



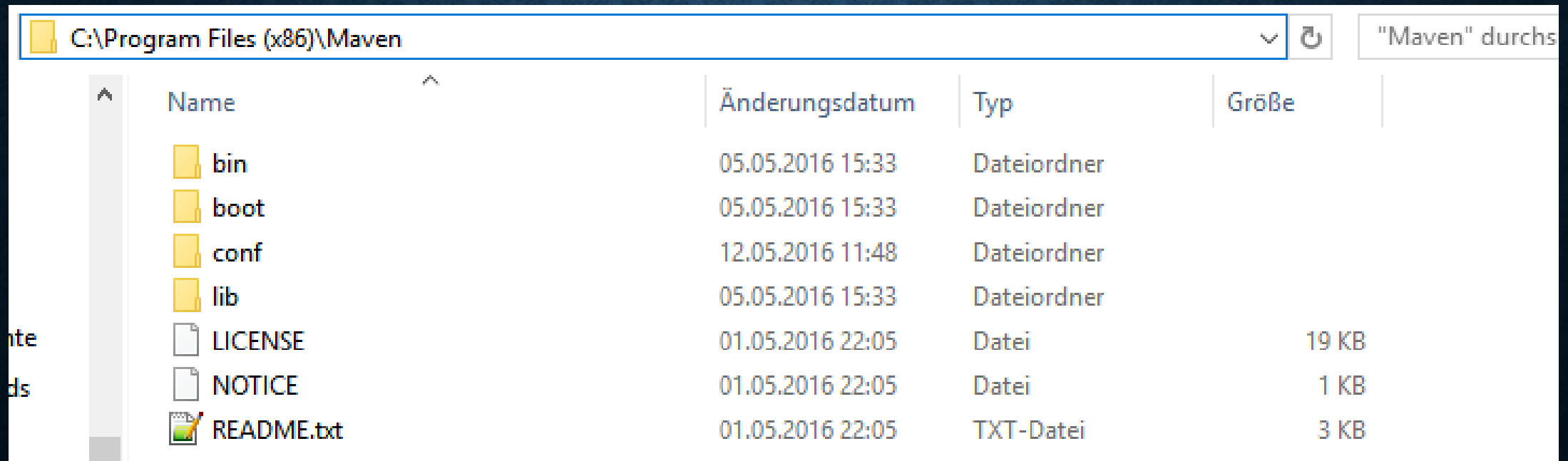
BUILD-MANAGEMENT

Einführung



INSTALLATION (I)

- Download von Maven: <https://maven.apache.org/download.cgi> (*-bin.zip unter Win)
- Download entpacken und Inhalt speichern (z.B. unter C:\Program Files (x86)\Maven)



INSTALLATION (II)

- Neue Systemvariablen hinzufügen (keine Benutzervariablen!):
 - Windows: Systemsteuerung → System und Sicherheit → System → Erweiterte Systemeinstellungen → Umgebungsvariablen → Systemvariablen
 - Variable *M2_HOME* anlegen, als Wert den Pfad zum Maven-Hauptverzeichnis eintragen (z.B. C:\Program Files (x86)\Maven)
 - Variable *M2* anlegen, als Wert *%M2_HOME%\bin* eintragen
- Systemvariable *Path* suchen, markieren, „Bearbeiten“ auswählen und *%M2%* der Liste hinzufügen

INSTALLATION (III)

Systemvariablen

| Variable | Wert |
|----------------------|-----------------------------------|
| LEJOS_EV3_JAVA_HOME | C:\Program Files\Java\jdk1.8.0_92 |
| M2 | %M2_HOME%\bin |
| M2_HOME | C:\Program Files (x86)\Maven |
| MAVEN_OPTS | -Xms256m -Xmx512m |
| MKL_SERIAL | YES |
| NUMBER_OF_PROCESSORS | 4 |
| OS | Windows NT |

Neu... Bearbeiten... Löschen

Umgebungsvariable bearbeiten

C:\ProgramData\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
C:\Program Files (x86)\Skype\Phone\
C:\Program Files\Git\cmd
C:\Program Files (x86)\nodejs\
C:\TexLive\bin\win32
C:\Program Files\Java\jdk1.8.0_92\bin
%M2%
C:\Program Files (x86)\IVI Foundation\VISA\WinNT\Bin
C:\Program Files (x86)\Lejos\bin
C:\Program Files (x86)\Haskell\mingw\bin
C:\Program Files (x86)\Windows Live\Shared
C:\Program Files (x86)\TortoiseGit\bin
%USERPROFILE%\dnx\bin
C:\Program Files\Microsoft DNX\Dnvm\
C:\Program Files\Microsoft SQL Server\130\Tools\Binn\
C:\Program Files (x86)\Windows Kits\8.1\Windows Performance To...

Neu
Bearbeiten
Durchsuchen...
Löschen
Nach oben
Nach unten
Text bearbeiten...

OK Abbrechen

INSTALLATION (IV)

- Hat alles geklappt, resultiert ein Aufruf von *mvn -v* auf der Konsole („Eingabeaufforderung“) nicht in einer Fehlermeldung

```
C:\> Eingabeaufforderung

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Jan>mvn -v
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-10T17:41:47+01:00)
Maven home: C:\Program Files (x86)\Maven
Java version: 1.8.0_92, vendor: Oracle Corporation
Java home: C:\PROGRA~1\Java\JDK18~1.0_9\jre
Default locale: de_DE, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "dos"

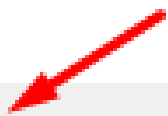
C:\Users\Jan>
```



INSTALLATION (V)

- Troubleshooting:
 - In den Systemvariablen muss die Variable *JAVA_HOME* gesetzt sein und auf den Ordner zeigen, in dem das auf dem PC installierte JDK liegt
 - Detailliertere Installationsanleitung (auch für andere Betriebssysteme):
https://www.tutorialspoint.com/maven/maven_environment_setup.htm
Immer Systemvariablen manipulieren, nicht die Benutzervariablen!

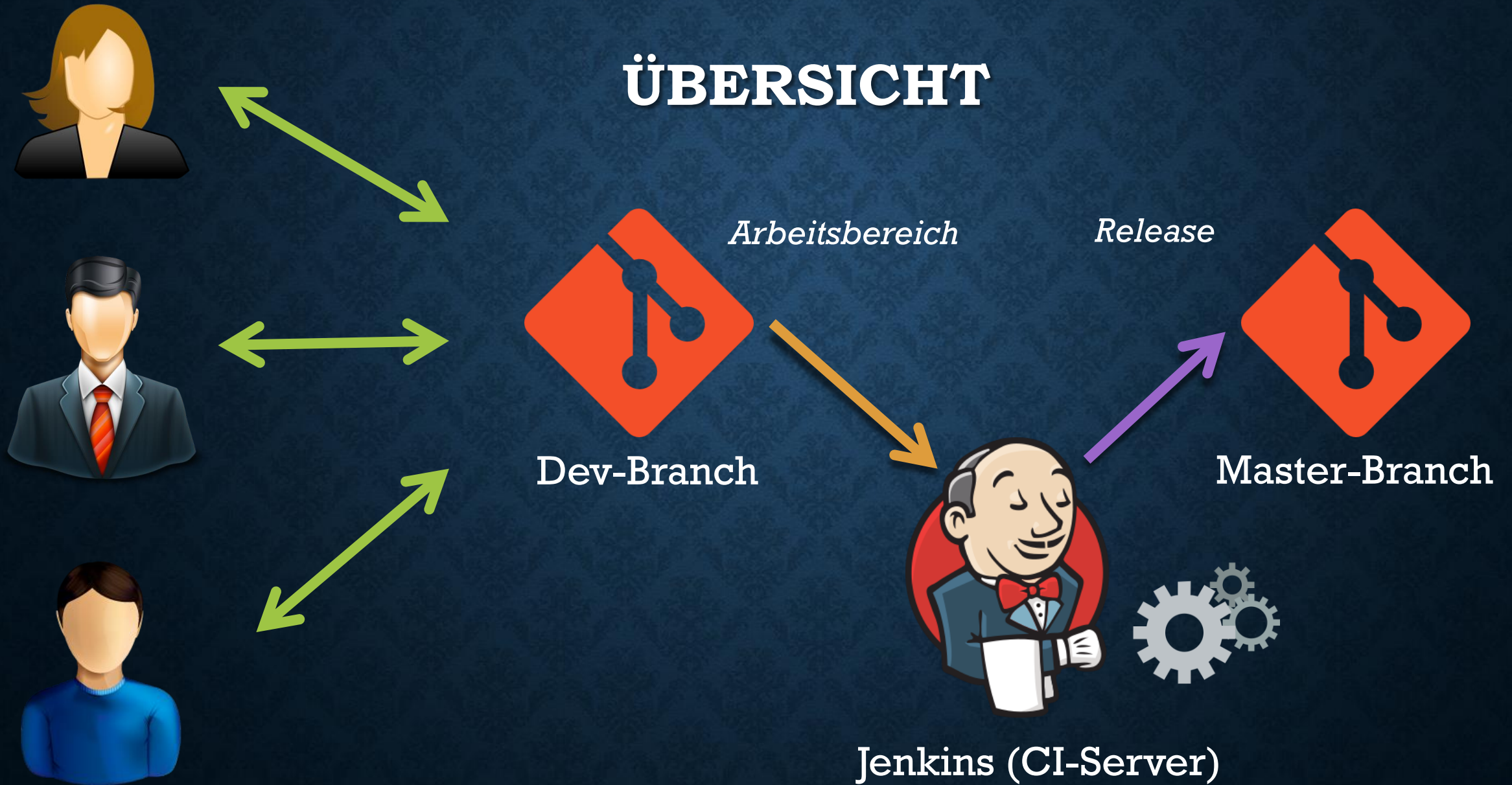
| Systemvariablen | |
|-----------------|------------------------------|
| Variable | Wert |
| ComSpec | C:\WINDOWS\system32\cmd.exe |
| EV3_HOME | C:\Program Files (x86)\Lejos |
| JAVA_HOME | C:\PROGRA~1\Java\JDK18~1.0_9 |



PROBLEMATIK

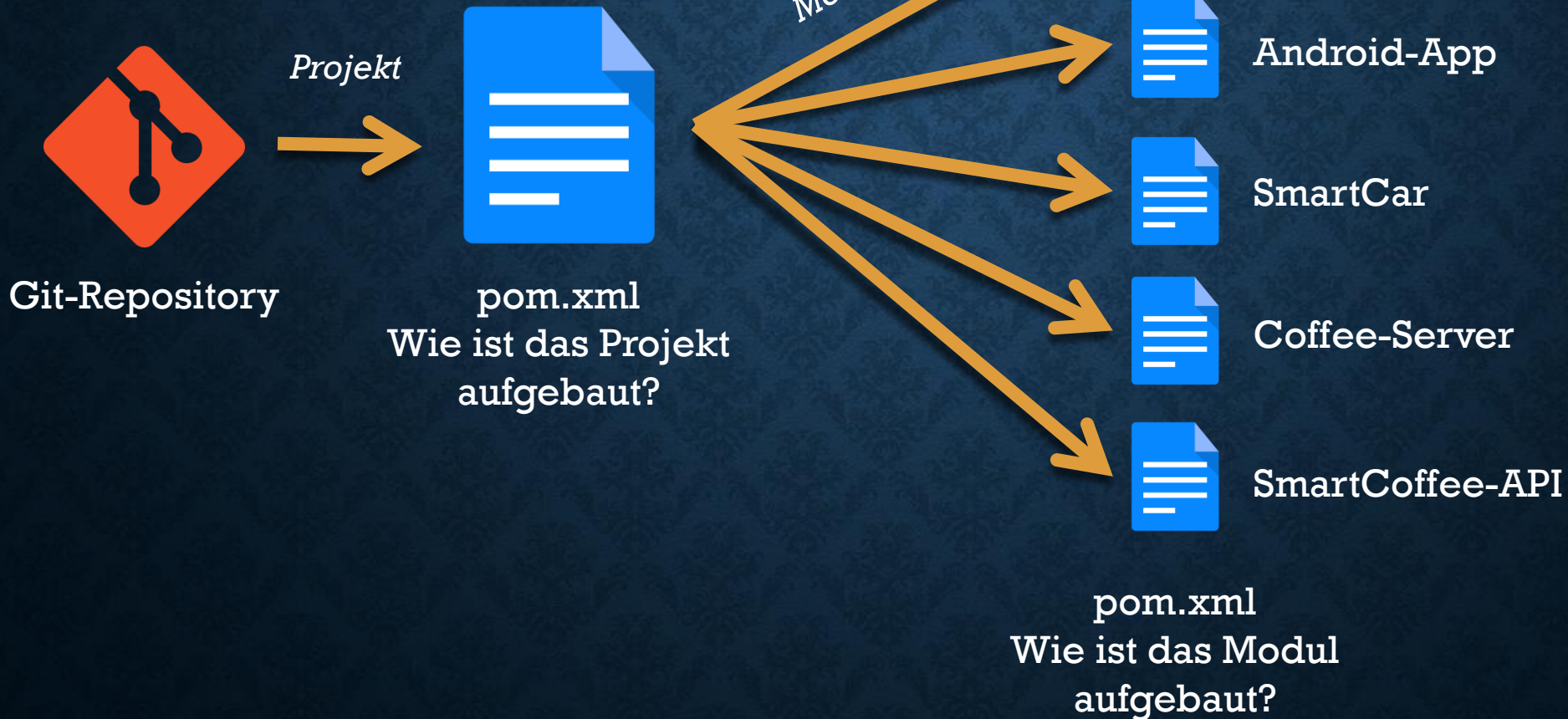
- Viele „lose“ Softwarekomponenten, die alle irgendwie zusammenhängen
- Kein richtiges Release, alles liegt irgendwo verteilt im Git-Repository
- Instabile Software, nicht direkt vorzeigbar/ausführbar bei Reviews
- Hardware, die zu verschiedenen Softwarekomponenten gehört
- Kein einheitlicher Code, wenig dokumentiert

ÜBERSICHT



Beschreibung des Aufbaus des
Projekt und den einzelnen
Abhängigkeiten durch Project-
Object-Model-Dateien

MAVEN (I)



MAVEN (II)

Die XML-Datei beschreibt, welche Plugins auf dem jeweiligen Modul ausgeführt werden sollen und welche anderen Module/Artefakte es benötigt, um kompiliert werden zu können



pom.xml

Lokales Maven-Repository



Abhängigkeit

Auto. Download

MAVEN (III)

- Maven-Befehle müssen im lokalen Repository (Hauptverzeichnis SmartHomeMain oder im Verzeichnis eines Submoduls) über die Konsole ausgeführt werden (Navigation in der Konsole über `cd ..` und `cd Ordner`)
- *mvn install*: Lädt die benötigten Plugins herunter, installiert sie in das lokale Maven-Repository, wendet sie an, führt Tests aus und kompiliert das Modul. Das Ergebnis dieses Vorgangs ist in den jeweiligen Modulen in einem *target*-Ordner gespeichert; zusätzlich wird bei Ausführung im Hauptverzeichnis ein *build*-Ordner angelegt, in den die Dokumentation und alle kompilierten Artefakte zusammengeführt werden
- *mvn clean*: Räumt die Module wieder auf und entfernt den *target*-Ordner
- *mvn clean install*: Kombination; erst wird aufgeräumt, dann installiert

PLUGINS (1)

- Android-App:
 - Checkstyle
 - Bezieht KAA-SDK automatisch aus dem Ordner Middleware/KAA-SDKs
 - Ausführung des Gradle-Builds aus Android-Studio → Führt Tests aus, erzeugt APK
- SmartCar:
 - Checkstyle, FindBugs
 - Bezieht KAA-SDK automatisch aus dem Ordner Middleware/KAA-SDKs
 - Testausführung, Kompilierung → Erzeugt JAR-Datei
 - Deploy auf den Mindstorm über SCP mit *mvn install antrun:run*
- SmartCoffee-API, CoffeeServer
 - Checkstyle, FindBugs
 - Abhängigkeit zwischen API und CoffeeServer
 - Bezieht KAA-SDK automatisch aus dem Ordner Middleware/KAA-SDKs
 - Testausführung, Kompilierung → Erzeugt JAR-Datei



PLUGINS (II)



- Checkstyle:
 - Wird während *mvn install* ausgeführt
 - Prüft Code hinsichtlich Code-Konventionen
 - Regeln sind als XML-Datei „checkstyle.xml“ im Hauptverzeichnis hinterlegt
 - Zeigt Warnungen (nicht regel-konformen Code-Style) auf der Konsole an

```
[INFO]
[INFO] --- maven-checkstyle-plugin:2.17:check (validate) @ android_app ---
[INFO] Starting audit...
C:\Softwaretechnik\Studienprojekt\SmartHomeMain\ControlThing\SmartHomeApp_android\app\src\main\gen\com\smartHome\SmartHomeApp\BuildConf
g.java:7:16: warning: Modifier 'static' weicht von der empfohlenen Modifier-Reihenfolge aus der Java-Sprachdefinition ab.
C:\Softwaretechnik\Studienprojekt\SmartHomeMain\ControlThing\SmartHomeApp_android\app\src\main\java\com\smartHome\SmartHomeApp\gui\About
Activity.java:15: warning: Import-Anweisung f  r 'com.smartHome.SmartHomeApp.R.id.buttonKaffeeeingie  en' ist in der falschen Reihenfolg
e. Sollte in der ' sein, STATIC' Gruppe, in der Erwartung nicht belegt Einfuhren auf dieser Linie.
C:\Softwaretechnik\Studienprojekt\SmartHomeMain\ControlThing\SmartHomeApp_android\app\src\main\java\com\smartHome\SmartHomeApp\gui\About
Activity.java:33:46: warning: Parameter name 'v' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*$'.
C:\Softwaretechnik\Studienprojekt\SmartHomeMain\ControlThing\SmartHomeApp_android\app\src\main\java\com\smartHome\SmartHomeApp\gui\About
Activity.java:34:31: warning: Local variable name 'i' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*$'.
C:\Softwaretechnik\Studienprojekt\SmartHomeMain\ControlThing\SmartHomeApp_android\app\src\main\java\com\smartHome\SmartHomeApp\gui\About
Activity.java:41:46: warning: Parameter name 'v' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*$'
```

PLUGINS (III)

- FindBugs:
 - Wird während *mvn install* ausgeführt
 - Untersucht den Code hinsichtlich typischer Fehlermuster
 - Zeigt Warnungen (potentielle Fehler) auf der Konsole an
 - Befehl „*mvn findbugs:gui*“ zeigt die Probleme auf einer grafischen Oberfläche an (funktioniert nur auf Modul-Ebene)



```
INFO] Unread field: KaaVerbindung.USER_ACCESS_TOKEN; should this field be static? [KaaVerbindung] At KaaVerbindung.java:[line 27]
INFO] Unread field: KaaVerbindung.USER_EXTERNAL_ID; should this field be static? [KaaVerbindung] At KaaVerbindung.java:[line 26]
INFO] LinieFolgen.blueLineParking() invokes System.exit(...), which shuts down the entire virtual machine [LinieFolgen] At LinieFolgen.j
ava:[line 212]
INFO] LinieFolgen.checkBlackline() invokes System.exit(...), which shuts down the entire virtual machine [LinieFolgen] At LinieFolgen.j
va:[line 634]
INFO] There is an apparent infinite loop in LinieFolgen.preventCollision() [LinieFolgen, LinieFolgen] At LinieFolgen.java:[line 237]Lo
o
r bottom at LinieFolgen.java:[line 374]
INFO] Check for oddness that won't work for negative numbers in LinieFolgen.followBlackLine() [LinieFolgen] At LinieFolgen.java:[line 6
']
INFO] Check for oddness that won't work for negative numbers in LinieFolgen.preventCollision() [LinieFolgen, LinieFolgen] At LinieFolge
i.java:[line 243]Another occurrence at LinieFolgen.java:[line 276]
INFO] Write to static field LinieFolgen.currentValue from instance method LinieFolgen.preventCollision() [LinieFolgen] At LinieFolgen.j
va:[line 223]
```


PLUGINS (IV)

FindBugs - Smart-Car

Datei Bearbeiten Ansicht Navigation Bewertung Hilfe

Class name filter: Filter

Group bugs by:

Fehler (10)

- Correctness (1)
 - Infinite Loop (1)
 - There is an apparent infinite loop in preventCollision()
- Bad practice (2)
 - Dubious method used (2)
 - blueLineParking() invokes System.exit(...), which shuts down the e
 - checkBlackline() invokes System.exit(...), which shuts down the e

No cloud selected

Klicken um Review zu erstellen...

unclassified

Saved Cancel

LinieFolgen.java in

```
228 // When a object is blocks the way, the minds
229 if (distanceValue <= 0.1) {
230 kaa.dodgingObjectEvent();
231 System.out.println("distanceValue");
232 // turns the mindstorm left for 90 degree
233
234 evadesObject();
235 boolean test = true;
236
237 while (test != false) {
238 counter++;
239 double delayZaehler = 0;
240
241 distanceValue = distance.getDistance(
242 System.out.println("counter++");
243 if (distanceValue <= 0.2 && counter %
244 while (test == true) {
245 delayZaehler++;
```

Suche Vorwärts suchen Rückwärts suchen

There is an apparent infinite loop
At LinieFolgen.java:[line 237]
In method LinieFolgen.preventCollision() [Zeilen 222 - 378]
Loop bottom at LinieFolgen.java:[line 374]

An apparent infinite loop
This loop doesn't seem to have a way to terminate (other than by perhaps throwing an exception).

Bug kind and pattern: IL - IL_INFINITE_LOOP

<http://findbugs.sourceforge.net>

UNIVERSITY OF MARYLAND

