

# **Dokumentation & Projekttagbuch**

Innovation Lab 3  
Jahr 2022/23

Projekt: **Code Quality Tools**

Team: **22**

# 1. Allgemeine Informationen

**Projektname:** Code Quality Tools

**Supervisor:** Dipl.-Ing. (FH) Bernhard Wallisch

Innovation Lab 3, Wintersemester 2022/23

**Projektteam:**

Taha Samaha, [if20b150@technikum-wien.at](mailto:if20b150@technikum-wien.at), Projektleitung

Fatema Aly, [if20b057@technikum-wien.at](mailto:if20b057@technikum-wien.at)

Jan Loos, [if20b118@technikum-wien.at](mailto:if20b118@technikum-wien.at)

Alexander Kramreiter, [if21b065@technikum-wien.at](mailto:if21b065@technikum-wien.at)

## Management-Summary des Projektes

Das Projekt dient zur Reduzierung des Korrekturaufwands von PHP und Java-Quellcode und soll zur Verbesserung der Code Qualität von Studenten beitragen. Der Fokus liegt primär auf Sicherheit. Klassische Programmierfehler, veraltete oder unsichere Funktionen sollen erkannt und verständlich aufgearbeitet werden, damit Probleme übersichtlich und leicht zu beheben sind. Nach Möglichkeit sollen auch bewährte Programmiertechniken (so genannte „Best Practices“) angewendet werden.

## Rahmenbedingungen und Projektumfeld

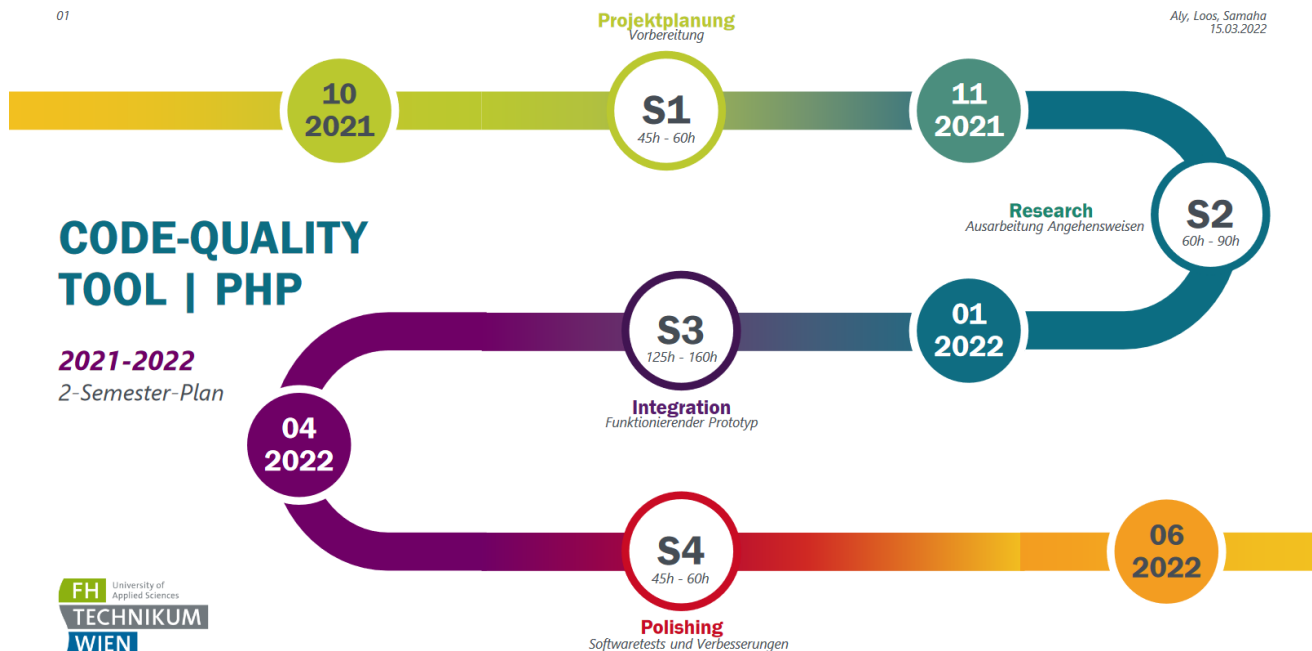
Die Software wird in einem Docker Container laufen. Die Ausgabe sollte in einer von EspressoIT ausgelegten HTML erfolgen. Die Code Quality wird hinsichtlich Safety und Security getestet, dies kann man sowohl mit Unit-Tests, also auch Code-Style oder Code Patterns bzw. best practice Ansätzen überprüft werden. In diesem Projekt wäre es Ziel hinsichtlich PHP und Java eine CI/CD Pipeline zu erzeugen, die es Studierenden ermöglichen soll ihren Quellcode inklusive weiteren Aspekten einfach über das bestehende EspressoIT-System zu überprüfen. Dieser Prozess soll zu einer verbesserten Codequalität führen und als Resultat eine Bewertung in Form eines einheitlichen HTML-Exports liefern.

## Semester-Roadmap

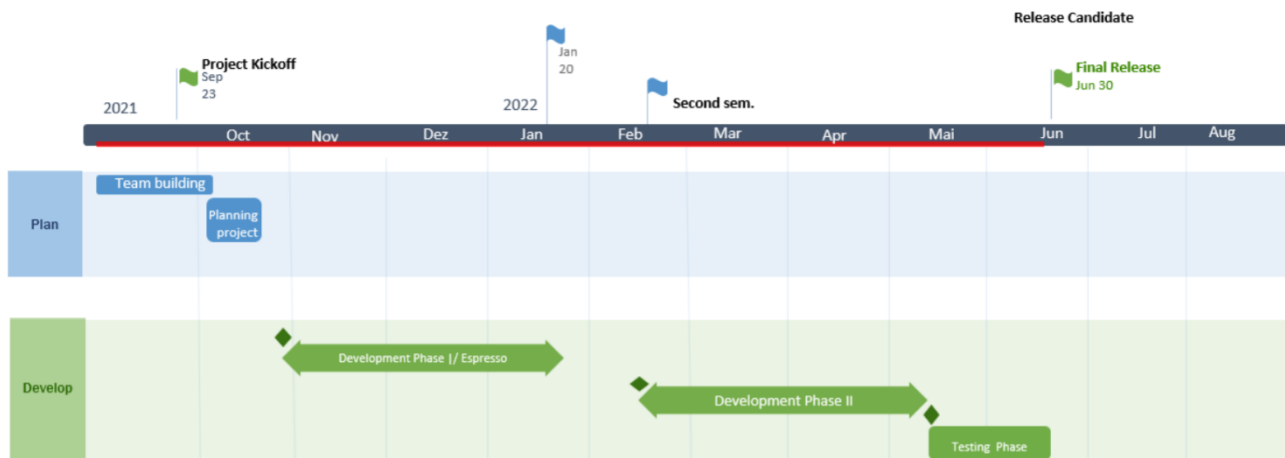
### Aktueller Plan, Stand WS2022/23



### INNO2 Plan, Stand SS2022



## Code quality Tool – PHP



### Collaboration & Tooling

ALM: [https://dev.azure.com/Fatema-aly/Inno\\_Lab1](https://dev.azure.com/Fatema-aly/Inno_Lab1)

Trello: <https://trello.com/codequalitytoolsphp>

GitHub: [https://github.com/Alyf92/innoLab\\_1](https://github.com/Alyf92/innoLab_1)

### Espresso-IT Tool Integration

<https://git.technikum-wien.at/innolab/espresso-it/espresso-it/-/tree/feature/php-multi-linter>

[https://git.technikum-wien.at/innolab/espresso-it/espresso-it/-/tree/feature/java\\_checkstyle](https://git.technikum-wien.at/innolab/espresso-it/espresso-it/-/tree/feature/java_checkstyle)

## 2. Projekt-Kurzbeschreibung

Ziel des Projekts ist es eine Verbesserung oder eine Erweiterung für das Moodle-Plugin EspressoIT zu entwickeln, um zusätzliche Code Quality und Safety Funktionen für PHP und Java-Dateien zu unterstützen. Die Dateien sollen automatisch im Hintergrund auf vorgefertigte Kriterien getestet werden und anschließend soll das Resultat in ein maschinenlesbares Format ausgegeben werden. Dies soll den Korrekturaufwand für Lektoren minimieren und Studenten übersichtliches und verständliches Feedback zu ihrem Code bieten.

Die größte Herausforderung wird bei der Integration in das bereits bestehende EspressoIT erwartet. Zugriff auf den Quellcode von EspressoIT und Testumgebungen wurden zur Verfügung gestellt, jedoch ist zurzeit noch nicht bekannt, wie kompliziert sich die Integration unserer Software, die wir im letzten Semester geschrieben haben, herausstellen wird. Es wird gegebenenfalls erstmal der fremde Quellcode studiert werden müssen.

Es war ursprünglich das Ziel zum Ende des Innovation Lab 2 erfolgreich PHP und Java Dateien mit EspressoIT zu verarbeiten, so dass am Ende ein Resultat herauskommt. Jedoch hat sich im Verlauf des Semesters herausgestellt, dass EspressoIT für die Weiterentwicklung nicht geeignet war und uns daher keine Ressourcen zur Entwicklung zur Verfügung gestellt werden konnten, weshalb stattdessen eine eigene Software entwickelt wurde.

In Innovation Lab 3 wird versucht unsere eigene Software in EspressoIT zu integrieren. Bis zum Abschluss des Semesters soll es Lektoren möglich sein bei Abgaben auf Moodle unsere Software in EspressoIT auszuwählen und Studierende sollen dazu in der Lage sein, Feedback für ihren Quellcode zu erhalten.

Um den besten Wert für AnwenderInnen bieten zu können, sollte das generierte Feedback qualitativ hochwertig und gut strukturiert sein. Schwerwiegende Fehler dürfen nicht in einer Flut Folgefehler untergehen. Das generierte Feedback sollte objektiv bleiben und sich möglichst auf Fakten beruhen und dem Anwender keine subjektiven Veränderungen aufzwingen.

Es ist wichtig klarzustellen, dass die Software nicht eigenständig Fehler behebt, sondern lediglich die Nutzer auf ihre Fehler aufmerksam machen soll. Unter gewissen Umständen, zum Beispiel bei veralteten oder unsicheren Funktionen, kann die Software einen Verbesserungsvorschlag beziehungsweise mehr Informationen über die Fehler anbieten.

Da für die Umsetzung bereits existierende quelloffene Software in EspressoIT integriert wird, muss die Funktionsweise von EspressoIT analysiert und gegebenenfalls angepasst werden.

Am Ende des Projekts soll die Software von einem Docker Container aus im Hintergrund gestartet werden können. Die Software soll PHP Dateien auswerten und Ergebnisse in einer von EspressoIT ausgelegten HTML ausgeben.

### 3. Spezifikation der Lösung

Während der Auftragsklärung wurde festgelegt, dass unser im Innovation Lab 2 fertiggestelltes Tool vollständig in EspressoIT und im Optimalfall das Tool Cocoa implementiert werden soll und keine eigene Schnittstelle entwickelt werden muss. Nach wie vor soll das Tool in einem eigenen Docker Container ausgeführt werden, isoliert von dem unterliegenden Betriebssystem, genauso wie andere Module von EspressoIT dies tun.

Lektoren sollen die Möglichkeit haben, unser Tool für eine Abgabe zu aktivieren, ähnlich wie es bereits für andere Programmiersprachen, wie zum Beispiel C, der Fall ist. Studenten sollen nach der Abgabe ihres Quellcodes dazu in der Lage sein, zeitnah Verbesserungsvorschläge einsehen zu können. Dabei sollten diese so detailliert wie möglich beschrieben werden, damit Studenten die Stellen im Quellcode schnell identifizieren können.

#### ▼ ESprESSO Plugin

Enable ESprESSO Plugin?

Submission type

Testcase tool

Status

ESprESSO-IT Result for student1

Findings

PHPML found 39 findings!

Show Findings

file	line	message	verbose	severity	bugclass	code
TenSampl es.php	11	The method mistake_1 is not named in camelCase.			PHPMD - Controversial Ruleset	public function mistake_1() { \$arr = array(1, 2, 3, 4); foreach (\$arr as \$value) { \$value = \$value * 2; } // \$arr is now array(2, 4, 6, 8) \$array = [1, 2, 3]; echo implode(',', \$array), "\n"; foreach (\$array as \$value) { // by reference echo implode(',', \$array), "\n"; foreach (\$array as \$value) { } // by value (i.e., copy) echo implode(',', \$array), "\n";
TenSampl es.php	11	Method Mistakes:mistake_1() has no return type specified.			PHPStan - MAX Level	public function mistake_1()

Ansicht Student / Lektor

Für PHP konnte unser Tool wie beschrieben implementiert werden. Es wird in einem Docker Container ausgeführt und kommuniziert über die REST-Schnittstelle mit dem Businesslayer. Der Businesslayer sendet periodisch abgegebenen Quellcode an das Tool und enthält Resultate, die über EspressoIT einheitlich wiedergegeben werden können.



Vereinfachtes Sequenz Diagramm

Für Java konnte dieses Funktionalitätslevel nicht erreicht werden, da von der Java Gruppe die meisten ausgestiegen sind. Das Tool wird zwar in einem Docker Container geladen, ist aber noch nicht mit Cocoa eingebunden. Es werden erfolgreich Resultate generiert, jedoch werden diese nicht über EspressoIT, sondern zurzeit noch nur als HTML-Ausgabe wiedergegeben.

Das Tool für PHP ist erweiterungsfähig, das heißt, dass weitere Linter eingebunden werden können, um genauere Resultate zu generieren oder bestimmte Fehler zu der Ausgabe hinzuzufügen. Dafür muss lediglich der jeweilige Code Block dupliziert und für die Ausgabe des jeweiligen Linters angepasst werden. Aktuell ist das Tool noch nicht parallelisiert, jedoch sollte es keine größere Herausforderung darstellen dies zu erreichen. Es konnten auch ohne Parallelisierung keine Mängel bei der Leistung oder Verfügbarkeit festgestellt werden.

Sollte sich doch dazu entschieden werden EspressoIT entweder neuzuschreiben oder zu ersetzen, befindet sich in der Git History des PHP-Multilinters Quellcode Ausschnitte, die genutzt werden können, um HTML-Resultate zu generieren. Diese sähen dann so aus:

## Results

PHPMD - Controversial Ruleset - Line: 11	The method mistake_1 is not named in camelCase.
PHPMD - Controversial Ruleset - Line: 31	mistake_2 accesses the super-global variable \$_POST.
PHPMD - Controversial Ruleset - Line: 31	The method mistake_2 is not named in camelCase.
PHPMD - Controversial Ruleset - Line: 57	The method mistake_3 is not named in camelCase.
PHPMD - Controversial Ruleset - Line: 68	The method mistake_4 is not named in camelCase.
PHPMD - Controversial Ruleset - Line: 85	The method mistake_5 is not named in camelCase.
PHPMD - Controversial Ruleset - Line: 92	mistake_6 accesses the super-global variable \$_POST.
PHPMD - Controversial Ruleset - Line: 92	The method mistake_6 is not named in camelCase.
PHPMD - Controversial Ruleset - Line: 98	mistake_7 accesses the super-global variable \$_POST.
PHPMD - Controversial Ruleset - Line: 98	The method mistake_7 is not named in camelCase.
PHPMD - Controversial Ruleset - Line: 104	The method mistake_8 is not named in camelCase.
PHPMD - Controversial Ruleset - Line: 113	mistake_9 accesses the super-global variable \$_GET.
PHPMD - Controversial Ruleset - Line: 113	The method mistake_9 is not named in camelCase.
PHPStan - MAX Level - Line: 130	Variable \$array in empty() always exists and is not falsy.

*Ergebnis der ehemaligen HTML-Ausgabe*

## 4. Aufwandschätzung

Diese Aufwandsschätzung wurde mit der Delphi Methode durchgeführt, um den Aufwand für dieses Semester zu schätzen. Dabei haben wir intern im Team Brainstorming angewandt, um die kommenden Arbeitspakete zu erfassen. Eine Hierarchiestufe darunter zählten wir die Epics auf und in der letzten Stufe die User Story oder die Beschreibung der Aufgabe. Dieses Szenario durchliefen wir einige Male, um auf alle einzelnen Aufgaben einzugehen. Anschließend haben wir gemeinsam als Team die Stunden für die Spalten Optimistisch (Sp), Wahrscheinlich (Sp) und Pessimistisch (Sp) geschätzt. Als Schlussfolgerung dieser Schätzung ist uns noch deutlicher geworden, dass die Aufwandsschätzung von der EspressoIT Implementation für uns schwer zu schätzen ist, in Hinsicht vom Zeitaufwand und den nötigen Arbeitspaketen. Hier ein Screenshot und in der Anlage befindet sich die Excel Datei.

Themes / Areas / Arbeitspakete Toplevel	Epics / User Stories / Arbeitspakete TopLevel	User Stories / Detail Level / Beschreibung	Optimistisch (Sp)	Wahrscheinlich (Sp)	Pessimistisch (Sp)
			85,0	123,0	162,0
<b>Projektumgebung</b>					
	Ziele definieren				
		Absprache mit Supervisor	4	6	8
		Absprache im Team	4	6	8
	<b>Entwicklungsumgebung</b>				
		Espresso-IT lokal einrichten (WLS2)	3	4	5
	<b>Dokumentation</b>				
		Projektanbahnung	3	4	5
		Paper schreiben	5	6	7
		Marketingvideo drehen/schneiden	6	8	10
		Präsentation vorbereiten	4	6	8
<b>Integration</b>					
	<b>Tool an EspressoIT anpassen</b>				
		Code Anpassungen	4	6	8
		Design Anpassungen	4	6	8
		Integration vom Tool in EspressoIT	5	10	15
<b>Deployment</b>					
	<b>Deployment in den Test Server</b>				
		Deployment	10	15	20
<b>Testing</b>					
	<b>Softwaretests</b>				
		Eingabetests (Upload)	4	5	6
		Ausgabeteests (Lint Result)	4	5	6
	<b>Software Peer Review</b>				
		Code Review	2	3	5
		Inspektion	3	4	5
		Plagiat Check	2	3	4
<b>Tool Nutzung</b>					
	<b>Code Übergabe</b>				
		Der User sollte eine PHP Datei zum linten hochladen können	4	6	8
		Der User sollte im CodeRunner geschriebenen Code linten können.	4	6	8
	<b>Lint Ergebnis</b>				
		Der Lektor sollte das Lint Ergebnis im Moodle aufrufen können	5	7	9
		Der Lektor sollte das Lint Ergebnis vom Moodle exportieren können	5	7	9

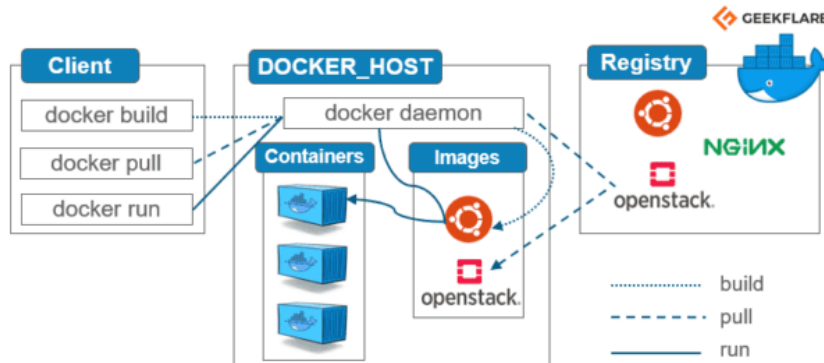


# 5. Auslieferung

Die fertige Lösungskomponente von PHP und unsere angefangene Implementation für Java können jeweils von den Branches `feature/php-multi-linter` und `feature/java_checkstyle` vom Espresso-IT Git Repository entnommen werden: [PHP-Git-Link](#) [Java-Git-Link](#)

Hierfür ist jedoch eine VPN-Verbindung zum FH Technikum-Wien nötig. Damit Sie von extern im Intranet der FH Technikum-Wien arbeiten können, wurde ein VPN (Virtual Private Network) Zugang eingerichtet. Dieser Dienst steht für Studierende, MitarbeiterInnen und Lehrende der FH Technikum Wien zur Verfügung. Für den Zugang benötigen Sie das FH Technikum-Wien Zertifikat [rulebreaker.ca.crt](#) und die VPN Konfigurationsdatei [rulebreaker.ovpn](#). Die VPN Client Software (openvpn) und die dazugehörige VPN Konfigurationsdatei können Sie per Klick auf das jeweils verwendete Betriebssystem herunterladen. Bitte denken Sie daran, Ihr Browsercache zu löschen, falls die ursprünglich von Ihnen gewünschte Seite nicht erreicht werden kann. Sollten Sie Probleme beim Verbinden mit dem VPN haben, ist der Support unter der E-Mail-Adresse [support@technikum-wien.at](mailto:support@technikum-wien.at) für Sie erreichbar.

Die [Docker Architektur](#) erklärt die Systemarchitektur des Espresso-IT und den dazugehörigen Tools im Projekt sehr gut. Beim Docker-Container haben Sie jetzt ein einziges Betriebssystem, und die Ressourcen werden von den Containern gemeinsam genutzt. Daher ist es klein und lädt in wenigen Sekunden. Unten sehen Sie das einfache Diagramm einer Docker-Architektur.



Das Installieren des gesamten Systems wurde ausführlich im [Git-Wiki](#) beschrieben. Das Tool PHPML läuft automatisch mittels eines Dockerfile vom Ordner [tool\\_php\\_multilinter](#) aus als Container.

## Verwendete Dritt-Software / Frameworks

- PHPMD <https://phpmd.org> By Manuel Pichler licensed under BSD 3-Clause
- PHPStan <https://phpstan.org> © 2016–2023 Ondřej Mirtes
- Checkstyle (Java) <https://checkstyle.org/> Licensed under GNU Lesser GPL

## System Anforderung

Hier halten wir uns an den Systemanforderungen von Docker Desktop bzw. Docker WSL

- Für Windows siehe: <https://docs.docker.com/desktop/install/windows-install/>
- Für Linux siehe: <https://docs.docker.com/desktop/install/linux-install/>
- Für Mac siehe: <https://docs.docker.com/desktop/install/mac-install/>

# 6. Unser Projekt-Tagebuch

## Sprint 1 bis 3

### Taha Samaha:

Meine Aufgaben waren Kommunikation mit unserem Betreuer, Aufgabenaufteilung und die erste Anlaufstelle bei Komplikationen zu sein. Die erste Espresso-IT/Moodle Einrichtung wurde im Laufe eines Zoom Calls auf meinem lokalen PC eingerichtet und diente zugleich als Anleitung für meine Teamkollegen. Momentan hat das gesamte Team das Projekt lokal eingerichtet. Nächste Aufgabenschritte wurden mit unserem Betreuer besprochen und nach Austausch im Team sind wir zur folgenden Aufteilung gekommen:

- Docker Container - (Java: Kramreiter, PHP: Samaha)
- tool.py / run file - (Java: Aly, PHP: Loos)

Weiters machen wir uns noch mit der Projektumgebung vertraut und notieren uns aufkommende Fragen.

### Jan Loos:

Für das PHP Tool Skript konnten wir unsere Arbeit aus den ersten beiden Innovation Labs zum größten Teil übernehmen. Konkret musste unser Code dafür lediglich in Funktionen eingepackt und ein paar Variablen ausgetauscht werden. Wir haben das tatsächliche Auswerten der Dateien und die Integration in EspressoIT in zwei getrennten Skripten gelassen, da es unseres Erachtens nach den Code übersichtlicher macht. Um die Funktionen von EspressoIT weitestgehend ausnutzen zu können, haben wir außerdem das Auslesen der tatsächlichen Zeilen Code in unserem Run Skript mit aufgenommen. Für das Tool Skript selbst, haben wir uns bereits zuvor geschriebene Tool Skripts angeschaut und das nötigste abgeschaut. Während der Entwicklung ist es zu keinen Problemen gekommen und auch vereinzelt Testen verlief einwandfrei, jedoch wurde das Tool Skript noch nicht in Zusammenhang mit EspressoIT getestet werden können.

### Fatema Aly:

Da die meisten von uns in den letzten beiden Innovation Labs nicht mit dem Java-Tool gearbeitet haben, sollten wir uns für das Java-Tool-Skript zunächst mit das von der anderen Gruppe ausgewählte Tool beschäftigen. Zu diesem Zweck wurde Java eingerichtet und das Tool Checkstyle lokal installiert. Beim Installieren und Starten des Tools stießen wir auf einige Fehlermeldungen, z.B.: **Error "A JNI error has occurred. Please check your installation and try again"**. Das Problem war, dass Java JDK 17 bereits installiert war. Gleichzeitig hatten wir aber Java 8 (JRE) installiert. Die Fehlermeldung konnte durch die Deinstallation von JAVA 8 (JRE) dann behoben werden. Für die Implementierung des Java Tool-Skripts selbst haben wir uns das PHPStan Tool Skript, das von den Lektoren auf GIT geschrieben wurde, genauer angeschaut sowie weitere Tools in Python online recherchiert und uns mit denen auseinandergesetzt, um bald mit der Entwicklung des Tools beginnen zu können.

## Sprint 4

### Taha Samaha

Mithilfe des bestehenden Branchs `feature/phpstan` konnten wir sehen, wie die zu erstellenden Dateien ausschauen sollen. Da unser zusammengesetztes Tool PHPStan und PHPMD nutzt, habe ich mir erlaubt das Dockerfile von dieser Branch zu nutzen und anzupassen. Für die Konfiguration des Docker Containers habe ich Zeilen in den Dateien **.env (Port)**, **initial\_db\_data.json** und **docker-compose.yml** eingetragen. Beim Builden des Docker Container kamen einige Komplikationen auf, jedoch waren diese größtenteils Tippfehler und wurden dann korrigiert. Momentan ist der Container buildfähig, jedoch besteht noch ein Package Fehler im `tool.py` File, worum sich mein Kollege Jan Loos kümmert. (siehe Screenshot). Währenddessen versuche ich aufkommende neue Begriffe nachzuschlagen und zu verstehen. Unser nächster Schritt ist das Testen des integrierten Tools am Testserver.

```
[2022-12-06 17:50:25 +0000] [7] [INFO] Starting gunicorn 20.0.4
[2022-12-06 17:50:25 +0000] [7] [INFO] Listening at: http://0.0.0.0:5858 (7)
[2022-12-06 17:50:25 +0000] [7] [INFO] Using worker: sync
[2022-12-06 17:50:25 +0000] [9] [INFO] Booting worker with pid: 9
[2022-12-06 17:50:25 +0000] [9] [ERROR] Exception in worker process
Traceback (most recent call last):
  File "/usr/local/lib/python3.8/site-packages/gunicorn/arbiter.py", line 583, in spawn_worker
    worker.init_process()
  File "/usr/local/lib/python3.8/site-packages/gunicorn/workers/base.py", line 119, in init_process
    self.load_wsgi()
  File "/usr/local/lib/python3.8/site-packages/gunicorn/workers/base.py", line 144, in load_wsgi
    self.wsgi = self.app.wsgi()
  File "/usr/local/lib/python3.8/site-packages/gunicorn/app/base.py", line 67, in wsgi
    self.callable = self.load()
  File "/usr/local/lib/python3.8/site-packages/gunicorn/app/wsgiapp.py", line 49, in load
    return self.load_wsgiapp()
  File "/usr/local/lib/python3.8/site-packages/gunicorn/app/wsgiapp.py", line 39, in load_wsgiapp
    return util.import_app(self.app_uri)
  File "/usr/local/lib/python3.8/site-packages/gunicorn/util.py", line 358, in import_app
    mod = importlib.import_module(module)
  File "/usr/local/lib/python3.8/importlib/__init__.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1014, in _gcd_import
  File "<frozen importlib._bootstrap>", line 991, in _find_and_load
  File "<frozen importlib._bootstrap>", line 975, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 671, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 843, in exec_module
  File "<frozen importlib._bootstrap>", line 219, in _call_with_frames_removed
  File "/var/toolserver/tool_server.py", line 12, in <module>
    from tool import run_tool
  File "/var/cocoa/tool/tool.py", line 2, in <module>
    import run as tool
  File "/var/cocoa/tool/run.py", line 5, in <module>
    from dominate import document
ModuleNotFoundError: No module named 'dominate'
[2022-12-06 17:50:25 +0000] [9] [INFO] Worker exiting (pid: 9)
[2022-12-06 17:50:25 +0000] [7] [INFO] Shutting down: Master
[2022-12-06 17:50:25 +0000] [7] [INFO] Reason: Worker failed to boot.
```

### Fatema Aly:

In diesem Sprint begannen wir mit der Implementierung des Skripts für das Java-Tool, wobei wir auf mehrere Probleme mit Python stießen. Das erste Problem war, dass Python zwar auf Ubuntu VM installiert war, aber nicht funktionierte. Als wir versuchten, den Python-Interpreter in VSCode auf Ubuntu auszuwählen, zeigte VSCode diese Fehlermeldung an: "Python is not installed, please download and install it", daher beschlossen wir, das Skript unter Windows zu implementieren.

Mit `subprocess()` haben wir das Tool in Python unter Verwendung von einem Java-code Snippet ausgeführt und folgende Ergebnisse erhalten:

```

Starting audit...
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:5:1: Missing a Javadoc comment. [MissingJavadocType]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:5:1: The name of the outer type and the file do not match.
[OuterTypeFilename]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:9:5: 'member def modifier' has incorrect indentation level
4, expected level should be 2. [Indentation]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:10:5: 'member def modifier' has incorrect indentation leve
1 4, expected level should be 2. [Indentation]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:13:5: 'ctor def modifier' has incorrect indentation level
4, expected level should be 2. [Indentation]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:13:5: Missing a Javadoc comment. [MissingJavadocMethod]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:14:9: 'ctor def' child has incorrect indentation level 8,
expected level should be 4. [Indentation]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:15:9: 'ctor def' child has incorrect indentation level 8,
expected level should be 4. [Indentation]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:18:5: 'ctor def rcurlly' has incorrect indentation level 4,
expected level should be 2. [Indentation]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:20:5: 'method def modifier' has incorrect indentation leve
1 4, expected level should be 2. [Indentation]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:21:9: 'method def' child has incorrect indentation level 8
, expected level should be 4. [Indentation]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:22:5: 'method def rcurlly' has incorrect indentation level
4, expected level should be 2. [Indentation]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:24:5: 'method def modifier' has incorrect indentation leve
1 4, expected level should be 2. [Indentation]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:25:9: 'method def' child has incorrect indentation level 8
, expected level should be 4. [Indentation]
[WARN] C:\Users\Fatem\OneDrive\Desktop\java_sample.java:26:5: 'method def rcurlly' has incorrect indentation level
4, expected level should be 2. [Indentation]
Audit done.

```

Als nächstes wurden die Ergebnisse in einem XML-Report generiert. Jedoch wurde das Tool noch nicht fertig implementiert.

```

<?xml version="1.0" encoding="utf-8"?>
<checkstyle version="7.6.1">
  <file name="C:\Users\Fatem\OneDrive\Desktop\java_sample.java">
    <error line="5" column="1" severity="warning" message="Missing a Javadoc comment." source="com.puppycrawl.tools.checkstyle.checks.javadoc.MissingJavadocTypeCheck">
    </error>
    <error line="5" column="1" severity="warning" message="The name of the outer type and the file do not match." source="com.puppycrawl.tools.checkstyle.checks.OuterTypeFilenameCheck">
    </error>
    <error line="9" column="5" severity="warning" message="<code>member def modifier</code> has incorrect indentation level 4, expected level should be 2." source="com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck">
    </error>
    <error line="10" column="5" severity="warning" message="<code>member def modifier</code> has incorrect indentation level 4, expected level should be 2." source="com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck">
    </error>
    <error line="13" column="5" severity="warning" message="<code>ctor def modifier</code> has incorrect indentation level 4, expected level should be 2." source="com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck">
    </error>
    <error line="13" column="5" severity="warning" message="Missing a Javadoc comment." source="com.puppycrawl.tools.checkstyle.checks.javadoc.MissingJavadocMethodCheck">
    </error>
    <error line="14" column="9" severity="warning" message="<code>ctor def</code> child has incorrect indentation level 8, expected level should be 4." source="com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck">
    </error>
    <error line="15" column="9" severity="warning" message="<code>ctor def</code> child has incorrect indentation level 8, expected level should be 4." source="com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck">
    </error>
    <error line="18" column="5" severity="warning" message="<code>ctor def rcurlly</code> has incorrect indentation level 4, expected level should be 2." source="com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck">
    </error>
    <error line="20" column="5" severity="warning" message="<code>method def modifier</code> has incorrect indentation level 4, expected level should be 2." source="com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck">
    </error>
    <error line="21" column="9" severity="warning" message="<code>method def</code> child has incorrect indentation level 8, expected level should be 4." source="com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck">
    </error>
    <error line="22" column="5" severity="warning" message="<code>method def rcurlly</code> has incorrect indentation level 4, expected level should be 2." source="com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck">
    </error>
    <error line="24" column="5" severity="warning" message="<code>method def modifier</code> has incorrect indentation level 4, expected level should be 2." source="com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck">
    </error>
    <error line="25" column="9" severity="warning" message="<code>method def</code> child has incorrect indentation level 8, expected level should be 4." source="com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck">
    </error>
    <error line="26" column="5" severity="warning" message="<code>method def rcurlly</code> has incorrect indentation level 4, expected level should be 2." source="com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck">
    </error>
  </file>
</checkstyle>

```

## Sprint 5

### Taha Samaha

Nun läuft das Tool als eigener Container über Docker und ist betriebsbereit. Nachdem Jan Loos den Fehler vom Sprint 4 behoben hat, habe ich versucht die Änderungen an das neue Docker Image zu adaptieren jedoch lief es einige Male schief und habe letztendlich das komplette EspressolT neu aufgesetzt mit dem Kommando „docker-compose build“. Momentan teste ich das Tool durch das Moodle und zeichne die Ergebnisse auf.

Bemerkung: Um Docker in der Ubuntu Konsole nutzen zu können, muss Docker Desktop laufen (Screenshot).

### Jan Loos

Da das Docker Container für unseren PHP Multilinter bereit ist, kann nun das Plugin getestet werden. Damit das Plugin erfolgreich Ergebnisse ausgibt, mussten die Pfade der Linter angepasst werden, sodass die mit denen des Containers übereinstimmen. Außerdem musste die „dominate“ Implementierung auskommentiert werden, da diese zurzeit nicht benötigt wird. Des Weiteren wurde vergessen, die Ergebnisse überhaupt auszugeben. Beim ersten Testanlauf fiel auf, dass das Plugin denselben Fehler mehrmals ausgibt. Es stellte sich heraus, dass Python Variablen als Referenzen übergibt und die Ergebnisse daher konstant überschrieben hat. Dies wurde mit der copy Funktion von Python behoben. Auch aufgefallen ist, dass die Dateipfade falsch angezeigt werden, welches an einem Fehler in der Implementierung lag. Der Code wurde überarbeitet, damit solche Fehler in Zukunft seltener passieren. Ansonsten sind keine weiteren Fehler beim Testen aufgefallen und es wird davon ausgegangen, dass der PHP Multilinter wie erwartet funktioniert.

## Fatema Aly

Für das Java-Tool haben wir versucht, die Xml-Output zu parsen, was bisher leider nicht funktioniert hat. Zu diesem Zweck haben wir verschiedene Module ausprobiert, wie **xml.dom.minidom** und **lxml Ertree**, die uns nicht geholfen haben.

Dann stießen wir auf ein weiteres Modul, nämlich **xmldict**, was die Arbeit mit XML wie die Arbeit mit JSON erleichtert. Hier wurden wir mit der Fehlermeldung konfrontiert: "not well-formed (invalid token)".

```
12
13     with open("xml_file.xml") as xml_file:
14         data_dict = xmldict.parse(xml_file.read())
```

**Exception has occurred: ExpatError** ×  
not well-formed (invalid token): line 1, column 238  
File "C:\Users\Fatem\OneDrive\Desktop\FH\5.Sem\INNO\tool.py", line 14, in <module>  
data\_dict = xmldict.parse(xml\_file.read())

Um den Fehler zu beheben, wurden mehrere Wege ausprobiert, z.B. "replace", was eine Kopie zurückgibt, in der alle Vorkommnisse des Subprozesses durch neue ersetzt werden. Auch auf diese Weise konnte das Problem nicht behoben werden, deswegen wird weitergesucht, wie der Fehler behoben werden kann.

```
11 # run chekstyle and return the error report as xml
12 _result = subprocess.run('java -jar checkstyle-10.4-all.jar -c google_checks.xml java_sample.java -f xml', shell=True, capture_output=True).stdout
13
14 with open("xml_file.xml", "w") as write:
15     json.dump(_result, write, default=str)
16
17 # open the input xml file and read data in form of python dictionary
18 with open("xml_file.xml") as xml_file:
19     data_dict = xmldict.parse(xml_file.read())
20
21
22 json_data = json.dumps(data_dict)
23 with open("json_data.xml", "r") as json_file:
24     json.load(json_data)
25
```

## Sprint 6

### Taha Samaha

Anfangs des Sprints gab es einige Komplikationen mit der VPN-Verbindung und dem Moodle Container, der sich nach einigen Sekunden wieder herunterfuhr. Doch nach löschen aller Images und Container samt Hintergrunddaten und anschließendem Rebuilden lief er wieder einwandfrei. Beim Anlegen eines Assignments konnte die Config Files nicht an den Server gesendet werden. Nach dem Neueintragen der Espresso-IT URL in den Moodle Settings konnten die Config Files wieder gesendet werden. Weiters wurde noch an der Projekt Dokumentation gearbeitet.

#### ESprESSO-IT Result for student2

##### Findings

PHPMD found 39 findings!						
Show Findings						
file	line	message	verbose	severity	bugclass	code
TenSamples.php	11	The method mistake_1 is not named in camelCase.			PHPMD - Controversial Ruleset	public function mistake_1() { \$arr = array(1, 2, 3, 4); foreach (\$arr as \$value) { \$value = \$value * 2; } // \$arr is now array(2, 4, 6, 8) \$array = [1, 2, 3]; echo implode(',', \$array); "\n"; foreach (\$array as \$value) { // by reference echo implode(',', \$array); "\n"; foreach (\$array as \$value) { // by value (i.e., copy) echo implode(',', \$array); "\n";
TenSamples.php	11	Method Mistakes: mistake_10 has no return type specified.			PHPStan - MAX Level	public function mistake_1()
TenSamples.php	31	mistake_2 accesses the super-global variable \$_POST.			PHPMD - Controversial Ruleset	public function mistake_2() { \$data['keyShouldBeSet'] = 1; if (!isset(\$data['keyShouldBeSet'])) { // do something here if 'keyShouldBeSet' is not set } \$_POST['active'] = null; if (\$_POST['active']) { \$postData = var_dump(\$_POST); } // other example if (!isset(\$postData)) { echo 'post not active'; }

Lecturer:  
espressoit\_admin

Logout

#### ESprESSO-IT

##### Provided Configuration

Submission Type: PHP

Testcases: Not provided

Makefile: Not provided

##### Abgabeübersicht

Link

##### Plagiatprüfung

Ergebnisse

Es wurde noch keine Plagiatprüfung gestartet!

Neue Prüfung

### Jan Loos

Während dieses Sprints wurde der PHP Multilinter ausführlich getestet. Es wurden unterschiedliche Konfigurationen und Abgaben ausprobiert, um qualitative Unterschiede feststellen zu können. Es konnten keine auffallende Qualitätsverluste festgestellt werden, jedoch entgehen PHPMD und PHPStan manche Inhalte, die verbessert werden könnten, da beide Tools sehr spezialisiert sind und somit nicht alle möglichen Inhalte abdecken können. Es wäre daher vielleicht ratsam, den Multilinter um weitere Tools zu erweitern für noch ausführlichere und weitreichendere Resultate.

### Fatema Aly

In diesem Sprint wurde das Python Skript mittlerweile fertig implementiert und ist nun in der Lage das Checkstyle Tool auszuführen und die Ergebnisse temporär in einer HTML-Datei zu sortieren.



# Results

Checked File: c:\Users\Fatem\OneDrive\Desktop\FH\5.Sem\INNO\java\_tool\java\_sample.java

'member def modifier' has incorrect indentation level 4, expected level should be 2.	10
'ctor def modifier' has incorrect indentation level 4, expected level should be 2.	13
Missing a Javadoc comment.	13
'ctor def' child has incorrect indentation level 8, expected level should be 4.	14
'ctor def' child has incorrect indentation level 8, expected level should be 4.	15
'ctor def rcurl' has incorrect indentation level 4, expected level should be 2.	18
'method def modifier' has incorrect indentation level 4, expected level should be 2.	20
'method def' child has incorrect indentation level 8, expected level should be 4.	21
'method def rcurl' has incorrect indentation level 4, expected level should be 2.	22
'method def modifier' has incorrect indentation level 4, expected level should be 2.	24
'method def' child has incorrect indentation level 8, expected level should be 4.	25
'method def rcurl' has incorrect indentation level 4, expected level should be 2.	26
Missing a Javadoc comment.	5
The name of the outer type and the file do not match.	5
'member def modifier' has incorrect indentation level 4, expected level should be 2.	9

Als Nächstes begannen wir mit dem Dockerfile, wo wir auf mehrere Probleme stießen.

```
tool_java_checkStyle > Dockerfile > ...
1 FROM python:3.8-slim
2
3 # Install Checkstyle
4 RUN apt-get update && \
5     apt-get -y --no-install-recommends install git && \
6     apt-get -y --no-install-recommends install curl && \
7     apt-get -y --no-install-recommends install unzip && \
8     apt-get -y --no-install-recommends install default-jdk && \
9     apt-get -y --no-install-recommends install default-jre && \
10    apt-get -y --no-install-recommends install zip && \
11    apt-get -y --no-install-recommends install unzip && \
12    apt-get -y --no-install-recommends install composer
13
14 RUN curl -L -o $(INSTALL_DIR)/checkstyle.jar https://github.com/checkstyle/checkstyle/releases/download/checkstyle-$(CHECKSTYLE_VERSION)/checkstyle-$(CHECKSTYLE_VERSION)-all.jar
15
16 RUN pip install --upgrade pip && \
17     pip install falcon==2.0.0 && \
18     pip install jsonschema==3.2.0 && \
19     pip install gunicorn==20.0.4
20
21 ENV PYTHONPATH "${PYTHONPATH}:/var/toolserver"
22 ENV PYTHONPATH "${PYTHONPATH}:/var/cocoa/tool"
23
24 WORKDIR /var/toolserver
25
26 CMD ["sh", "-c", "gunicorn ${RELOAD} --timeout ${TOOL_SERVER_TIMEOUT} -b :${CHECKSTYLE_PORT} tool_server:app"]
```

Als wir versuchten, das Image aus dem Dockerfile zu erstellen, stießen wir auf die folgende Fehlermeldung:

```
fatema@DESKTOP-9PVB2HB:~/espresso-it/tool_java_checkStyle$ docker build .
Sending build context to Docker daemon 6.144kB
Error response from daemon: dial unix /mnt/wsl/docker-desktop/shared-sockets/guest-services/docker.sock: connect: connection refused
```

Das Problem war, dass das Image ziemlich groß ist und die einzige Lösung dafür war, andere Docker-Images zu löschen, um den Fehler zu beheben.

Das nächste Problem betrifft den Moodle-Container, der immer wieder herunterfährt, obwohl er mit "docker compose up -d" gestartet wurde.

```
moodle 20:51:16.18
moodle 20:51:16.18 Welcome to the Bitnami moodle container
moodle 20:51:16.18 Subscribe to project updates by watching https://github.com/bitnami/containers
moodle 20:51:16.19 Submit issues and feature requests at https://github.com/bitnami/containers/issues
moodle 20:51:16.19
moodle 20:51:16.19 INFO ==> ** Starting Moodle setup **
realpath: /bitnami/apache/conf: No such file or directory
moodle 20:51:16.22 INFO ==> Configuring Apache ServerTokens directive
moodle 20:51:16.26 INFO ==> Configuring PHP options
moodle 20:51:16.26 INFO ==> Setting PHP expose_php option
moodle 20:51:16.29 INFO ==> Validating settings in MYSQL_CLIENT * env vars
moodle 20:51:16.30 INFO ==> Validating settings in POSTGRES_CLIENT * env vars
moodle 20:51:16.34 WARN ==> You set the environment variable ALLOW_EMPTY_PASSWORD=yes. For safety reasons, do not use this flag in a production environment.
moodle 20:51:16.44 INFO ==> Restoring persisted Moodle installation
moodle 20:51:27.52 INFO ==> Trying to connect to the database server
```