

# Spezifikation Entwicklungsumgebung

## IDE / Editor

Kriterien	Eclipse	IntelliJ IDEA	NetBeans
Lizenz	Eclipse Public License	kostenpflichtig	Apache License 2.0
Erfahrungswerte	gut	-	-
vorhandene Fähigkeiten	ja	nein	nein

**Auswahl:** Eclipse

**Begründung:** Gute Erfahrungen sind ebenso vorhanden wie der erfahrene Umgang der Mitarbeiter mit dieser IDE. Weiterhin ist Eclipse sehr umfangreich und bietet gute Schnittstellen zur Erweiterung des Funktionsumfangs.

## Versionierung

Kriterien	Git	hg/Mercurial	svn/Subversion
Funktionsumfang	groß	mittel	mittel
Verbreitung	sehr hoch	hoch	mittel
Organisation	dezentral	dezentral	zentral
Funktionsumfang	GPLv2.0	GPLv2+	Apache License 2.0
Erfahrungswerte	gut	-	-
vorhandene Fähigkeiten	ja	nein	nein

**Auswahl:** Git

**Begründung:** Git ist weit verbreitet, besitzt einen großen Funktionsumfang und es bestehen gute Erfahrungswerte mit dieser Versionierungssoftware.

## Build Tool

Kriterien	Ant	Maven	Gradle
Dokumentation	gut	gut	gut, mit Online-Tutorial
Lizenz	Apache License 2.0	Apache License 2.0	Apache License 2.0
Aktualität	fortgeführt, aber alt	aktuell	aktuell

**Auswahl:** Gradle

**Begründung:** Gradle ist sehr gut dokumentiert und aktuell.

### UML Tool

Kriterien	Eclipse Top-cased	StarUML	Papyrus	draw.io
Aktualität	nicht fortgeführt	aktuell	aktuell	aktuell
Code-Generierung	ja	ja	ja	nein
Lizenz	Eclipse Public License	frei, kommerziell kostenpflichtig	Eclipse Public License	kostenlos
Portabilität	Java: Mac/Win/Linux	Mac/Win/Linux	Java: Mac/Win/Linux	ja, Web-App

**Auswahl:** draw.io

**Begründung:** Die Aktualität sowie die Portabilität durch die Webfunktionalität sind gewährleistet. Es ist keine Codegenerierung erforderlich.

### Tool für Prototyping

Kriterien	Gomocking-bird	Ninjamock	Pencil	Papier etc.
Einschätzung	Webdesign	skizzenhaft, aber digital	simpel, geometrisch	freihand, dynamisch
Plattform	Web-App	Web-App	Mac/Win/Linux	analog
Lizenz	Anmeldung benötigt	kostenpflichtig	Open Source	kostenlos

**Auswahl:** Pencil, Papier

**Begründung:** Die Software besitzt eine geeignete Lizenz, arbeitet lokal und unterstützt mehrere Plattformen.

### Obfuscator

Kriterien	ProGuard	JavaGuard	yGuard
Lizenz	GPLv2	LGPL	LGPL
Dokumentation	ausführlich	wenig	wenig

**Auswahl:** ProGuard

**Begründung:** ProGuard besitzt eine gute Dokumentation.

## Test-Automatisierung

Kriterien	JUnit	TestNG
Erfahrungswerte	gut	-
vorhandene Fähigkeiten	ja	nein

**Auswahl:** JUnit

**Begründung:** Es existieren gute Erfahrungswerte und Fähigkeiten mit diesem Automatisierungswerkzeug. Außerdem lässt es sich in der gewählten IDE integrieren.

## Codeconvention

Kriterien	Google [1]	Mozilla [2]	Oracle [3]
Kommentare	alle	einfache, keine Dokumentationskommentare	alle
Zeichensatz	UTF-8	UTF-8	UTF-8
Tabulatoren	nur Leerzeichen	nur Leerzeichen, Tab = 4 Leerzeichen	nur Leerzeichen
Bezeichnungen	CamelCode	CamelCode	CamelCode
Zeichen pro Zeile	80	100	80
Deklarationen pro Zeile	max. 1	max. 1	max. 1
Statements pro Zeile	max. 1	-	max. 1
Sonstiges	-	-	Code in Sections gegliedert, pro Datei ein einleitender Kommentar mit Infos

[1] <https://google.github.io/styleguide/javaguide.html>

[2] [https://developer.mozilla.org/en-US/docs/Mozilla/Developer\\_guide/Coding\\_Style#Java\\_practices](https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide/Coding_Style#Java_practices)

[3] <https://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

**Auswahl:** Google

**Begründung:** Google's Codeconvention für Java bietet eine ausreichende Strukturierung des Codes, ohne zu restriktiv zu sein.

### Dokumentationstool

Kriterien	Javadoc	Doxygen
Integration in IDE	gut	nicht vorhanden

**Auswahl:** Javadoc

**Begründung:** Javadoc ist im Gegensatz zu Doxygen mit der gewählten IDE kompatibel beziehungsweise lässt sich gut in diese integrieren.

### Kollaborationstool

Kriterien	Mailingliste	persönliches Treffen	Issue Tracker	Owncloud
Eigenschaften	verteilte Kommunikation	individuell, dynamisch	zentrale Organisation von Bugs	zentrale Datenspeicherung, keine Parallelität

**Auswahl:** persönliches Treffen, Issue Tracker

**Begründung:** In persönlichen Treffen ist der Informationsaustausch wesentlich intensiver als bei vergleichbaren Kommunikationswegen. Über Issue Tracker der Bug-Status des Projektes überwacht werden.