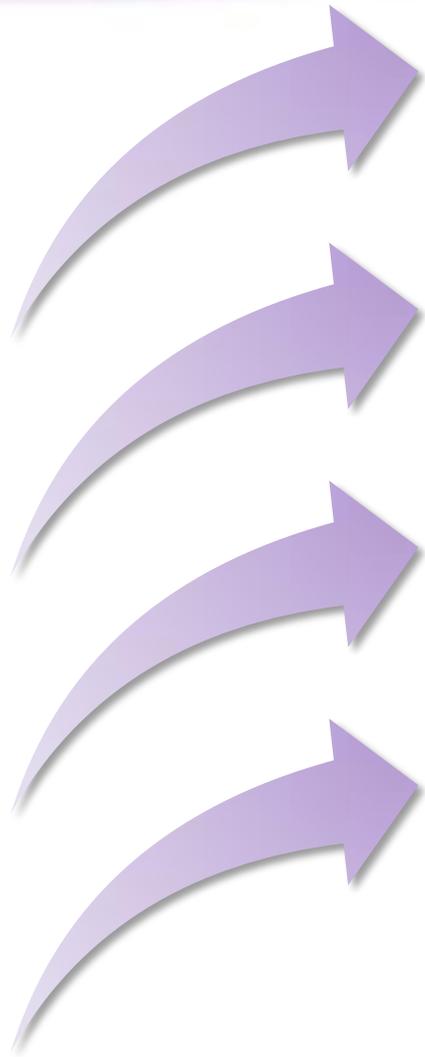
Health Report

Group	Metric	Status	Value	Good	Regular	Bad
LOC	LOC.Total		6.519	-	-	-
	LOC.NonComment		4.538	-	-	-
Code test	CodeTest.Coverage	●	0%	100%	-	0% - 99%
	CodeTest.Coverage	●	72%	90% - 100%	75% - 89%	0% - 74%
Duplicate code	DuplicateCode	●	17%	60% - 100%	43% - 59%	0% - 44%
	DuplicateCode	●	0	0	1 - 5	> 5
Code size	CodeSize	●	145	0 - 500	501 - 700	> 700
	CodeSize	●	246	0 - 1.400	1.401 - 2.300	> 2.300
CCN	CCN	●	1.923	0 - 45.000	45.001 - 50.000	> 50.000
	CCN	●	8%	0% - 5%	6% - 15%	> 15%
Documentation	Documentation	●	3%	0% - 5%	6% - 10%	> 10%
	Documentation	●	37%	80% - 100%	60% - 79%	0% - 59%
Dependencies	Dependencies	●	10.764	0 - 100	101 - 200	> 200
	Dependencies	●	537	0 - 100	101 - 200	> 200
Alerts	Alerts	●	0	0 - 1	-	> 1
	Alerts	●	70	0 - 1	-	> 1
Alerts	Alerts	●	0	0 - 10	11 - 14	> 14
	Alerts	●	61	0 - 10	10 - 65	> 65
Alerts	Alerts	●	0	0 - 10	-	> 10
	Alerts	●	0	0 - 10	-	> 10

Business Goals:

- Prevent resource lock-in
- Speed time-to-market
- Reduce failure risks
- Reduce maintenance costs
- Help to measure productivity

Dimensões da análise



Arquitetura

Garante que alterações podem ser feitas com baixo risco

Estática

Analisa o código focando em defeitos e não conformidades

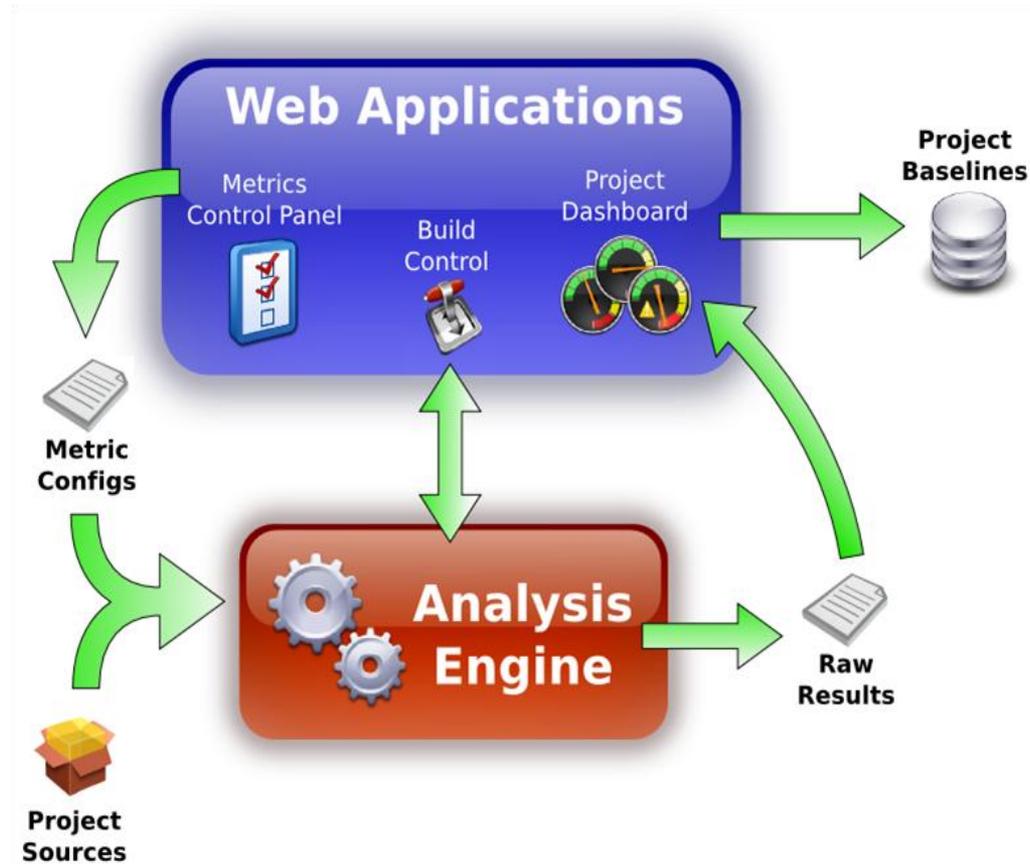
Build

Analisa a build de softwares para identificar problemas

Dinâmica

Executa e analisa os testes unitários da aplicação

Arquitetura



O produto





Dashboard consolidado de análise da qualidade dos componentes, serviços e projetos



Total Ativos: 65.22%

Ativos

Acegi Security - v. 1.0.7

Última análise: 10/06/09

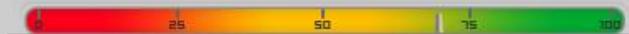


Executar

Configuração

Configuration - v. 1.0

Última análise: 10/06/09



Executar

Configuração

Dashboard Core - v. 1.0

Última análise: 10/06/09

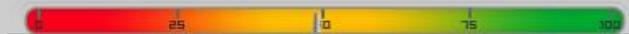


Executar

Configuração

SGC Pregao - v. 1

Última análise: 22/06/09

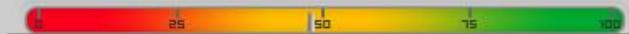


Executar

Configuração

V-Security - v. 1.0.230

Última análise: 15/04/09



Executar

Configuração

Administração



Bem vindo ao Dashboard

Logo
de sua
empresa

Administrar

Ativos habilitados para a análise com sua respectiva relevância

Administrar

Ativos

	Ativos	Peso	Excluir
<input checked="" type="checkbox"/>	Dashboard Core - v. 1.0	3	
<input checked="" type="checkbox"/>	Acegi Security - v. 1.0.7	1	
<input checked="" type="checkbox"/>	WS-S wrapper - v. 0.9	1	
<input checked="" type="checkbox"/>	SGC Pregao - v. 1	1	
<input checked="" type="checkbox"/>	V-Security - v. 1.0.230	2	
<input checked="" type="checkbox"/>	Configuration - v. 1.0	1	
<input type="checkbox"/>	SGC - v. 3.0	1	

Salvar

Cancelar

Novo Ativo...

« Voltar

Conjunto extensível de indicadores

Painel de Controle

Configurar Ativo: Dashboard Core - v. 1.0

Ativo

Tamanho de Código Complexidade Ciclomática Coesão e Acoplamento

Regras de Estilo Testes Unitários Otimização de Código

Design de Programação

Nome* Versão*

Dashboard Core 1.0

Descrição

Esse projeto tem o objetivo de criar uma solução de software para auxiliar as atividades de acompanhamento da saúde de ativos de software (projetos de desenvolvimento, componentes, serviços, etc.) baseado em diversas métricas de qualidade configuráveis e extensíveis.

Responsável* Raiz do Ativo*

José Vahl C:\projs\core-1.0

Análises salvas no histórico* Baselines salvas no histórico*

15 5

Peso Data de início das análises

3 03/04/09

Grupos de indicadores e sua relevância para o ativo

Indicadores para executar

Indicador de Qualidade	Peso
<input checked="" type="checkbox"/> Tamanho de Código	1
<input type="checkbox"/> Complexidade Ciclomática	1
<input checked="" type="checkbox"/> Coesão e Acoplamento	1
<input checked="" type="checkbox"/> Regras de Estilo	1
<input type="checkbox"/> Testes Unitários	1
<input checked="" type="checkbox"/> Otimização de Código	1
<input checked="" type="checkbox"/> Design de Programação	1

Salvar Cancelar Executar

Painel de Controle

Configurar Ativo: Dashboard Core - v. 1.0

Ativo

Tamanho de Código

Complexidade Cidomática

Coesão e Acoplamento

Testes Unitários

Otimização de Código

Sub-indicadores
configurados
individualmente

Otimização de Código

Análise de Variáveis Locais

Peso*

Analisa quando uma variável ao ser atribuída apenas uma vez, deveria ser declarada como "final".

- Menor que Violações
- Entre e Violações
- Maior que Violações

Análise de Argumentos de um Método

Peso*

Analisa um argumento de um método quando nunca é atribuído, deveria ser declarado como "final".

- Menor que Violações
- Entre e Violações
- Maior que Violações

Análise de criação de objeto dentro de um loop

Peso*

Analisa quando um novo objeto é criado dentro de um loop.

- Menor que Violações
- Entre e Violações
- Maior que Violações

Resultados



Bem vindo ao Dashboard

Painel de semáforos com os resultados da análise

Logo de sua empresa

Administrar

Configuração

Ativo: Dashboard Core - v. 1.0

Grupo	Indicador	Status	Valor	Bom	Regular	Ruim
Regras de Estilo	Alertas	●	0	< 25	25 - 50	> 50
	Erros	●	30	< 15	15 - 40	> 40
Tamanho de Código	Análise de NCSS por Método	●	13	< 10	10 - 20	> 20
	Análise de NCSS por Tipo	●	7	< 15	15 - 30	> 30
	Análise de NCSS por Construtor	●	0	< 8	8 - 15	> 15
	Quantidade excessiva de Campos	●	0	< 5	5 - 10	> 10
Coesão e Acoplamento	Violações	●	2	< 10	10 - 25	> 25
Design de Programação	Violações de Design	●	29	< 40	40 - 80	> 80
	Análise de Variáveis Locais	●	120	< 40	40 - 100	> 100
Otimização de Código	Análise de Argumentos de um Método	●	92	< 5	5 - 10	> 10
	Análise de criação de objeto dentro de um loop	●	10	< 5	5 - 10	> 10
	Análise de ArrayList e Vector	●	0	< 15	15 - 30	> 30
	Análise do uso de "CharAt"	●	0	< 20	20 - 40	> 40
	Análise do uso de concatenação de Strings	●	0	< 10	10 - 20	> 20
	Análise do uso de Array com List	●	0	< 10	10 - 20	> 20
	Análise de Array e Loops	●	0	< 100	100 - 200	> 200
	Análise do uso Parsing	●	0	< 10	10 - 20	> 20
	Análise de conversão	●	0	< 30	30 - 100	> 100

◀ Voltar

Resultado detalhado

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<pmd version="4.2.2" timestamp="2009-06-10T23:02:03.833" />
```

```
- <file name="C:\projs\core-1.0\src\br\com\da\dashboard\commons\config\MetricConfigSerializerFactory.java">
```

```
<violation beginline="24" endline="24" begincolumn="17" endcolumn="46" rule="LocalVariableCouldBeFinal" ruleset="Optimization
```

```
externalInfoUrl="http://pmd.sourceforge.net/rules/optimizations.html#LocalVariableCouldBeFinal" priority="3">Local variable  
entries could be declared final</violation>
```

```
<violation beginline="26" endline="26" begincolumn="23" endcolumn="60" rule="LocalVariableCouldBeFinal" ruleset="Optimization
```

```
Rules" package="br.com.da.dashboard.commons.config" class="MetricConfigSerializerFactory" variable="iterator"  
externalInfoUrl="http://pmd.sourceforge.net/rules/optimizations.html#LocalVariableCouldBeFinal" priority="3">Local variable  
iterator could be declared final</violation>
```

```
<violation beginline="27" endline="27" begincolumn="25" endcolumn="71" rule="LocalVariableCouldBeFinal" ruleset="Optimization
```

```
Rules" package="br.com.da.dashboard.commons.config" class="MetricConfigSerializerFactory" variable="entry"  
externalInfoUrl="http://pmd.sourceforge.net/rules/optimizations.html#LocalVariableCouldBeFinal" priority="3">Local variable  
entry could be declared final</violation>
```

```
<violation beginline="28" endline="28" begincolumn="25" endcolumn="87" rule="LocalVariableCouldBeFinal" ruleset="Optimization
```

```
Rules" package="br.com.da.dashboard.commons.config" class="MetricConfigSerializerFactory" variable="clazz"  
externalInfoUrl="http://pmd.sourceforge.net/rules/optimizations.html#LocalVariableCouldBeFinal" priority="3">Local variable
```

```
<violation beginline="49" endline="49" begincolumn="74" endcolumn="90" rule="MethodArgumentCouldBeFinal" ruleset="Optimization
```

```
Rules" package="br.com.da.dashboard.commons.config" class="MetricConfigSerializerFactory"
```

```
method="getMetricConfigSerializerInstance"
```

```
externalInfoUrl="http://pmd.sourceforge.net/rules/optimizations.html#MethodArgumentCouldBeFinal"
```

```
priority="3">Parameter metricName is not assigned and could be declared final</violation>
```

```
- <file name="C:\projs\core-1.0\src\br\com\da\dashboard\commons\config\checkstyle\CheckstyleConfig.java">
```

```
<violation beginline="32" endline="32" begincolumn="32" endcolumn="44" rule="MethodArgumentCouldBeFinal" ruleset="Optimization
```

```
Rules" package="br.com.da.dashboard.commons.config.checkstyle" class="CheckstyleConfig" method="setConfig"  
externalInfoUrl="http://pmd.sourceforge.net/rules/optimizations.html#MethodArgumentCouldBeFinal"  
priority="3">Parameter config is not assigned and could be declared final</violation>
```

```
<violation beginline="40" endline="40" begincolumn="35" endcolumn="50" rule="MethodArgumentCouldBeFinal" ruleset="Optimization
```

```
Rules" package="br.com.da.dashboard.commons.config.checkstyle" class="CheckstyleConfig" method="setMaxErrors"
```

Principais resultados

- Modelo diferenciado

- ✓ Arquitetura extensível pelo cliente
- ✓ Modelo comercial Beta



- Aplicabilidade

- ✓ Carteira de 3 clientes de grande porte na área de TI
- ✓ Implantação interna integrado a build diária de desenvolvimento



- Qualificação da equipe

- ✓ Dissertação *latu sensu* apresentada na FGV





sensedia

SOA that makes sense!

www.sensedia.com/br

Copyright (C) 2009

Sensedia. - Todos os direitos reservados.

Todos os nomes e produtos são usados apenas com o propósito de identificação e são marcas registradas de seus respectivos proprietários.