**10 Anforderungen an die Entwicklungsumgebung**

**10.1 Software**

Wie schon in 2.2.3 beschreiben, wird die Programmiersprache C# für die Entwicklung empfohlen, somit auch als ihre Umgebung das Visual Studio Express von Microsoft.

Da die wichtigste Anforderung an die entwickelte Software eine Windowsumgebung ist, wurde sich für eine von Microsoft selbst entwickelte Programmiersprache C-Sharp (C#) [http://psc.informatik.uni-jena.de/publ/CSharp-it0402.pdf; 01.11.2012, 10.14] und die dazu passende, von Microsoft kostenlos zur Verfügung gestellte Entwicklungsumgebung Visual Studio Express[http://www.microsoft.com/visualstudio/deu/downloads#d-2012-express; 01.11.2012, 10.30] entschieden, was den Vorteil hat, dass alle Zielrechner das .NET Framework von vornherein besitzen und keine externe Laufzeitumgebung vom Anwender installiert werden muss.

Zusätzliche Software für die Entwicklung ist ein Tool zur Versionskontrolle und eine Anwendung für die Continuous Integration(siehe 10.3).

Aufgrund guter Erfahrungen mit der Versionskontrollsoftware Git, wird diese für die Benutzung bei der Entwicklung empfohlen. Durch Erweiterungen kann Git auch in die Entwicklungsumgebung Visual Studio integriert werden[http://visualstudiogallery.msdn.microsoft.com/63a7e40d-4d71-4fbb-a23b-d262124b8f4c; 01.11.2012, 10.21].

**10.2 Hardware**

Die Rechnerhardware muss die Systemanforderungen der Entwicklungsumgebung und des Betriebssystems erfüllen.

Empfohlen wird ein Prozessor der neueren Generation und ausreichend Arbeitsspeicher, um alle Funktionen der Software ohne Performanceverlust testen und ausführen zu können.

Vorausgesetzt wird ein Monitor zur Ausgabe, Maus und Tastatur zur Eingabe der Software.

**10.3 Orgware**

Für den Betrieb der Continuous Integration wird ein CI-Server benötigt. Dazu gibt es eine Vielzahl von CI-Serverangebote mit einem Web-Interface, Empfehlenswert sind die Anwendungen Jenkins[http://jenkins-ci.org/; 01.11.2012, 10.49] und Teamcity[www.jetbrains.com/teamcity/; 01.11.2012, 10.51]. Es gibt auch die Möglichkeit einen eigenen CI-Server aufzusetzen.

Jenkins hat den Vorteil der Unterstützung von Git (10.1), benötigt aber zusätzliche Plug-Ins für die Programmierung unter dem .NET Framework. Teamcity dagegen ist von vornherein mit Visual Studio verträglich. Beide CI-Server geben Möglichkeiten zum Verwalten und grafischer Darstellung von Commits, Tests und Änderungen der Entwicklung.