JOHANNES KEPLER
UNIVERSITY LINZ

Praktikum Software Engineering

Johannes Bräuer

Sprint 1 Review, Release 2 Planung

JOHANNES KEPLER UNIVERSITY LINZ

Agenda

- Feedback Sprint 1 / Release 1
- Fokus Sprint 2 / Release 2
- Team Präsentationen
  → + 5min Team Meeting

© 2025 Business Informatics - Software Engineering

Feedback zu Sprint 1 / Release 1

<u>Jedes Teammitglied</u> hat Implementierungsaufgaben zu übernehmen – auf annähernd gleiche Verteilung achten!

- Commit (Pull Request) mindestens <u>1x pro Woche</u> von jedem Teammitglied
- <u>Lauffähige Builds</u> in GitHub Actions (läuft `mvn package`?)
  - Build limits für private Repositories → darum sind die Repos nun public
- <u>Unit Tests</u> für einzelne Klassen und automatisiert über GitHub Actions ausführen (läuft `mvn test`?)
- Issues geschlossen und dem Milestone zugeordnet

© 2025 Business Informatics - Software Engineering

Feedback zu Sprint 1 / Release 1

- Tickets/Issues haben eine Beschreibung (I as a user,..) und Acceptance Criteria nicht nur einen Titel
- Zeit auf Clockify buchen

Nachzureichen sind:

- Markdown-File von Tool-Präsentation im Repo speichern
- Link zum Prototyp
- Markdown-File oder Link zum UML Diagramm

© 2025 Business Informatics - Software Engineering

Sprint 2 - 2. Mai

- **Fokus: Dokumentation & Codequalität**
- **Deliverables**
  - ➜ Erweiterte/aktualisierte UML Diagramme, Unit Tests
  - ➜ Implementierung
    - Ca. 2/3 des Funktionsumfangs umgesetzt
  - ➜ Analyse der Codequalität mit PMD (SonarCloud)
  - ➜ Erste Version der Projektdokumentation: (1) Systemdokumenation, (2) Benutzerdokumenation
  - ➜ Planung Release 3
  - ➜ Live Demo der Applikation

© 2025 Business Informatics - Software Engineering

PMD

PMD ist ein Programmierwerkzeug für die statische Codeanalyse von Quelltexten.

Es findet häufige Programmierfehler wie zum Beispiel:

- Mögliche Bugs – Leere try/catch/finally/switch-Blöcke.
- Toter Code – Ungenutzte Lokale Variablen, Parameter und private Methoden
- Leere if/while-Ausdrücke
- Überkomplizierte Ausdrücke – Unnötige if-Ausdrücke, oder for-Schleifen, die stattdessen als while-Schleifen genutzt werden könnten
- Suboptimaler Code – verschwenderische Nutzung von String/StringBuffer
- Klassen mit hoher Zyklomatischer Komplexität.

© 2025 Business Informatics - Software Engineering

PMD in GitHub Actions integriert

Summary

Jobs

continuous_integration

Run details

Usage

Workflow file

Triggered via pull request 11 hours ago

johannes-b synchronize #5 pmd-test

Status	Total duration	Artifacts
Failure	23s	1

continuous_integration.yaml

on: pull_request

continuous_integration 15s

Annotations

5 errors and 10 warnings

continuous_integration
Process completed with exit code 1.

The final field name 'btnNewButton_1' doesn't match '[a-z][a-zA-Z0-9]*':
at.jku.se.prse.example.calculator/src/main/java/at/jku/se/prse/example/calculator/SimpleCalculator.java#L41
Configurable naming conventions for field declarations. This rule reports variable declarations which do not match the regex…
Show more

The local variable name 'panel_2' doesn't match '[a-z][a-zA-Z0-9]*':
at.jku.se.prse.example.calculator/src/main/java/at/jku/se/prse/example/calculator/SimpleCalculator.java#L122
Configurable naming conventions for local variable declarations and other locally-scoped variables. This rule reports variab…
Show more

Logger calls should be surrounded by log level guards.:
at.jku.se.prse.example.calculator/src/main/java/at/jku/se/prse/example/calculator/operators/AddOperation.java#L15
Whenever using a log level, one should check if the loglevel is actually enabled, or otherwise skip the associate String cre…
Show more

The local variable name 'panel_1' doesn't match '[a-z][a-zA-Z0-9]*':
at.jku.se.prse.example.calculator/src/main/java/at/jku/se/prse/example/calculator/SimpleCalculator.java#L109
Configurable naming conventions for local variable declarations and other locally-scoped variables. This rule reports variab…
Show more

Avoid using Literals in Conditional Statements:
at.jku.se.prse.example.calculator/src/main/java/at/jku/se/prse/example/calculator/operators/AddOperation.java#L17
Avoid using hard-coded literals in conditional statements. By declaring them as static variables or private members with des…
Show more

Avoid empty catch blocks:
at.jku.se.prse.example.calculator/src/main/java/at/jku/se/prse/example/calculator/SimpleCalculator.java#L273
Empty Catch Block finds instances where an exception is caught, but nothing is done.
In most circumstances, this swallows an exception which should either be acted on
or reported.

EmptyCatchBlock (Priority: 3, Ruleset: Error Prone)
https://pmd.github.io/pmd-6.40.0/pmd_rules_java_errorprone.html#emptycatchblock
Show less

Filter changed files

- .github/workflows
  - continuous_integration.yaml
- at.jku.se.prse.example.calculator/src…
  - SimpleCalculator.java
  - factory
    - ICalculationOperation.java
    - OperationFactory.java
  - operators
    - AddOperation.java
    - NullOperation.java
- ruleset.xml

2 ....example.calculator/src/main/java/at/jku/se/prse/example/calculator/SimpleCalculator.java

Viewed

```
270 270             }
271 271             try {
272 272                 txt = new Double(txt).intValue() + "";
273 273             } catch (NumberFormatException ex) {
274 274             }
```

Check warning on line 274 in at.jku.se.prse.example.calculator/src/main/java/at/jku/se/prse/example/calculator/SimpleCalculator.java

GitHub Actions / continuous_integration

Avoid empty catch blocks

Empty Catch Block finds instances where an exception is caught, but nothing is done.
In most circumstances, this swallows an exception which should either be acted on
or reported.

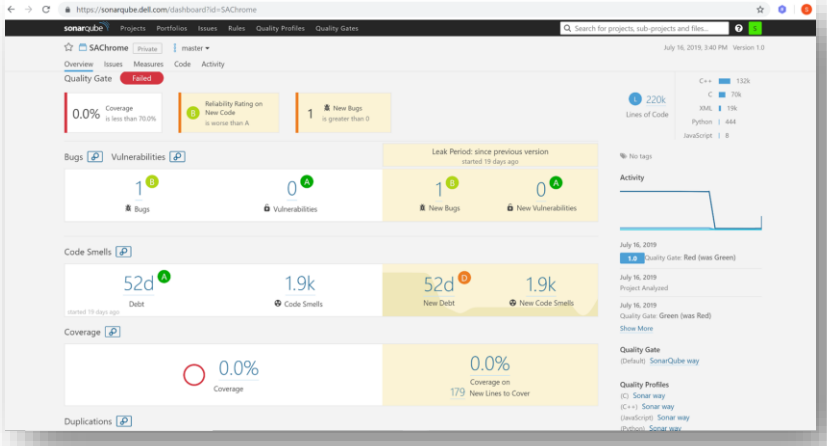
EmptyCatchBlock (Priority: 3, Ruleset: Error Prone)
https://pmd.github.io/pmd-6.40.0/pmd_rules_java_errorprone.html#emptycatchblock

```
275 275             }catch(IllegalArgumentException e) {
276 276                 txt = "Error";
277 277             }
```


© 2025 Business Informatics - Software Engineering

SonarCloud

- Plattform für die statische Analyse und Bewertung der technischen Qualität von Sourcecode.
- Analysiert den Quelltext hinsichtlich verschiedener Qualitätsbereiche und stellt die Ergebnisse über eine Plattform dar.
- Einbindung von SonarQube in den Entwicklungsprozess und eine Ermittlung der Qualitätsmetriken auf einem Build-Server für die kontinuierliche Integration.
  - ➜ GitHub Actions



© 2025 Business Informatics - Software Engineering

Nächste Schritte

- **Bis 29.4.2025 - 12:00**
  - ➜ Implementierung (+Test) geplanter User-Stories für Sprint 2
  - ➜ `git checkout -b release-v0.2.0`

- **Sprint 2 Review und Planung Release 3 am 2.5.2025**
  - ➜ Template (Template-ReleasePräsentation2.pptx) verwenden

© 2025 Business Informatics - Software Engineering

Team Präsentationen

- Team 5
- Team 2
- Team 3
- Team 1
- Team 4

JOHANNES KEPLER
UNIVERSITY LINZ