

# Entwicklungsumgebung: Toolvergleich

*Projekt „Entwicklung eines Softwaresystems für einen Bankautomaten“  
für das Shoppinghaus Dresden*



## Inhaltsverzeichnis

1 Versionierung.....	3
2 UML-Tool mit Quellcodegenerierung.....	5
3 Build-Tool.....	6
4 Prototyping der Benutzerschnittstelle.....	8
5 IDE/Editor.....	11
6 Testautomatisierung.....	13
7 Dokumentationstool.....	15
8 Obfuscator.....	17
9 Code Conventions.....	19
10 Kollaborationstool.....	20
11 Quellen.....	22

# 1 Versionierung

Versionierung ist ein systematischer Ansatz zum Management aller Änderungen an einem Produkt oder System.

Lösung	Besonderheiten	Zugänglichkeit und Bedienbarkeit	Sicherheit	Fazit: Eignung für den verfolgten Zweck
Git	<ul style="list-style-type: none"> <li>• paralleles Arbeiten<sup>1</sup></li> <li>• Flexibilität</li> <li>• einfache Beiträge</li> <li>• Performance</li> <li>• Robustheit<sup>2</sup></li> <li>• Offsite und Offline</li> <li>• Erweiterbarkeit und Integrationsfähigkeit</li> <li>• Gemeinschaft und Freiheit</li> </ul>	<ul style="list-style-type: none"> <li>• Open Source und cross-platform</li> <li>• kostenlos</li> <li>• die Arbeit ohne Zugriff auf den Server</li> <li>• Git auf einem Mac, Windows und Linux installierbar</li> </ul>	<ul style="list-style-type: none"> <li>• Repository: lokal vorliegende Repository-Kopien, in denen gearbeitet wird</li> <li>• Es ist offen für individuelle Anwendungen und Zusammenarbeit mit nahezu jeder beliebigen anderen Software.</li> <li>• wird auf einer Workstation installiert und fungiert als Client und Server</li> <li>• der Quellcode steht offen bereit</li> </ul>	<ul style="list-style-type: none"> <li>• <b>schwer</b> zu lernen</li> <li>• Mit der Nutzung von Git-Management-Tools können Unternehmen alle Vorteile der Arbeit mit Git nutzen und gleichzeitig die höchsten Anforderungen ihres Unternehmens an Sicherheit, Skalierbarkeit und Normeinhaltung erfüllen.</li> <li>• Sicherheit: sehr gut. Git ist besser für erfahrene Benutzer.</li> </ul>
SVN (Subversion)	<ul style="list-style-type: none"> <li>• zentrales Versionskontrollsystem</li> <li>• Trägheit<sup>3</sup></li> <li>• offline nicht verfügbar</li> </ul>	<ul style="list-style-type: none"> <li>• Open-Source-Tool (Aufwand)</li> <li>• kostenlos</li> <li>• verfügbar auf Unix, Windows, macOS</li> </ul>	<ul style="list-style-type: none"> <li>• Repository: ein zentrales Repository, in dem Arbeitskopien erzeugt werden</li> <li>• Entwickler können ihre Änderungen direkt an dieses zentrale Server-Repository übertragen</li> <li>• hat einen separaten Server und Client</li> <li>• Wenn bei SVN das zentrale Repository ausfällt oder ein Code den Build beschädigt, können andere Entwickler ihren Code nicht übertragen, bis das Repository repariert ist.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>leicht</b> zu lernen</li> <li>• Subversion erleichtert einerseits die Zusammenarbeit im Team und erstellt andererseits auch ein automatisches Backup unserer Programme.</li> <li>• Sicherheit: sehr gut</li> </ul>

[1] *Paralleles Arbeiten: Git erlaubt es mehreren Entwicklern parallel und dezentral an einem Projekt (Repository) in verschiedenen Entwicklungszweigen (Branches) zu arbeiten (verteiltes Versionskontrollsystem).*

[2] *Robustheit: Die Daten gelöschter und zurückgenommener Aktionen und Entwicklungszweige bleiben vorhanden (und können wiederhergestellt werden), bis sie explizit gelöscht werden.*

[3] *Trägheit: Es wird nur eine riesige Codebasis erstellt. Versionskontrollsysteme können schwierig zu wechseln sein.*

Hg (Mercurial)	<ul style="list-style-type: none"> <li>• verteiltes Versionskontrollsystem</li> <li>• Flexibilität</li> <li>• kann Offline verwendet werden</li> </ul>	<ul style="list-style-type: none"> <li>• Open Source</li> <li>• kostenlos</li> <li>• Es wird auf Windows- und Unix-Systemen wie macOS und Linux unterstützt.</li> </ul>	<ul style="list-style-type: none"> <li>• verteiltes Source-Control-Management-Tool (wie Git)</li> <li>• hg init erstellt ein neues Repository</li> <li>• hg commit speichert Änderungen im aktuellen Repository</li> <li>• hg log, um alle Änderungen im Repository anzuzeigen</li> <li>• hg pull bringt alle Änderungen aus einem anderen Repository in das aktuelle Repository</li> <li>• hg push sendet alle Änderungen aus dem aktuellen Repository in ein anderes Repository</li> <li>• hg serve erstellt einen Webserver, auf dem Mitarbeiter den Verlauf sehen und daraus ziehen können</li> </ul>	<ul style="list-style-type: none"> <li>• <b>leicht</b> zu lernen</li> <li>• schnell und kraftvoll</li> <li>• verarbeitet Projekte jeder Größe effizient und bietet eine einfache und intuitive Benutzeroberfläche</li> <li>• Mercurial ist sicherer für weniger erfahrene Benutzer.</li> </ul>
----------------	--	---	---	--

Wir werden als Versionierung *Git* auswählen, denn:

- Jeder Entwickler hat eine vollständige Kopie des Codes auf seinen lokalen Systemen.
- Alle Änderungen am Quellcode können von anderen nachverfolgt werden.
- Es findet regelmäßiger Austausch zwischen den Entwicklern statt.

## 2 UML-Tool mit Quellcodegenerierung

Die Darstellung von Ideen in einem visuellen Format ist eine effektive Möglichkeit, den Menschen verständlich zu machen, was Sie vermitteln möchten. Und UML-Diagramme bieten Ihnen genau die Möglichkeit dazu.

Lösung	Besonderheiten	Zugänglichkeit	Bedienbarkeit	Fazit: Eignung für den verfolgten Zweck
Papyrus	<ul style="list-style-type: none"> <li>• ist das am weitesten verbreitete Werkzeug zur Entwicklung von UML-Profilen</li> <li>• basierend auf Modellierungsstandards: UML, SysML etc.</li> </ul>	<ul style="list-style-type: none"> <li>• erweiterbare Plattform</li> <li>• Eclipse</li> <li>• Open Source</li> <li>• Online</li> <li>• Plattformen: Windows, Linux, Mac</li> </ul>	<ul style="list-style-type: none"> <li>• UML</li> <li>• Flowcharts</li> <li>• Diagrams</li> </ul>	<ul style="list-style-type: none"> <li>• Leistungen gut geeignet für unsere Zwecke</li> </ul>
StarUML	<ul style="list-style-type: none"> <li>• Ziel des StarUML-Projekts ist es, ein Softwaremodellierung</li> <li>• ist ein Programm zur Entwicklung einer schnellen, flexiblen, erweiterbaren, funktionsreichen und frei verfügbaren UML/MDA- Plattform, die auf einer Win32- Plattform läuft</li> <li>• als PDF exportieren</li> </ul>	<ul style="list-style-type: none"> <li>• zur Zeit nicht Open Source</li> <li>• Online</li> <li>• Plattformen: Windows, Linux, Mac</li> </ul>	<ul style="list-style-type: none"> <li>• Flowcharts</li> <li>• Entity Relationship Diagrams</li> <li>• Sequence Diagrams</li> <li>• UML</li> </ul>	<ul style="list-style-type: none"> <li>• Leistungen gut geeignet, beliebt bei Entwicklern</li> </ul>
VS	<ul style="list-style-type: none"> <li>• ein populäres Tool</li> <li>• Benutzer können auch zur wachsenden und lebendigen Community auf GitHub beitragen</li> <li>• kombiniert das Beste aus Web-, nativen und sprachspezifischen Technologien</li> <li>• enthält ein öffentliches Erweiterbarkeitsmodell, mit dem Entwickler Erweiterungen erstellen und verwenden und ihre Erfahrung beim Bearbeiten, Erstellen und Debuggen umfassend anpassen können</li> </ul>	<ul style="list-style-type: none"> <li>• kostenlos</li> <li>• Requirements: <ul style="list-style-type: none"> <li>➢ Install VS Code</li> <li>➢ Install PlantUML Extension</li> </ul> </li> <li>• Open Source</li> </ul>	<ul style="list-style-type: none"> <li>• UML</li> <li>• als Desktop-Anwendung auf Linux, Windows, Mac bedienbar</li> <li>• etwas schwerer zu bedienen</li> </ul>	<ul style="list-style-type: none"> <li>• Leistungen wären für unsere Zwecke ausreichend, jedoch ist Einstiegsaufwand in dieses Programm etwas höher</li> </ul>

Wir werden als UML-Tool *Papyrus* auswählen. Da wir als IDE Eclipse nutzen werden, erweist sich Papyrus als UML-Tool am praktischsten, da dieses hier bereits mit integriert ist und wir schon etwas erfahren damit sind.

### 3 Build-Tool

Ein Build-Tool ist ein Programm, das den Prozess des Kompilierens und Testens von Projekten automatisiert und somit das Erstellen einer lauffähigen Software aus Quellcode erleichtert.

Lösung	Besonderheiten	Zugänglichkeit und Bedienbarkeit	Fazit: Eignung für den verfolgten Zweck
Ant	<ul style="list-style-type: none"><li>• Es ist ein Open-Source-Projekt.</li><li>• Es ist plattformübergreifend, da es von JVM abhängt.</li><li>• Angebote und umfangreiche Auswahl an vordefinierten Aufgaben</li><li>• Ant ermöglicht die Bereitstellung der Binärdateien zum Testen des Servers</li><li>• bietet eine erweiterbare Architektur</li><li>• bietet Abwärtskompatibilität</li></ul>	<ul style="list-style-type: none"><li>• verwendet, um eine Anzahl von Dateien in relativ kurzer Zeit zu kompilieren</li><li>• bietet eine Option, die Datei nur bei Bedarf neu zu kompilieren</li><li>• Ant ist prozedural, erfordert also Informationen durch Code</li><li>• Ant hat keinen Lebenszyklus</li><li>• Ant hat keine formalen Konventionen, daher müssen wir alles in der Datei <i>build.xml</i> bereitstellen</li><li>• es ist eine Werkzeugkiste</li><li>• Ant-Skripte sind nicht verwendbar</li></ul>	<ul style="list-style-type: none"><li>• Ant wird hauptsächlich zum Erstellen und Bereitstellen von Java-Projekten verwendet, kann aber für alle möglichen sich wiederholenden Aufgaben verwendet werden, z.B. zum Dokumentation erstellen.</li></ul>
Maven	<ul style="list-style-type: none"><li>• Es bietet Abhängigkeitsverwaltung einschließlich automatischer Aktualisierung.</li><li>• Modellbasierter Build: Maven ist in der Lage, eine beliebige Anzahl von Projekten in vordefinierten Ausgabetypen wie war, JAR usw. zu erstellen.</li><li>• Kohärente Seite mit Projektinformationen: kann eine vollständige Dokumentation erstellen</li><li>• bessere Fehler- und Integritätsberichte</li><li>• Maven bietet parallele Builds</li></ul>	<ul style="list-style-type: none"><li>• Maven ist deklarativ, Sie müssen alles in der Datei <i>pom.xml</i> definieren.</li><li>• Maven hat einen Lebenszyklus.</li><li>• Maven hat Konventionen, um Quellcode zu platzieren und zu kompilieren. Es ist nicht erforderlich, Projektinformationen in der Datei <i>pom.xml</i> bereitzustellen.</li><li>• Es ist ein Framework.</li><li>• Maven-Plugins sind wiederverwendbar.</li><li>• Es sind hauptsächlich die Projektmanagement-Tools.</li></ul>	<ul style="list-style-type: none"><li>• Wenn es viele Abhängigkeiten für das Projekt gibt, dann ist es einfach, diese Abhängigkeiten mit Maven zu handhaben.</li><li>• Wenn die Abhängigkeitsversion häufig aktualisiert wird, dann muss man nur die Versions-ID in der Pom-Datei aktualisieren, um die Abhängigkeiten zu aktualisieren.</li><li>• Kontinuierliche Builds, Integration und Tests können einfach mit Maven gehandhabt werden.</li></ul>

Gradle	<ul style="list-style-type: none"> <li>• Gradle bietet inkrementelle Builds: Projekt wird nur ausgeführt, wenn Änderungen vorgenommen werden</li> <li>• Build-Caching: Wenn die Aufgabe bereits auf dem anderen Computer erstellt wurde, kann sie die Ausführung lokal überspringen</li> <li>• inkrementelle Anmerkungsverarbeitung: erhöhen Sie die Effektivität der inkrementellen Kompilierung, wenn unterstützte Anmerkungen vorhanden sind</li> <li>• parallele Ausführungen: bietet die parallele Ausführung von Aufgaben und aufgabeninterne Arbeit über die Worker API</li> <li>• Task-Timeouts: Verwenden Sie diese Option, um die Ausführungszeit von Tasks zu begrenzen.</li> </ul>	<ul style="list-style-type: none"> <li>• einfach zu bedienen</li> <li>• Es unterstützt auch Multiprojektstrukturen.</li> <li>• Es ist einfach, von Maven zu Gradle zu migrieren.</li> <li>• Es bietet leistungsstarke und skalierbare Tools.</li> <li>• Es verwendet eine auf Groovy basierende spezifische Sprache.</li> <li>• Es vermeidet Kompilierung.</li> <li>• Es ist leistungsfähiger als Maven, da es nur die aktuelle Aufgabe überwacht.</li> <li>• Es ist sinnlos, Anwendungen durch Hinzufügen neuer Funktionen zu entwickeln.</li> </ul>	<ul style="list-style-type: none"> <li>• Bekannt dafür, Support zu leisten, verwenden wir Gradle für ANT-Build-Projekte. Aufgaben können aus ANT-Build-Projekten importiert und in Gradle wiederverwendet werden. Gradle unterstützt auch Maven-Repositories, die zum Veröffentlichen und Abrufen von Abhängigkeiten des Projekts erstellt werden.</li> </ul>
--------	--	---	---

Wir werden als Build-Tool *Maven* nutzen, denn:

- Zusammenfassend vereinfacht und standardisiert Maven den Projekterstellungsprozess.
- Es bewegt die Teamzusammenarbeit, Kompilierung, Verteilung, Dokumentation und separate Aufgaben nahtlos.

## 4 Prototyping der Benutzerschnittstelle

Lösung	Besonderheiten	Zugänglichkeit	Bedienbarkeit	Leistung	Fazit: Eignung für den verfolgten Zweck
wireframe	<ul style="list-style-type: none"> <li>Verwendung für Erstellung einfacher Mockups</li> </ul>	<ul style="list-style-type: none"> <li>grundsätzlich frei erhältlich, jedoch kann eine Premiumversion in Höhe von ca. 15€ im Monat erworben werden</li> </ul>	<ul style="list-style-type: none"> <li>Sprache: Englisch</li> <li>über Browser bedienbar</li> <li>unterstützt die neusten Versionen von Chrome, Safari und Firefox, andere Browser werden evtl. auch unterstützt, dafür gibt es aber keine Garantie</li> </ul>	<ul style="list-style-type: none"> <li>Teilen des Arbeitslinks möglich</li> <li>Kommentarfunktion</li> <li>Erstellung von Protokollverläufen</li> <li>Gestaltung von Interaktionen</li> <li>Export von PNG- und PDF-Dateien (Premium)</li> <li>Team-Kollaboration möglich (Premium)</li> </ul>	<ul style="list-style-type: none"> <li>Einsicht auf Arbeitsverlauf durch andere Teammitglieder gut, jedoch sind die Leistungen allgemein für unseren Zweck zu gering</li> </ul>
gomockingbird	<ul style="list-style-type: none"> <li>Wireframe-Mockup-Tool</li> </ul>	<ul style="list-style-type: none"> <li>kostenpflichtig (je nach dem wie viele Projekte man bearbeiten möchte steigt der Preis → 3 Projekte = ca. 11€/Monat bis unbegrenzte Anzahl an Projekten = ca. 84€/Monat)</li> </ul>	<ul style="list-style-type: none"> <li>über Browser bedienbar</li> <li>Sprache: Englisch</li> </ul>	<ul style="list-style-type: none"> <li>Verknüpfung mehrerer Mockups miteinander</li> <li>Anzeige einer interaktiven Vorschau möglich</li> <li>Teamkollegen können in Echtzeit zusammen Wireframes bearbeiten</li> </ul>	<ul style="list-style-type: none"> <li>Echtzeit-Zusammenarbeit gut und Leistung für unsere Zwecke geradezu ausreichend jedoch sind die Kosten auf mind. 3 Projekte gelegt, dabei wird von uns nur eins bearbeitet</li> </ul>
popapp	<ul style="list-style-type: none"> <li>App, die Skizzen in anklickbare Prototypen verwandelt</li> <li>weitverbreitet</li> </ul>	<ul style="list-style-type: none"> <li>kostenlos</li> </ul>	<ul style="list-style-type: none"> <li>auf mehreren Sprachen bedienbar</li> <li>über App bedienbar</li> </ul>	<ul style="list-style-type: none"> <li>Erstellung von Prototypen durch Foto von Handskizze</li> <li>Simulation der Benutzeroberfläche</li> <li>Nutzung von zahlreichen Fotos, Schriftarten und Symbolen möglich</li> </ul>	<ul style="list-style-type: none"> <li>Leistungen sehr gut für unsere Zwecke geeignet</li> <li>Teilen an Teamkollegen gut</li> </ul>



	<ul style="list-style-type: none"> <li>• Nutzer sind z.B. Yahoo, Zynga, LinkedIn</li> <li>• von Marvel aufgekauft</li> <li>• wird noch stetig aktualisiert</li> </ul>		<ul style="list-style-type: none"> <li>• Prototyp kann auf jedem Gerät ausprobiert werden</li> <li>• einfache Bedienung</li> </ul>	<ul style="list-style-type: none"> <li>• Teilen des Prototypen an Kollegen/Freunde möglich</li> </ul>	
hotgloo	<ul style="list-style-type: none"> <li>• sehr gute Bewertungen, auch von großen Unternehmen</li> </ul>	<ul style="list-style-type: none"> <li>• kostenpflichtig (ca. 11€/Monat für 4 Nutzer und 2 Projekte bis ca. 48€/Monat für 20 Nutzer und 30 Projekte)</li> </ul>	<ul style="list-style-type: none"> <li>• einsteigerfreundlich (umfassende Dokumentation aller Features &amp; Online-Tutorials)</li> <li>• auf allen Geräten über den Browser bedienbar</li> <li>• Sprache: Deutsch/Englisch</li> </ul>	<ul style="list-style-type: none"> <li>• über 2000 User Interface Elemente und Icons für jegliche Devices: Desktop, Mobile, Wearables</li> <li>• Möglichkeit Seitennotizen oder Kommentare zu hinterlassen</li> <li>• bei Lizenz Zusammenarbeit von bis zu 100 Personen möglich</li> </ul>	<ul style="list-style-type: none"> <li>• Leistungen gut für unsere Zwecke geeignet</li> </ul>
ninjamock	<ul style="list-style-type: none"> <li>• ist das einzige Online Wireframe Tool, das einen Vektor-Editor enthält</li> <li>• enthält Cloud-basiertes Dateisystem</li> <li>• wird auch von größeren Unternehmen genutzt</li> </ul>	<ul style="list-style-type: none"> <li>• kostenpflichtig (5€/Monat pro Person)</li> </ul>	<ul style="list-style-type: none"> <li>• auf allen Geräten über den Browser bedienbar</li> <li>• einfache Bedienung</li> <li>• Anleitung zur Nutzung vorhanden</li> <li>• Sprache: Englisch</li> </ul>	<ul style="list-style-type: none"> <li>• liefert alle typischen Steuerelemente, die in mobilen Apps und Web-Designs verwendet werden</li> <li>• umfangreiche Auswahl von Vektor-Icons für Mockups</li> <li>• Echtzeit-Zusammenarbeit mit mehreren Nutzern möglich</li> <li>• Testen der Anwendung durch klickbare Wireframes</li> <li>• Export in PNG, HTML oder PDF möglich</li> </ul>	<ul style="list-style-type: none"> <li>• für unsere Zwecke eher ungeeignet, da sich Erstellung der Prototypen eher auf Apps und Web-Designs spezialisiert</li> </ul>
moqups	<ul style="list-style-type: none"> <li>• enthält Cloud-basiertes Dateisystem</li> <li>• eine der wenigen Anwendungen bei der Erstellung von</li> </ul>	<ul style="list-style-type: none"> <li>• ersten 3 Monate kostenlos, danach kostenpflichtig (ca. 13€/Monat für einen Nutzer bis ca. 67€/Monat für</li> </ul>	<ul style="list-style-type: none"> <li>• über Browser bedienbar</li> </ul>	<ul style="list-style-type: none"> <li>• Echtzeit-Zusammenarbeit mit mehreren Nutzern möglich</li> <li>• umfassende Bibliothek an Widgets und Icons</li> <li>• integrierte Schablonen-Kits für Mobile-Apps und Webdesigns</li> </ul>	<ul style="list-style-type: none"> <li>• sehr gut für unsere Zwecke geeignet</li> <li>• Echtzeit-Zusammenarbeit ist ein großer Vorteil</li> </ul>

	Diagrammen möglich ist • bereits über 2 Mio. zufriedene Nutzer	unbegrenzt viele Nutzer)		• Import von Bildern • schnelle Umwandlung von Entwurf zu interaktivem Prototypen • umfangreiche Auswahl an Schriftarten und Styling-Optionen	
pencil	• eines der wenigen Anwendungen bei der die Erstellung von Diagrammen möglich ist	• kostenlos	• über Desktop-Anwendung für Linux, Windows und Mac bedienbar • etwas schwieriger zu bedienen als Einsteiger	• bietet eine Sammlung verschiedener integrierter Schablonen zum Zeichnen verschiedener Arten von Benutzeroberflächen, von Desktop-bis zu mobilen Plattformen • beliebte Zeichnungsfunktionen sind ebenfalls in Pencil implementiert, um die Zeichenvorgänge zu vereinfachen • Export in mehrere Formate möglich (PNG, PDF, Webseite, Textdokument etc.)	• Leistungen sind für unsere Zwecke ausreichend
Auf Papier	• keine Installation oder Erstellung eines Accounts nötig	• sehr geringer Kostenaufwand	• sehr einfach • keine digitalen Anforderungen	• keine	• ungenügend für unsere Zwecke

Wir werden *moqups* als Tool zur Erstellung des Prototyp der Benutzerschnittstelle benutzen, da es von seinen Leistungen her gut unseren Zwecken entspricht und es sich schon bei vielen anderen Nutzern bewährt hat. Zudem hat es den Vorteil, dass hier eine Echtzeit-Zusammenarbeit möglich ist, was die Bearbeitung dieses Projekts deutlich vereinfacht. Ein Kostenaufwand wäre hier auch nicht vorhanden aufgrund der ersten drei kostenlosen Monate, die man nutzen kann.

## 5 IDE/Editor

Lösung	Besonderheiten	Zugänglichkeit	Bedienbarkeit	Leistung	Fazit: Eignung für den verfolgten Zweck
Eclipse	<ul style="list-style-type: none"> <li>eine der beliebtesten IDEs</li> <li>ursprünglich als Entwicklungsumgebung für Java genutzt, wird jedoch mittlerweile aufgrund der Erweiterbarkeit der Anwendung auch für viele andere Entwicklungsaufgaben eingesetzt</li> </ul>	<ul style="list-style-type: none"> <li>kostenlos</li> </ul>	<ul style="list-style-type: none"> <li>über Desktop-Anwendung bedienbar</li> <li>plattformunabhängig</li> </ul>	<ul style="list-style-type: none"> <li>Versionsverwaltung</li> <li>integrierter Hub zur gemeinschaftlichen Arbeit in Teams</li> <li>Syntaxhervorhebung und automatische Einrückung des Quelltexts</li> <li>viele Erweiterungen durch Installation von Paketen möglich, somit könnten auch Programmiersprachen wie C, C++, Python etc. unterstützt werden</li> </ul>	<ul style="list-style-type: none"> <li>gut für unsere Zwecke geeignet</li> </ul>
IntelliJ IDEA	<ul style="list-style-type: none"> <li>tiefe Codeanalyse und sehr intelligente Codekorrektur</li> </ul>	<ul style="list-style-type: none"> <li>kostenpflichtig</li> <li>2 verschiedene Versionen vorhanden („Ultimate“ und „All Products Pack“)</li> <li>1. Version kostet 149€ und 2. Version 249€ bei persönlicher Nutzung)</li> <li>kostenlose Community-Version vorhanden</li> </ul>	<ul style="list-style-type: none"> <li>als Desktop-Anwendung bedienbar</li> <li>plattformunabhängig</li> <li>einfacher Einstieg</li> <li>Sprache: Englisch</li> <li>ausführliche Anleitung zur Bedienung auf Webseite verfügbar</li> </ul>	<ul style="list-style-type: none"> <li>Tools zur Versionskontrolle</li> <li>automatisches Refactoring von Code</li> <li>Funktionsumfang kann durch Plugins erweitert werden</li> <li>automatische Codevervollständigung</li> <li>unterstützt Vielzahl von Programmiersprachen (u.a. Java, Python, SQL)</li> <li>Datenbanktools vorhanden</li> <li>Erstellung von Diagrammen möglich</li> <li>Live-Vorlagen, die ein schnelles Programmieren ermöglichen</li> <li>Paarprogrammierung in Echtzeit über Remote-Verbindung möglich sowie Video- und Sprachanrufe</li> </ul>	<ul style="list-style-type: none"> <li>sehr gut für unsere Zwecke geeignet</li> <li>Online-Paarprogrammierung von Vorteil</li> </ul>

NetBeans	<ul style="list-style-type: none"> <li>• gehört zu Apache</li> <li>• relativ ähnlich zu Eclipse</li> </ul>	<ul style="list-style-type: none"> <li>• kostenlos</li> </ul>	<ul style="list-style-type: none"> <li>• als Desktop-Anwendung bedienbar</li> <li>• plattformunabhängig</li> <li>• Sprache: Englisch</li> </ul>	<ul style="list-style-type: none"> <li>• Syntaxhervorhebung und Refactoring des Codes</li> <li>• Funktionsumfang durch Installation von Plug-ins erweiterbar</li> <li>• unterstützt Java und noch viele andere Programmiersprachen wie C, C++ oder Python</li> <li>• GUI-Builder ermöglicht grafische GUI-Gestaltung</li> <li>• Versionskontrollsysteme vorhanden</li> </ul>	<ul style="list-style-type: none"> <li>• Leistungen sind für unsere Zwecke vollkommen ausreichend</li> </ul>
Nano	<ul style="list-style-type: none"> <li>• Texteditor</li> </ul>	<ul style="list-style-type: none"> <li>• kostenlos</li> </ul>	<ul style="list-style-type: none"> <li>• als Desktop-Anwendung auf Unix-ähnlichen Betriebssystemen und Windows bedienbar</li> <li>• gut für Anfänger geeignet, da Bedienung einfach ist</li> <li>• auch auf deutscher Sprache bedienbar</li> </ul>	<ul style="list-style-type: none"> <li>• nicht sehr leistungsfähig, unterstützt jedoch grundlegende Funktionen (Bearbeiten, Ausschneiden, Kopieren, Einfügen etc.)</li> <li>• Syntaxhervorhebung</li> </ul>	<ul style="list-style-type: none"> <li>• Leistungen sind ungenügend für unsere Zwecke</li> </ul>

Als IDE werden wir *Eclipse* benutzen, da wir mit dieser Entwicklungsumgebung bereits vertraut sind und auch das UML-Tool Papyrus integriert ist, welches wir auch nutzen wollen.

## 6 Testautomatisierung

Lösung	Besonderheiten	Zugänglichkeit	Bedienbarkeit	Fazit: Eignung für den verfolgten Zweck
Selenium	<ul style="list-style-type: none"> <li>• unterschiedliche Sprachen, Browserimplementierung</li> </ul>	<ul style="list-style-type: none"> <li>• Lizenz: kostenlos</li> <li>• unterstützte Technologien: Webseiten, Webapplikationen, Shops</li> </ul>	<ul style="list-style-type: none"> <li>• einfach und effektiv</li> <li>• eins der beliebtesten Software Testing Tools für Oberflächenautomatisierung von Webanwendungen</li> </ul>	<ul style="list-style-type: none"> <li>• umfasst breites Spektrum und ist kostenlos</li> </ul>
Visual Studio Coded UI Tests	<ul style="list-style-type: none"> <li>• stabile Testautomatisierung von Standard-Windows-Anwendungen und Webseiten</li> </ul>	<ul style="list-style-type: none"> <li>• Lizenz: MSDN</li> <li>• Technologien: Net, WPF, WinForms, Webseiten, Webapplikationen, Shops</li> </ul>	<ul style="list-style-type: none"> <li>• relativ unübersichtliche Codegenerierung</li> </ul>	<ul style="list-style-type: none"> <li>• für Standard-Windows-Anwendungen, aber unübersichtlicher Code</li> </ul>
Ranorex Automation Framework	<ul style="list-style-type: none"> <li>• umfangreiche Testautomatisierung</li> </ul>	<ul style="list-style-type: none"> <li>• Lizenz: Floating Runtime (690€), Node-Locked Premium (1900€), Floating Premium (3490€)</li> <li>• Technologien: Net, WPF, WinForms, Java, Swing, AWT, Webseiten, Shops, Flex, Flash, iOS, Android, Windows Phone</li> </ul>	<ul style="list-style-type: none"> <li>• angenehmes Ansprechen/Identifizieren der Objekte</li> <li>• Tool-Support auf deutsch möglich</li> </ul>	<ul style="list-style-type: none"> <li>• sehr umfangreich, aber sehr preisintensiv</li> </ul>
Watir	<ul style="list-style-type: none"> <li>• automatisierte Tests für Webapplikationen</li> </ul>	<ul style="list-style-type: none"> <li>• Lizenz: kostenlos, Open Source</li> <li>• Technologien: Webapplikationen, Webbrowser werden angesteuert (sehr breitgefächert), xpath zur object identification</li> </ul>	<ul style="list-style-type: none"> <li>• Skriptsprache: Ruby</li> <li>• Unterstützt nur Internet Explorer</li> </ul>	<ul style="list-style-type: none"> <li>• kostenlos, aber nicht für unsere Anwendung geeignet</li> </ul>

TestComplete	<ul style="list-style-type: none"> <li>• deckt großen Umfang von Technologien-Applications-Arten ab</li> </ul>	<ul style="list-style-type: none"> <li>• Lizenz: 30 Tage kostenlos, danach kostenpflichtig</li> <li>• Technologien: Desktop, Web, Mobile Applications, NET, Java, WPF (XAML) Java, HTML5, AJAX, C/C++, Delphi, Qt, Android, iOS</li> </ul>	<ul style="list-style-type: none"> <li>• Programmiersprachen: VBScript, JScript, C++Script, C#Script, DelphiScript, Ruby</li> </ul>	<ul style="list-style-type: none"> <li>• großer Umfang, aber kostenpflichtig</li> </ul>
Jubula	<ul style="list-style-type: none"> <li>• automatisierte funktionale Tests</li> </ul>	<ul style="list-style-type: none"> <li>• Lizenz: kostenlos, Open Source</li> <li>• Technologien: Swing-Anwendungen, SWT-/RCP-/GEF-Anwendungen, JavaFX-Anwendungen, HTML-Anwendungen, iOS-Apps</li> </ul>	<ul style="list-style-type: none"> <li>• kann in Eclipse integriert werden (kann bereits in Eclipse enthalten sein)</li> <li>• Nutzung über GUI</li> <li>• Ist auch als Stand-Alone-Version verfügbar</li> </ul>	<ul style="list-style-type: none"> <li>• Eclipse-Schnittstelle und kostenlos</li> </ul>
Certify	<ul style="list-style-type: none"> <li>• Tool zur Automatisierung von End-To-End-Business-Prozessen auf oberer Abstraktionsebene</li> </ul>	<ul style="list-style-type: none"> <li>• Lizenz: kostenpflichtig</li> </ul>	<ul style="list-style-type: none"> <li>• für hochkomplexe IT-Systeme</li> </ul>	<ul style="list-style-type: none"> <li>• für sehr komplexe Systeme, aber für unsere Anwendung zu groß</li> </ul>
Testsigma	<ul style="list-style-type: none"> <li>• Plattform für automatisierte Web-, iOS- und Android-Apps sowie API-Tests</li> </ul>	<ul style="list-style-type: none"> <li>• Lizenz: kostenpflichtig</li> </ul>	<ul style="list-style-type: none"> <li>• enthält Funktion für Cross-Browser-Testing</li> <li>• Tests in Englisch</li> </ul>	<ul style="list-style-type: none"> <li>• zusätzliche Funktion, aber kostenpflichtig</li> </ul>

Wir werden für die Test-Automatisierung *Jubula* verwenden, da es kostenlos ist und wir bereits mit Eclipse arbeiten. Dadurch ist die Anwendung einfacher.

**Update vom 23.04.2022:** Wir entscheiden uns für das Tool *Junit*, da dieses seine Tests näher am Quellcode durchführt und unabhängig von einer grafischen Oberfläche arbeitet.

## 7 Dokumentationstool

Lösung	Besonderheiten	Zugänglichkeit	Bedienbarkeit	Fazit: Eignung für den verfolgten Zweck
FlowShare	<ul style="list-style-type: none"> <li>• automatisierte Schritt-für-Schritt-Anleitung</li> </ul>	<ul style="list-style-type: none"> <li>• Format: Word, PowerPoint, HTML, PDF, PNG</li> <li>• Software für Windows</li> <li>• Lizenz: 21€/Monat</li> </ul>	<ul style="list-style-type: none"> <li>• nimmt am PC jeden Mausklick automatisch auf</li> <li>• Speicherung in Dokument mit Screenshots, Pfeil und kurz automatisch erstellter Textbeschreibung (kann bearbeitet werden)</li> </ul>	<ul style="list-style-type: none"> <li>• automatisierte Dokumentation, aber nur für Windows und nicht kostenfrei</li> </ul>
Greenshot	<ul style="list-style-type: none"> <li>• Screenshots erstellen und bearbeiten</li> </ul>	<ul style="list-style-type: none"> <li>• Lizenz Windows: kostenlos</li> <li>• Lizenz Mac: 1,99\$</li> </ul>	<ul style="list-style-type: none"> <li>• Screenshot vom ganzen Bildschirm oder ausgewählten Bereich</li> <li>• Ergänzungen (Grafikelemente, Beschriftungen) im Editor</li> </ul>	<ul style="list-style-type: none"> <li>• grundlegendes Tool, aber ähnliches Programm existiert bereits bei Windows</li> </ul>
Snagit	<ul style="list-style-type: none"> <li>• Screenshots erstellen und bearbeiten</li> <li>• Videoaufzeichnungsfunktion</li> </ul>	<ul style="list-style-type: none"> <li>• Lizenz: 53,97€</li> </ul>	<ul style="list-style-type: none"> <li>• Screenshots können direkt in Snagit bearbeitet werden</li> <li>• große Auswahl an Werkzeugen zur Bearbeitung</li> </ul>	<ul style="list-style-type: none"> <li>• umfangreiche Bearbeitungstools, aber zu Preisintensiv</li> </ul>
Problem Step Recorder	<ul style="list-style-type: none"> <li>• Aufzeichnung von Prozessen mittels Screenshots</li> <li>• Problemaufnahme und Weiterleitung an Supportstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Windows, bereits vorinstalliert</li> <li>• Dokumente als Zip speichern und teilen</li> </ul>	<ul style="list-style-type: none"> <li>• Kommentare können hinzugefügt werden</li> <li>• Design der Anleitung ist wenig anwendergerecht</li> </ul>	<ul style="list-style-type: none"> <li>• ist bereits vorinstalliert und einfach in der Handhabung, allerdings keine Automatisierung</li> </ul>
Screensteps	<ul style="list-style-type: none"> <li>• Screenshots erstellen und Anbindung an Wiki-System</li> </ul>	<ul style="list-style-type: none"> <li>• Lizenz: 2500\$/Jahr mit 25 aktiven Nutzern</li> </ul>	<ul style="list-style-type: none"> <li>• Screenshots mit nachträglicher Bearbeitung</li> <li>• Sammlung in Knowledge Bases, Kollegen zuordnen</li> </ul>	<ul style="list-style-type: none"> <li>• Anbindung an Wiki-System, aber preisintensiv und für ca. 25 Nutzer verwendbar</li> </ul>

iorad	<ul style="list-style-type: none"> <li>• Aufzeichnung von im Browser gemachter Klicks und Schritt-für-Schritt-Anleitung</li> </ul>	<ul style="list-style-type: none"> <li>• kann per Link geteilt werden oder auf eigener Website oder Zendesk veröffentlicht werden</li> <li>• Lizenz: 200\$/Monat</li> </ul>	<ul style="list-style-type: none"> <li>• automatische Generierung von Screenshots und Texten, kann aber auch bearbeitet werden</li> </ul>	<ul style="list-style-type: none"> <li>• Automatisierung, aber nur im Browser und preisintensiv</li> </ul>
-------	--	---	---	--

Wir werden für die Dokumentation *Problem Step Recorder* verwenden, da es kostenlos ist und man dadurch Screenshots erstellen und bearbeiten kann. Anschließend wird die Dokumentation in einem Word-Dokument erstellt.

**Update vom 23.04.2022:** Aus Gründen der Praktikabilität entscheiden wir uns um für das Dokumentationstool *Javadoc*.



## 8 Obfuscator

Obfuskation kommt bei der Veröffentlichung des Produktes zum Einsatz und soll die Software für den Menschen möglichst unlesbar machen, um eine unbefugte Einsicht in die Programmstruktur zu verhindern. Damit soll bspw. Reverse Engineering und Diebstahl erschwert werden. Dies kann manuell oder automatisiert durch Tools erfolgen.

Da das Projekt auf der Programmiersprache Java basieren soll, werden hier verschiedene Lösungsansätze speziell für Java verglichen.

Lösung	Besonderheiten	Zugänglichkeit und Bedienbarkeit	Sicherheit	Fazit: Eignung für den verfolgten Zweck
ProGuard	<ul style="list-style-type: none"> <li>• Obfuskation, Optimierung und Komprimierung von bereits kompilierten Java-Bytecode</li> <li>• letzte Aktualisierung: 18. Februar 2022 (aktuell)</li> <li>• professionelle Lösung</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Lizenz:</b> frei (GPL 2+)</li> <li>• plattformübergreifend</li> <li>• grafisch und in Kommandozeile nutzbar</li> </ul>	<ul style="list-style-type: none"> <li>• hohe Sicherheit <ul style="list-style-type: none"> <li>➢ tiefgreifende Obfuskation</li> </ul> </li> <li>• wird u.a. von Google empfohlen</li> </ul>	<ul style="list-style-type: none"> <li>• <b>sehr gut</b> <ul style="list-style-type: none"> <li>➢ aktuell</li> <li>➢ hohe Sicherheit</li> <li>➢ professionell und umfangreich</li> </ul> </li> </ul>
JavaGuard	<ul style="list-style-type: none"> <li>• Obfuskation von Java-Bytecode</li> <li>• letzte Aktualisierung: 26. Februar 2016 (leicht veraltet)</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Lizenz:</b> frei (LGPLv2)</li> <li>• plattformübergreifend</li> </ul>	<ul style="list-style-type: none"> <li>• ausreichend als Obfuscator</li> <li>• Software im Beta-Status <ul style="list-style-type: none"> <li>➢ Sicherheitslücken möglich</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• <b>ausreichend</b> <ul style="list-style-type: none"> <li>➢ jedoch noch in Entwicklungsphase</li> </ul> </li> </ul>
DashO	<ul style="list-style-type: none"> <li>• tiefgreifende Software zur Erhöhung der Sicherheit von Java-Anwendungen</li> <li>• große Vielfalt an Features, u.a.: <ul style="list-style-type: none"> <li>➢ Obfuskation</li> <li>➢ Codeoptimierung</li> <li>➢ String- und Ressourcenverschlüsselung</li> <li>➢ Schutz vor unbefugten Eingriffen während der Laufzeit</li> </ul> </li> <li>• professionelle Lösung</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Lizenz:</b> proprietär</li> <li>• Lizenz muss käuflich erworben werden</li> <li>• grafische Oberfläche</li> </ul>	<ul style="list-style-type: none"> <li>• sehr hohe Sicherheit <ul style="list-style-type: none"> <li>➢ tiefgreifende Obfuskation</li> <li>➢ zusätzliche Sicherheitsfeatures</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• <b>schlecht</b> <ul style="list-style-type: none"> <li>➢ zu umfangreich in Relation zum Projektumfang</li> <li>➢ zusätzliche Kosten</li> </ul> </li> </ul>

yGuard	<ul style="list-style-type: none"> <li>• Obfuskation und Komprimierung von bereits kompilierten Java-Bytecode</li> <li>• letzte Aktualisierung: 1. März 2022 (aktuell)</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Lizenz:</b> frei (MIT)</li> </ul>	<ul style="list-style-type: none"> <li>• hohe Sicherheit <ul style="list-style-type: none"> <li>➢ Obfuskation von Namen</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• <b>gut</b> <ul style="list-style-type: none"> <li>➢ aktuell</li> <li>➢ hohe Sicherheit</li> </ul> </li> </ul>
Zelix KlassMaster	<ul style="list-style-type: none"> <li>• große Vielfalt an Features, u.a.: <ul style="list-style-type: none"> <li>➢ Obfuskation verschiedener Codeelemente</li> <li>➢ String- und Integerverschlüsselung</li> <li>➢ Datenreduzierung</li> </ul> </li> <li>• professionelle Lösung</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Lizenz:</b> proprietär</li> <li>• Lizenz muss käuflich erworben werden (265 USD und höher)</li> <li>• grafische Oberfläche</li> <li>• eigene Skriptsprache</li> </ul>	<ul style="list-style-type: none"> <li>• hohe Sicherheit <ul style="list-style-type: none"> <li>➢ vielseitige Obfuskation</li> <li>➢ zusätzliche Sicherheitsfeatures</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• <b>schlecht</b> <ul style="list-style-type: none"> <li>➢ hohe zusätzliche Kosten</li> </ul> </li> </ul>
Manuelle Obfuskation	<ul style="list-style-type: none"> <li>• Manuelles Verschleiern des Programmcodes</li> <li>• kein direkter Eingriff in den Java-Bytecode</li> </ul>	<ul style="list-style-type: none"> <li>• keine zusätzliche Software notwendig</li> <li>• starke Erhöhung des Arbeitsaufwandes</li> </ul>	<ul style="list-style-type: none"> <li>• geringere Sicherheit durch unberührten Java-Bytecode</li> <li>• hohe Wahrscheinlichkeit für Lücken und Fehler</li> </ul>	<ul style="list-style-type: none"> <li>• <b>schlecht</b> <ul style="list-style-type: none"> <li>➢ geringe Sicherheit</li> <li>➢ hoher zusätzlicher Arbeitsaufwand</li> </ul> </li> </ul>
Keine Obfuskation	<ul style="list-style-type: none"> <li>• Auslassen der Obfuskation</li> </ul>	<ul style="list-style-type: none"> <li>• keine zusätzliche Software notwendig</li> <li>• kein zusätzlicher Arbeitsaufwand</li> </ul>	<ul style="list-style-type: none"> <li>• keine</li> </ul>	<ul style="list-style-type: none"> <li>• <b>schlecht</b> <ul style="list-style-type: none"> <li>➢ keine Sicherheit</li> </ul> </li> </ul>

Für die Verwendung im Projekt wurde das Tool *ProGuard* ausgewählt, da mit dieser Software die höchste Sicherheit erzielt werden kann, während der zusätzliche Aufwand (Arbeitszeit und Kosten) minimal bleibt. Zudem können gleichzeitig Effizienz und Speicherbedarf des Produktes optimiert werden.

## 9 Code Conventions

Code Conventions sind festgelegte Richtlinien über Programmiermethoden und die Formatierung des Programmcodes, sodass dieser auch durch Außenstehende einfach lesbar ist.

Da das Projekt auf der Programmiersprache Java basieren soll, werden hier verschiedene Lösungsansätze speziell für Java verglichen.

Lösung	Besonderheiten	Zugänglichkeit und Bedienbarkeit	Fazit: Eignung für den verfolgten Zweck
Oracle <sup>4</sup>	<ul style="list-style-type: none"><li>• weit verbreitet</li><li>• stammt vom Entwickler von Java</li><li>• letzte Änderung: 20. April 1999 (nicht aktuell)<ul style="list-style-type: none"><li>➤ neuere Java-Features werden nicht behandelt</li></ul></li></ul>	<ul style="list-style-type: none"><li>• frei zugänglich</li><li>• geringer Einarbeitungsaufwand durch häufige Nutzung</li></ul>	<ul style="list-style-type: none"><li>• <b>gut</b><ul style="list-style-type: none"><li>➤ geringer Einarbeitungsaufwand</li></ul></li></ul>
Google <sup>5</sup>	<ul style="list-style-type: none"><li>• weit verbreitet</li><li>• letzte Änderung: 22. Mai 2018 (aktuell)<ul style="list-style-type: none"><li>➤ Großteil der Java-Features ist abgedeckt</li></ul></li></ul>	<ul style="list-style-type: none"><li>• frei zugänglich</li><li>• geringer bis mittlerer Einarbeitungsaufwand</li></ul>	<ul style="list-style-type: none"><li>• <b>gut</b><ul style="list-style-type: none"><li>➤ geringer bis mittlerer Einarbeitungsaufwand</li></ul></li></ul>
Eigene Konventionen	<ul style="list-style-type: none"><li>• projektspezifischer Programmierstil<ul style="list-style-type: none"><li>➤ Festlegung eigener Standards</li></ul></li></ul>	<ul style="list-style-type: none"><li>• firmenintern zugänglich</li><li>• großer zusätzlicher Aufwand bei der Festlegung</li><li>• großer Einarbeitungsaufwand durch eigene Standards</li></ul>	<ul style="list-style-type: none"><li>• <b>schlecht</b><ul style="list-style-type: none"><li>➤ hoher Einarbeitungsaufwand</li><li>➤ zusätzlicher Zeitaufwand bei der Festlegung</li></ul></li></ul>

Für den Quellcode im Projekt sollen die Code Conventions von *Oracle* genutzt werden, da deren Grundsätze weit verbreitet sind und häufig verwendet werden. Dies minimiert den Einarbeitungsaufwand. Da diese Standards jedoch veraltet sind und sich deshalb nicht auf neuere Java-Features beziehen, werden bei der Verwendung solcher Features projektspezifische Richtlinien festgelegt und eingehalten.

---

[4] Link: <https://www.oracle.com/java/technologies/javase/codeconventions-contents.html>

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

## 10 Kollaborationstool

Kollaborationstools sollen die Kommunikation, Organisation und Zusammenarbeit vor allem in Teams erleichtern, aber auch generell die Kommunikation erleichtern.

Lösung	Besonderheiten	Zugänglichkeit und Bedienbarkeit	Fazit: Eignung für den verfolgten Zweck
Microsoft Teams	<ul style="list-style-type: none"> <li>Kommunikation über Chats, Chatgruppen und Videokonferenzen</li> <li>Schnittstellen zu Microsoft-Office-Anwendungen</li> </ul>	<ul style="list-style-type: none"> <li>mittlerer Einrichtungsaufwand</li> <li>geringer Einarbeitungsaufwand</li> <li>schnelle Kommunikation</li> <li>Nutzbarkeit an Microsoft-Account gebunden</li> </ul>	<ul style="list-style-type: none"> <li><b>gut</b> <ul style="list-style-type: none"> <li>➤ schnelle Kommunikation</li> <li>➤ Schnittstellen zu anderen Microsoft-Programmen</li> </ul> </li> </ul>
Discord	<ul style="list-style-type: none"> <li>Kommunikation über Chats, Chatgruppen und Videokonferenzen</li> <li>Möglichkeit zum Austausch von Dateien</li> </ul>	<ul style="list-style-type: none"> <li>geringer Einrichtungsaufwand</li> <li>geringer Einarbeitungsaufwand</li> <li>schnelle Kommunikation</li> <li>Nutzbarkeit an Account gebunden</li> </ul>	<ul style="list-style-type: none"> <li><b>gut</b> <ul style="list-style-type: none"> <li>➤ schnelle Kommunikation</li> <li>➤ geringer Einrichtungsaufwand</li> </ul> </li> </ul>
Issue Tracker (allgemein)	<ul style="list-style-type: none"> <li>Organisation über Tickets, die eine Teilaufgabe darstellen (je nach Lösung erstellt vom Auftraggeber oder -nehmer)</li> </ul>	<ul style="list-style-type: none"> <li>hoher Einrichtungs- und Organisationsaufwand</li> <li>mittlerer Einarbeitungsaufwand</li> </ul>	<ul style="list-style-type: none"> <li><b>schlecht</b> <ul style="list-style-type: none"> <li>➤ zu hoher Organisationsaufwand in Relation zur Projektgröße</li> </ul> </li> </ul>
Regelmäßige Absprachen in Präsenz	<ul style="list-style-type: none"> <li>synchrone Kommunikation in persönlichen Treffen</li> </ul>	<ul style="list-style-type: none"> <li>hohe Produktivität möglich</li> <li>hoher Organisationsaufwand</li> </ul>	<ul style="list-style-type: none"> <li><b>gut</b> <ul style="list-style-type: none"> <li>➤ für komplexe Themen (firmenintern und mit dem Auftraggeber)</li> </ul> </li> </ul>
Regelmäßige Absprachen über Videokonferenzen	<ul style="list-style-type: none"> <li>synchrone Kommunikation in Videokonferenzen</li> </ul>	<ul style="list-style-type: none"> <li>hohe Produktivität möglich</li> <li>geringer Organisationsaufwand, da ortsunabhängig</li> </ul>	<ul style="list-style-type: none"> <li><b>sehr gut</b> <ul style="list-style-type: none"> <li>➤ für komplexe Themen (firmenintern und mit dem Auftraggeber)</li> </ul> </li> </ul>
Kommunikation und Absprache per E-Mail	<ul style="list-style-type: none"> <li>asynchrone Kommunikation per E-Mail</li> </ul>	<ul style="list-style-type: none"> <li>allgemeines und gängiges Kommunikationsmittel</li> <li>Kommunikation oft zeitintensiv (und nicht immer bidirektional)</li> </ul>	<ul style="list-style-type: none"> <li><b>gut</b> <ul style="list-style-type: none"> <li>➤ für die Kommunikation mit dem Auftraggeber</li> </ul> </li> <li><b>schlecht</b> <ul style="list-style-type: none"> <li>➤ für firmeninterne Themen</li> </ul> </li> </ul>

Kommunikation und Absprache per Chat	<ul style="list-style-type: none"> <li>• asynchrone Kommunikation über eine Chat-Plattform</li> </ul>	<ul style="list-style-type: none"> <li>• geringer Einrichtungsaufwand</li> <li>• kein Einarbeitungsaufwand</li> <li>• deutlich schneller als über E-Mails</li> </ul>	<ul style="list-style-type: none"> <li>• <b><i>schlecht</i></b> für die Kommunikation mit dem Auftraggeber</li> <li>• <b><i>sehr gut</i></b> für firmeninterne Angelegenheiten</li> </ul>
---	---	--	---

Die Zusammenarbeit im Projekt soll in einer Kombination aus verschiedenen Lösungen erfolgen, die gemeinsam den zusätzlichen Einarbeitungs- und Organisationsaufwand minimieren und dabei dennoch die bestmögliche Organisation und Kommunikation ermöglichen.

Firmeninterne Themen sollen dabei vor allem über die Chat-Funktionalitäten von *Discord* behandelt werden. Diese Plattform besitzt zudem die Möglichkeit der Durchführung von Videokonferenzen, welche ggf. genutzt werden kann.

Die Kommunikation mit dem Auftraggeber soll vor allem über E-Mails und persönliche Treffen realisiert werden.

# 11 Quellen

## Versionierung

- Git, Hg: <https://www.perforce.com/blog/vcs/mercurial-vs-git-how-are-they-different>
- SVN: <https://www.javatpoint.com/svn>
- SVN, Git: <https://www.ionos.de/digitalguide/websites/web-entwicklung/git-vs-svn-versionsverwaltung-im-vergleich/>
- Hg: <https://www.incredibuild.com/blog/mercurial-vs-git-lets-examine>

## UML-Tool mit Quellcodegenerierung

- StarUML, Papyrus: <https://alternativeto.net/software/staruml/about/>
- VS: <https://towardsdatascience.com/drawing-a-uml-diagram-in-the-vs-code-53c2e67deffe>

## Build-Tool

- <https://www.guru99.com/apache-ant-tutorial.html>
- [https://www.tutorialspoint.com/maven/maven\\_overview.htm](https://www.tutorialspoint.com/maven/maven_overview.htm)
- <https://www.javatpoint.com/difference-between-ant-and-maven>
- <https://gradle.org/features/>
- <https://www.geeksforgeeks.org/difference-between-gradle-and-maven/>
- <https://www.simplilearn.com/tutorials/jenkins-tutorial/what-is-jenkins>
- <https://www.javatpoint.com/jenkins-vs-maven>
- <https://www.vogella.com/tutorials/ApacheAnt/article.html#:~:text=Ant%20is%20primarily%20used%20for,repetitive%20tasks%2C%20e.g.%20generating%20documentation.>
- <https://www.geeksforgeeks.org/introduction-apache-maven-build-automation-tool-java-projects/>

## Prototyping der Benutzerschnittstelle

- wireframe
  - <https://www.omt.de/online-marketing-tools/wireframe-tools/>
  - <https://wireframe.cc/>
- gomockingbird
  - <https://gomockingbird.com/home>
- popapp
  - <https://www.crunchbase.com/organization/popapp>

- <https://apps.apple.com/de/app/pop-prototyping-on-paper/id555647796>
- hotgloo
  - <https://www.hotgloo.com/>
- ninjamock
  - <https://ninjamock.com/>
- moqups
  - <https://moqups.com/>
- pencil
  - <https://pencil.evolus.vn/>

### **IDE/Editor**

- Eclipse
  - <https://mission-mobile.de/knowhow/eclipse-ide/>
- IntelliJ IDEA
  - [https://de.wikipedia.org/wiki/IntelliJ\\_IDEA](https://de.wikipedia.org/wiki/IntelliJ_IDEA)
  - <https://www.jetbrains.com/de-de/idea/>
- NetBeans
  - <https://netbeans.apache.org/>
  - [https://de.wikipedia.org/wiki/NetBeans\\_IDE](https://de.wikipedia.org/wiki/NetBeans_IDE)
- Nano
  - [https://de.wikipedia.org/wiki/Nano\\_\(Texteditor\)](https://de.wikipedia.org/wiki/Nano_(Texteditor))

### **Testautomatisierung**

- <https://www.testing-board.com/testautomatisierung-tools/?cn-reloaded=1>

### **Dokumentationstool**

- <https://getflowshare.com/de/7-tools-zur-softwaredokumentation-die-ihr-leben-einfacher-machen/>

### **Obfuscator**

- [https://de.wikipedia.org/wiki/Obfuskation\\_\(Software\)](https://de.wikipedia.org/wiki/Obfuskation_(Software)) (abgerufen am 05.04.2022, 17:49 Uhr)
- <https://de.wikipedia.org/wiki/ProGuard> (abgerufen am 05.04.2022, 17:53 Uhr)
- <https://sourceforge.net/projects/javaguard/> (abgerufen am 07.04.2022, 08:42 Uhr)
- <https://www.guardsquare.com/proguard> (abgerufen am 07.04.2022, 08:56 Uhr)

- <https://github.com/Guardsquare/proguard/> (abgerufen am 07.04.2022, 08:59 Uhr)
- <https://www.preemptive.com/products/dasho> (abgerufen am 07.04.2022, 10:17 Uhr)
- <https://www.yworks.com/products/yguard> (abgerufen am 07.04.2022, 10:17 Uhr)
- <http://www.zelix.com/klassmaster/> (abgerufen am 07.04.2022, 10:17 Uhr)

### **Code Conventions**

- <https://codegym.cc/groups/posts/491-java-coding-conventions-which-ones-to-follow-and-why> (abgerufen am 05.04.2022, 17:32 Uhr)

### **Kollaborationstool**

- <https://de.wikipedia.org/wiki/Issue-Tracking-System> (abgerufen am 07.04.2022, 07:09 Uhr)