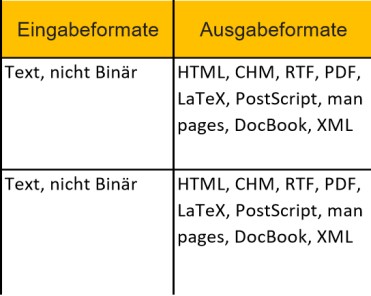
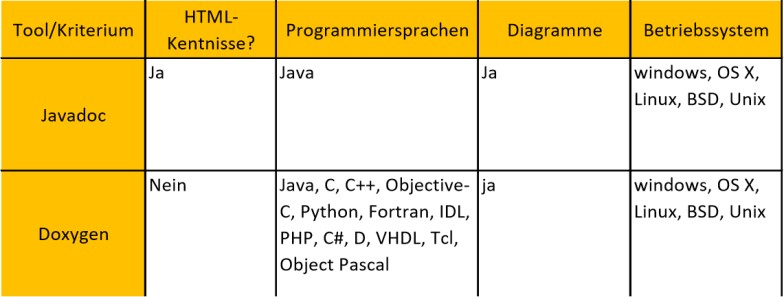
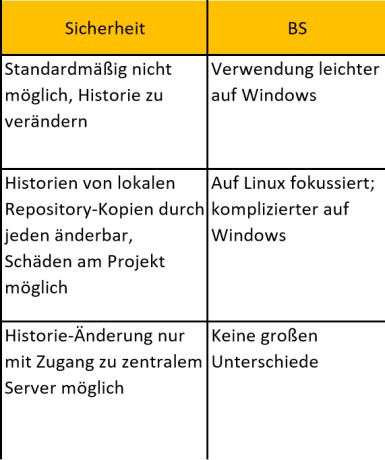
# Dokumentationstools



Entscheidung: Wir haben uns als Dokumentationstool für Javadoc entschieden, da es Industriestandard zur Dokumentation von Java-Klassen und in JDK enthalten ist.

Javadoc kann außerdem auch eine Vielzahl von Diagrammen erstellen und bietet eine strukturierte Ansicht des Quellcodes.

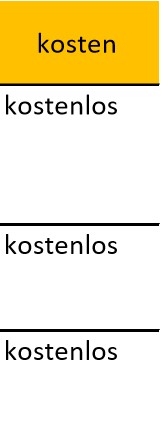
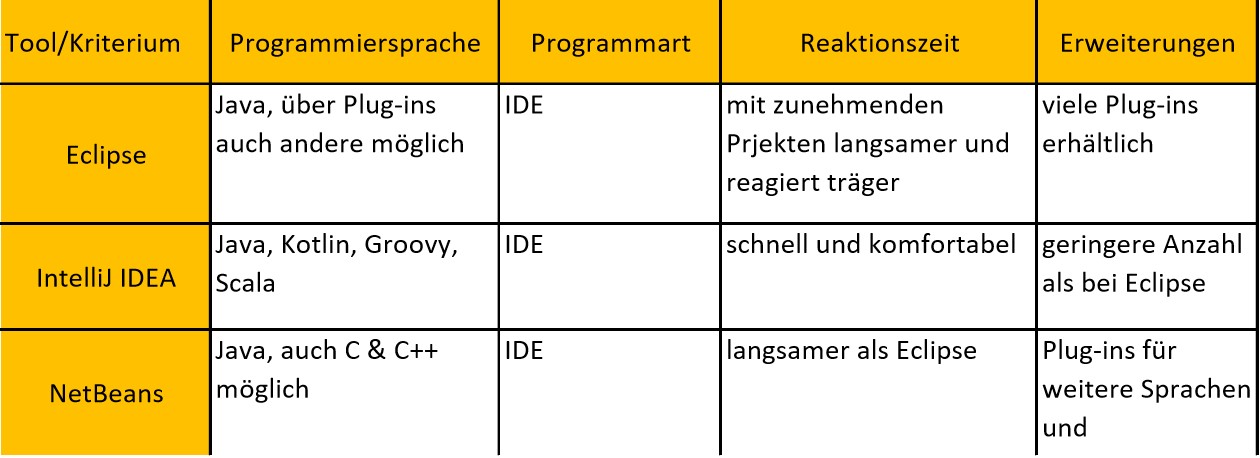
# Versionierung



Entscheidung: Unser Versionierungstool soll für dieses Projekt Git sein. Der Hauptgrund hierbei besteht darin, dass Git ein sehr bekanntes und weit verbreitetes Versionierungswerkzeug ist.

Es erwies sich in vergangenen Projekten, dass es sich durch das einfache „mergen“ von „Branches“ hervorragend zum Zusammenarbeiten und Strukturieren unsere Projekte eignet.

# IDE



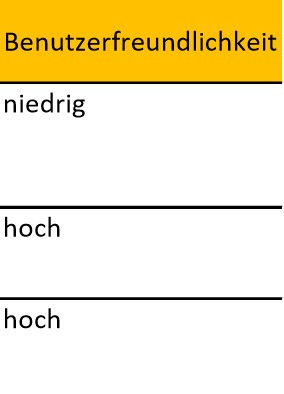
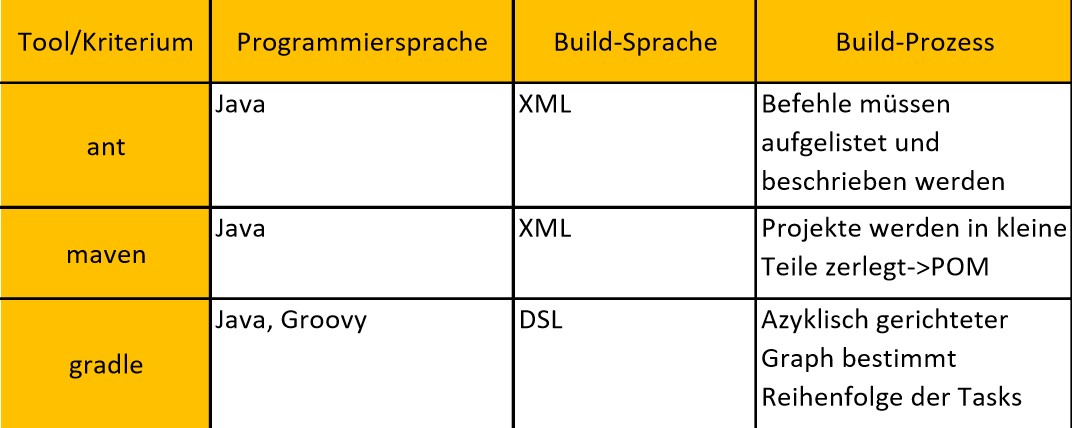
Entscheidung: Als IDE werden wir Eclipse benutzen, da wir diese Entwicklungsumgebung im Rahmen vorheriger Projekte ins Herz geschlossen haben. Es ist kostenlos und eignet sich hervorragend zur Entwicklung in der Programmiersprache JAVA.

# Codeconventions



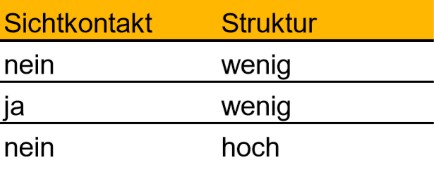
Entscheidung: Wir möchten für unseren Quellcode die Code-Conventions von Google benutzen. Es ist ein großer anerkannter Standard für die Gestaltung von Code und erfüllt seine Rolle in unserem Projekt hinsichtlich der einheitlichen Strukturierung des Quellcodes.

# Build-Tool



Entscheidung: Wir werden Maven benutzen, da es XML als Build-Sprache benutzt und gut strukturiert ist. Das Programm ist unserer Meinung außerdem recht benutzerfreundlich.

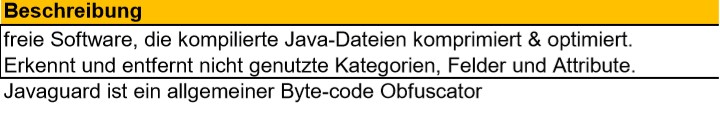
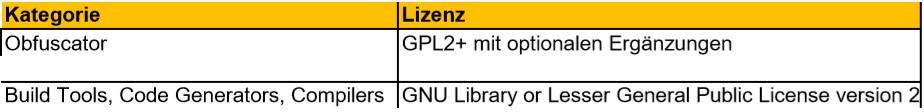
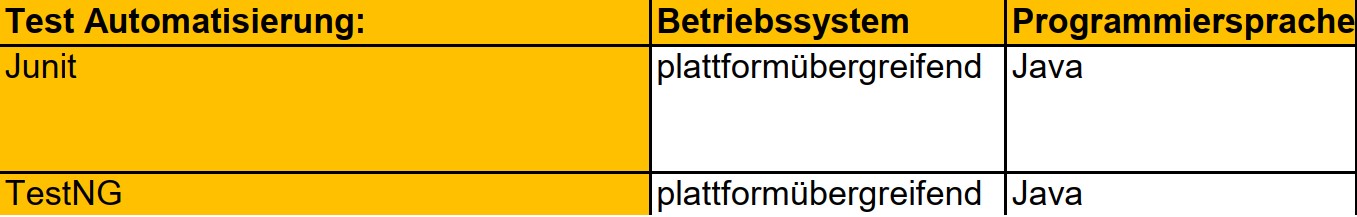
# Kolaborationstools



Entscheidung: Wir benutzen "Treffen im Praktikum" um miteinander zu kolaborieren. Es bietet sich für uns als IT-Dienstleister am besten an, da wir im Rahmen unserer vorherigen Projekte gute Erfahrung damit gemacht haben. Mit der Namensgebung machen wir eine Anspielung auf die Zeit als wir noch studierten, da unsere Projekte in der Studienzeit damit immer gut funktioniert haben. Es ist außerdem hoch effizient.

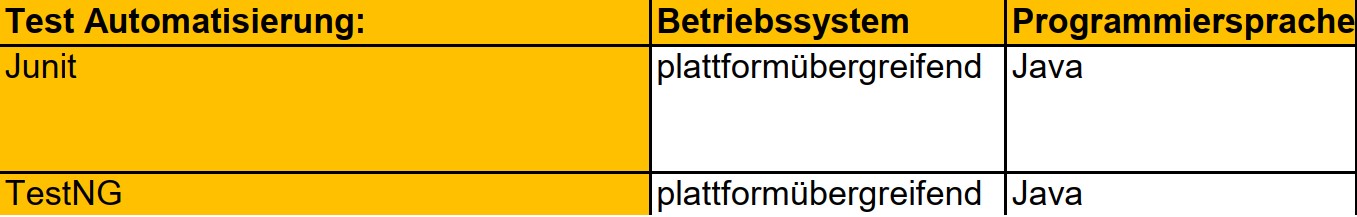
# Obfuscator

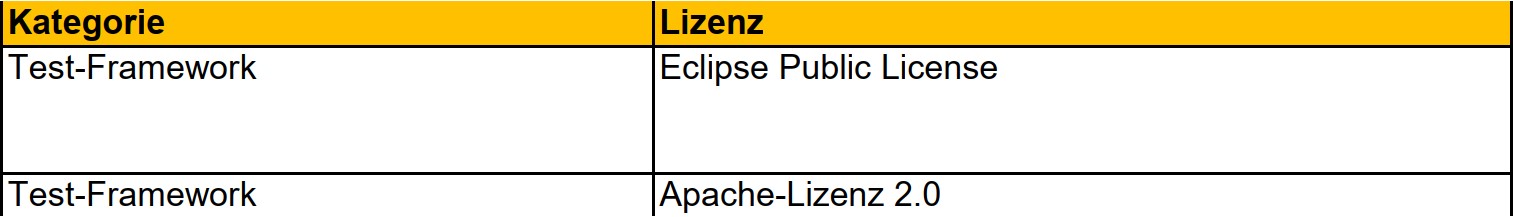


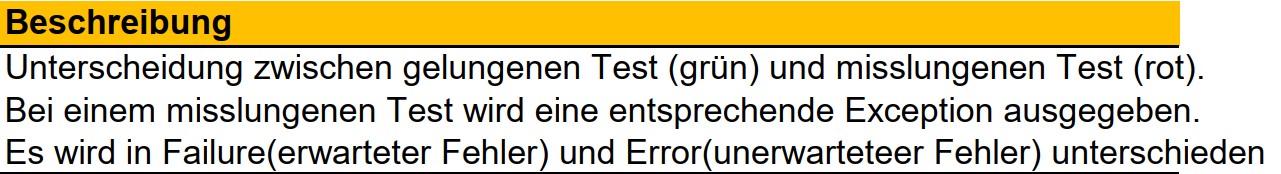


Wahl: Proguard: Aufgrund der Tatsache, dass sich Javaguard aktuell noch in der Beta-Version befindet, ist damit zu rechnen, dass viele Erneuerungen auf das Tool drauf zu kommen. Insofern ist ProGuard die weit aus bessere Wahl.

# Test-Automatisierung

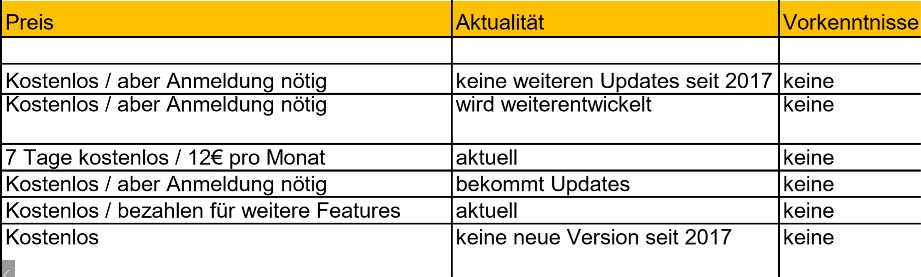






Wahl: Junit: Aufgrund guter Arbeit des gesamten Teams mit der Entwicklungsumgebung Eclipse, wird als Test-Automatisierungstool Junit gewählt um somit Kosten beim Lizenzkauf zu sparen.

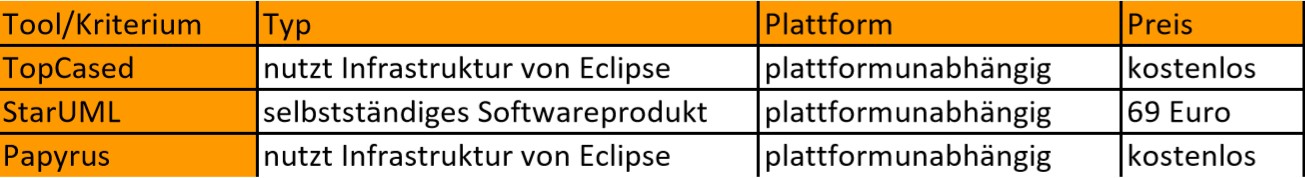
# Prototyping der Benutzerschnittstelle



Entscheidung : Unsere Entscheidung fällt nicht auf Grund von Vorerfahrungen, da wir in diesem Bereich noch keine haben.

Wir entscheiden uns für NinjaMock, weil es kostenlos und einfach zu benutzen ist.

# Uml-Tools





Entscheidung: Wir möchten TopCased als UML Tool verwenden, da wir bereits im Praktikum Erfahrungen damit sammeln konnten. Es ist in unseren Büros bereits installiert und steht uns kostenlos zur Verfügung.