# Lösungsvarianten

## Sprache

* Java: Multiplatform Programmiersprache
* VBA: Scriptsprache für Office Wekrzeuge
* C#: Programmiersprache von Microsoft

## IDE

* Eclipse: Java basierte IDE
* VisualStudio: IDE von Microsoft

## Versionsverwaltung

* Git: Versionsverwaltung ursprünglich für den Linux-Kernel
* CVS: Alte bekannte Versionsverwaltung
* Subversion: Oft benutzte Versionsverwaltung

## Testing

* TDD: Test Driven Developement
* Tests im Nachhinein schreiben

## GUI

* WindowsForms: von Lehrer empfohlen
* GTK: Multiplatform
* QT: Multiplatform, viele Firmen wechseln von GTK zu QT

# Lösungsvarianten bewerten

### Vorteile Nachteile

|  |  |  |
| --- | --- | --- |
|  | Vorteile | Nachteile |
| Sprache |  |  |
| VBA |  | Alles hassen es |
| Java | Multiplatform | Keiner von uns hat Erfahrung mit Java |
| C# | Gute Integration in Microsoft Umfeld | Microsoft abhängig |
| IDE |  |  |
| Eclipse | Super für Java | Langsam |
| VisualStudio | Super für C# | Läuft nur auf Windows |
| Versionsverwaltung |  |  |
| Git | Skaliert super, wird oft benutzt | Relative kompliziert |
| CVS | Einfach | Community weniger hilfsbereit wie Git |
| Subversion |  | Community weniger hilfsbereit wie Git |
| Testing |  |  |
| TDD | Nach dem die Tests geschrieben sind ist es relative einfach zu programmieren | Man kann auf unerwartete Probleme schlechter Reagieren |
| Im Nachhinein | Sehr agile | Programmierung des Programms ein bisschen aufwändiger |
| GUI |  |  |
| WindowsForms | Von Lehrer empfohlen, Einfach | Nicht „mächtig“ |
| GTK | Grosse Funktionalität | Ästhetik schlecht |
| QT | Grosse Funktionalität |  |

## Entscheid

### Sprache

VBA konnten wir von Anfang an ausschliessen, da wir alle VBA aus tiefstem Herzen hassen. Schlussendlich gewann C#, da schon ein paar Mitglieder Erfahrung damit gehabt haben.

### IDE

Nach kurzer Recherche im Internet stellte sich heraus, dass alle VisualStudio für C# empfehlen. Dementsprechend benutzen wir VisualStudio.

### Versionsverwaltung

Hier haben wir uns für Git entschieden aus folgenden Gründen:

* Github.com ist sehr übersichtlich und wir oft benutzt
* Unser Projekt kann schier endlos gross werden, es genügt sogar für den Linux-Kernel
* Viele Leute aus unserer Gruppe haben schon Erfahrung damit

### Testing

Wir haben uns dazu entschieden die Tests im Nachhinein zu Schreiben. Niemand von uns hat jemals mit TDD gearbeitet. Wir befürchten, dass wir mit TDD zu wenig agile auf unerwartete Probleme reagieren können.

### GUI

Wir haben uns für WindowsForms entschieden, da es vom Lehrer empfohlen wurde und wir bereits Erfahrungen damit haben.