

Optimal configuration and development

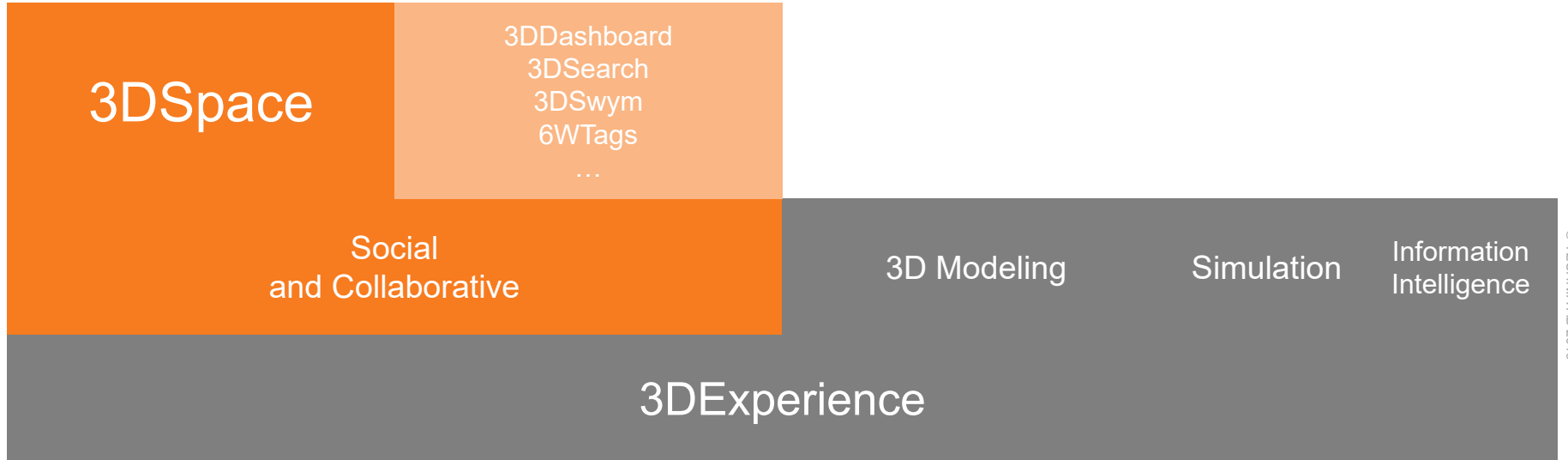
Dmitry Ponomarev
Application Developer, TECHNIA



Content

- Scope
- General aspects:
 - Performance
 - Code quality
- Code Management
- Productivity tools

Scope

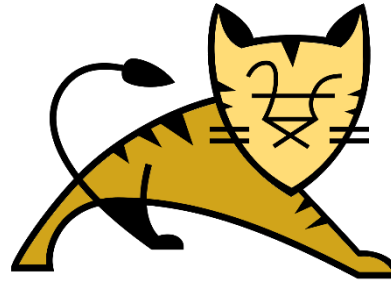


© TECHNIA AB 2019

Scope

Tech stack

- Java
- Java Server Pages
- JavaScript
- Apache Tomcat
- MQL/TCL
- Other (VBScript, EKL, ...)



Performance

Layers where developer can affect performance:

- Database interaction



- Java coding



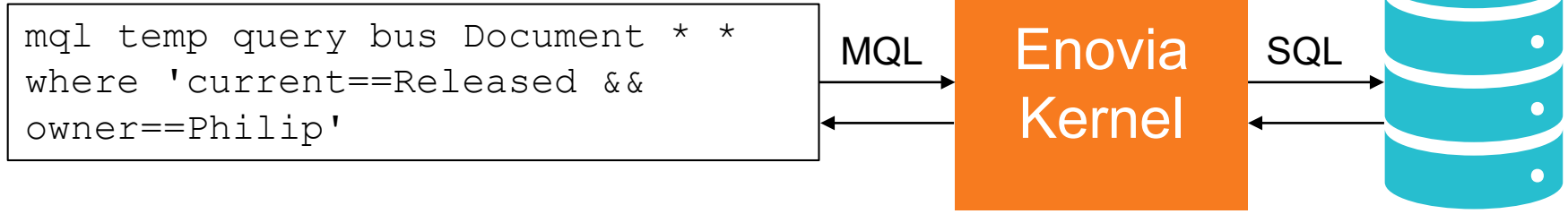
- Access model



Performance

Database interaction

- Main goal: minimize number of database calls



© TECHNIA AB 2019

Performance

Database interaction



- Example: OOTB table vs. TVC DataHandler

Type	Name	Revision	Description
Part	A-0001	1	Shaft
<code>select type</code>	<code>select name</code>	<code>select revision</code>	<code>select description</code>

4 queries

Type	Name	Revision	Description
Part	A-0001	1	Shaft
<code>select type, name, revision, description</code>			

1 query



Performance

Database interaction

- Example: never retrieve data in a loop

```
// Never do something like this:
```

```
String[] objectIds = ...
```

```
for (String objectId : objectIds) {
```

```
    String type = ObjectUtils.select(objectId, Statement.TYPE);
```

```
}
```

```
// Do instead:
```

```
String[] types = ObjectUtils.select(objectIds, Statement.TYPE);
```


Performance

Database interaction



- Example: limit on queries and expansions

```
mql temp query bus Part * * limit 1 where 'attribute[Title]==Shaft'
```

- If the same object is used several times it should be cached



Performance

Java coding tips



- Keep low memory imprint: don't create large amounts of objects unless necessary
- Use proper collection implementations
- Switching context is a slow operation
- Don't reinvent basic functionality (e.g. sorting methods)

Performance

Access model



- Is user a system administrator?
 - Does user person definition doesn't allow the action?
 - Does the revoke definition of the policy prevent the action?
 - Does the policy definition allow the action (public, owner, user)?
 - Has this user been granted the access?
 - Is there a rule preventing the action?



Performance

Tools

Technia Profiler

- Dynamic performance analysis

Set KPIs and follow up:

- No query over 4 seconds
- Max 20 queries per action

Profiler for ENOVIA PLM (Licensed to: dmpo01@technia.com Development)

Host: localhost Port: 9,000

Overview Trace Sessions Timeline

Description	Begin Time	Duration	Duration (%)	DB Duration	DB Calls
open.bosBusinessType	05:31:46,593 PM	.015	0.7%	.015	1
getPoliciesForPerson.bosBusinessType	05:31:46,608 PM	.016	0.7%	.016	1
print policy Manufacturer Equivalent select property[PolicyClassification].value dump	05:31:46,624 PM	.016	0.7%	.016	1
print policy EC Part select property[PolicyClassification].value dump	05:31:46,640 PM	.000	0.0%	.000	1
print policy Configured Part select property[PolicyClassification].value dump	05:31:46,640 PM	.015	0.7%	.015	1
print policy Standard Part select property[PolicyClassification].value dump	05:31:46,655 PM	.000	0.0%	.000	1
print policy Development Part select minorsequence dump	05:31:46,655 PM	.015	0.7%	.015	1
print policy EC Part select minorsequence dump	05:31:46,670 PM	.000	0.0%	.000	1
allocate context	05:31:46,670 PM	.000	0.0%	.000	1
set.bosContext	05:31:46,670 PM	.032	1.5%	.032	1
deleteWorkspace.bosContext	05:31:46,702 PM	.000	0.0%	.000	1
logout.bosContext	05:31:46,702 PM	.000	0.0%	.000	1
freeContext.bosInterface	05:31:46,702 PM	.000	0.0%	.000	1
allocate context	05:31:46,702 PM	.000	0.0%	.000	1

temp query bus Organization *select id dump |

Begin:	05:31:46,358 PM	DB Calls:	1
End:	05:31:46,577 PM	DB Duration:	0:00.219
Duration:	0:00.219	DB Duration (%):	100.0%
Duration (% of parent):	10.0%	DB Duration (% of parent):	10.4%
Duration (% of top):	10.0%	DB Duration (% of top):	10.4%

Overview Parameters Content Location Error

Code quality

Java coding tips

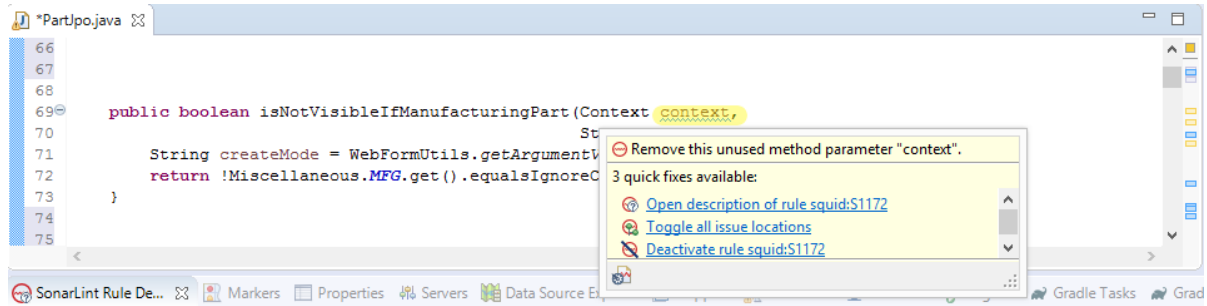
- Design for high cohesion and loose coupling
- Don't duplicate code
- Know the rules of using Enovia context
- Avoid nested transactions
- Store custom logic in POJOs instead of JPOs
- Methods should never return null
- Define detailed custom exceptions



Code quality

Tools: Static Code Analysis

- SonarLint
 - Detects bugs and bad practices
 - Instant feedback
 - Facilitates learning



Unused method parameters should be removed (squid:S1172)

Code smell Major

Unused parameters are misleading. Whatever the values passed to such parameters, the behavior will be the same.

Noncompliant Code Example

```
void doSomething(int a, int b) { // "b" is unused
    compute(a);
}
```

Compliant Solution

```
void doSomething(int a) {
    compute(a);
}
```

© TECHNIA AB 2019

Code quality

Tools: Static Code Analysis

- SonarQube
 - Detects bugs and bad practices
 - 25+ languages
 - Custom rule sets



The screenshot displays the SonarQube interface with the following components:

- Navigation:** Home, Technical Debt, Coverage, Duplications, Structure, Dashboards, Components, Issues.
- Issues Summary:**

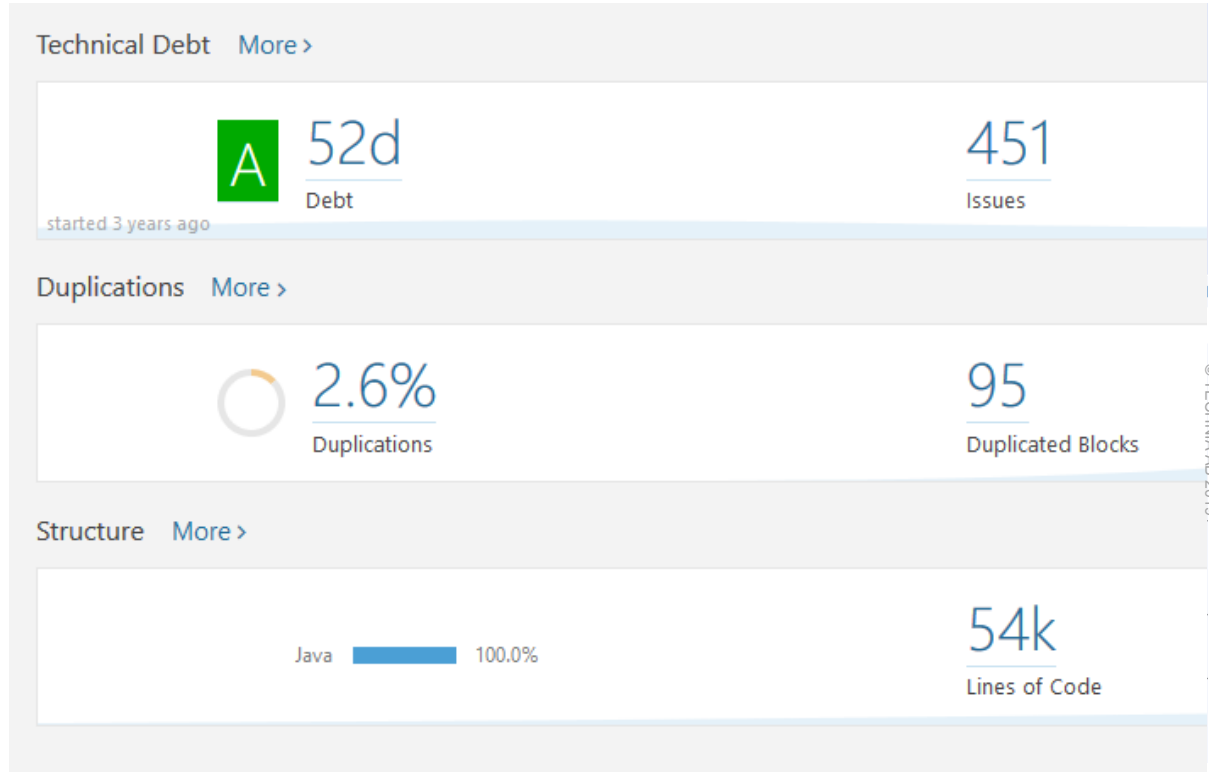
Severity	Count	Info	Count
Blocker	1	Minor	187
Critical	13	Info	0
Major	250		

Resolution	Count	Count	
Unresolved	451	Fixed	134
False Positive	0	Won't fix	0
Removed	0		
- Issue Details:**
 - Issue 1: "Reduce this lambda expression number of lines from 25 to at most 20." (Major, 20min debt)
 - Issue 2: "Remove the parentheses around the 'materialCATIA' parameter" (Minor, 2min debt)
 - Issue 3: "Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException" (Major, 5min debt)
- Rule Configuration Panel:**
 - Rule 1: "if ... else if" constructs should end with "else" clauses (Java, Code Smell, cert, misra) - **Activate**
 - Rule 2: Control structures should use curly braces (Java, Code Smell, cert, misra, pitfall) - **Activate**
 - Rule 3: Equality operators should not be used in "for" loop termination conditions (Java, Code Smell, cert, cwe, misra, suspicious) - **Activate**
 - Rule 4: Floating point numbers should not be tested for equality (Java, Bug, misra) - **Activate**
 - Rule 5: Functions should not be defined with a variable number of arguments (Java, Code Smell, cert, misra, pitfall) - **Activate**
 - Rule 6: Increment (++) and decrement (--) operators should not be used in a method call or mixed with other operators in an expression (Java, Code Smell, cert, misra) - **Activate**

Code quality

Tools: Static Code Analysis

- SonarQube
 - Code quality metrics
 - Technical debt tracking

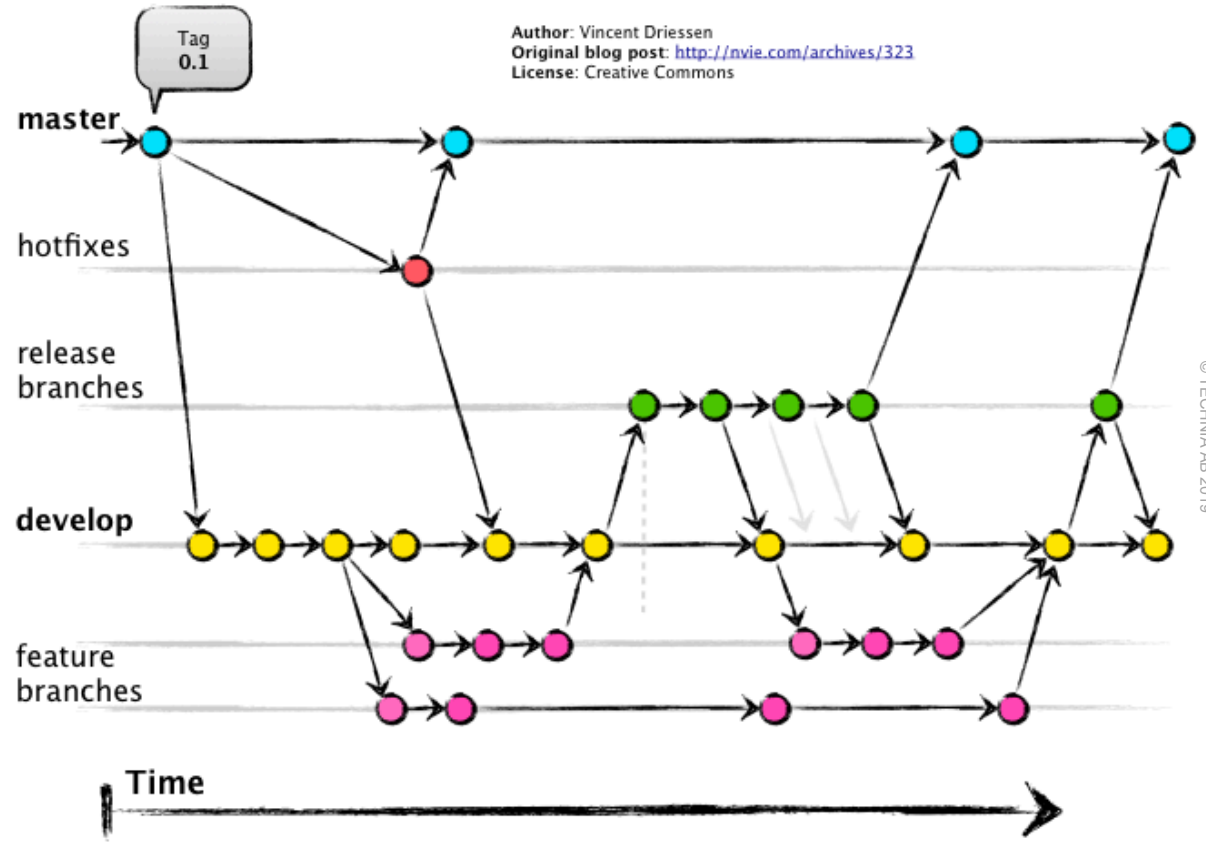


© TECHNIA AB 2019.

Code Management

Tools: Version Control

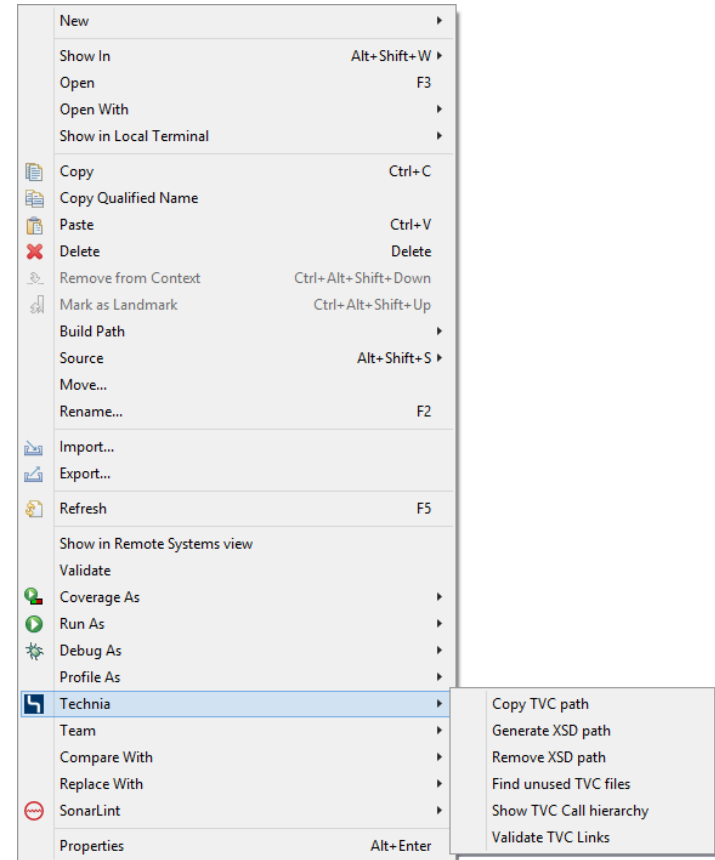
- GIT (GitFlow)
- Full history tracking
- Branching model



Productivity Tools

Eclipse plugin

- Faster and simpler configuration of TVCs:
 - Generates links for config files
 - Enables quick navigation through config files
 - Validates configs against schema
 - Finds related configurations
 - Finds unused files



Productivity Tools

Runtime class redefinition

- JRebel/DCEVM + HotswapAgent
 - “Hot deployment” of code changes:
 - Java classes
 - Resource files
 - Web application files

JRebel

Hot  wap



Optimal configuration and development Best practices for efficient and reliable 3DX configuration and customization



Dmitry Ponomarev
Application Developer
dmitry.ponomarev@technia.com



DIGITALIZATION FOR SUSTAINABLE PRODUCT DEVELOPMENT AND MANUFACTURE