Projektmanagment und Teamarbeit
Klasse 5IA, TFO Bozen*

Berger Dominik Bernard Simon Dalvai Ragnioli Peter
Baron von Treuheim Stefan Maximilian Di Pauli
Domaneg Andreas Gobbi Viktor
Hopfgartner Christopher Mussner Marcel
Niederegger Andreas Niedrist Alexander Nocker Thomas
Oberprantacher Hannes Pirpamer Damian
Pazeller Daniel Prader Matthias Rassele Phillip
Romen René Rizzolli Thomas Rufinatscha Samuel

29. Mai 2015

Zusammenfassung

TODO: Eine kurze Einleitung zum Dokument....

*Erstellt im Kurs "Projektmanagment und Betriebsorganisation", TFO Bozen, Mai 2015

Inhaltsverzeichnis

Abbildungsverzeichnis

1 SVN

1.1 Allgemeines über Subversion (SVN)

Apache Subversion (kurz: SVN) ist eine Software zur Versionsverwaltung von Dateien und Verzeichnissen. Es ist ein zentrales Verwaltungssystem. SVN ist unter der Apache-Lizenz 2.0 veröffentlicht und somit Open-Source. Bei SVN erfolgt die Versionierung über ein zentrales Repository. Bei einer Veränderung einer Datei oder eines Verzeichnisses werden zwischen dem Repository und dem Arbeitsplatz nur die Unterschiede übertragen. Merkmale eines zentralen Verwaltungssystemes:

- Es gibt nur eines Server aber mehrere Workingcopies
- Versionsnummern werden zentral vergeben
- Versionshistorie ist nur auf dem Server gespeichert
- ermöglicht Zugriffs- und Rechtemanagement

Abbildung 1: Subversion Logo

1.2 Geschichtlicher Hintergrund

SVN entstand im Jahre 2000 im amerikanischen Unternehmen CollabNet. Im November 2009 entschloss CollabNet, dass das Projekt (SVN) an die Apache Software Foundation geht. Danach wurde das Projekt von Apache zu Apache Subversion unbenannt.

1.3 Vorteile gegenüber CVS

Die wichtigsten Vorteile

- Versionsschema
  - CVS kann nur die Versionsgeschichte von einzelnen Dateien speichern. SVN von Dateien und Verzeichnissen
- Änderungsverfolgung
  - CVS muss bei einer Änderung immer die gesamte Datei übertragen
- Umbenennungen und Verschiebungen

  - Bei einer Umbenennung oder Verschiebung einer Datei wird in CVS die gesamte Versionsgeschichte gelöscht
- Löschmarkierung von Verzeichnissen
  - In CVS können nur leere Verzeichnisse gelöscht werden

1.4 Vorteile gegenüber GIT

Die wichtigsten Vorteile

- Kostenfrei
- Sehr gut getestet
- Sehr aktuell, da es stetig weiterentwickelt wird von Apache
- Sehr gute Unterstützung von IDE's und Shared Hosting Anbietern (z.B. SourceForge)
- Einfache Handhabung

1.5 Die Architektur von Subversion

Subversion ist auf einem Client-Server-Modell aufgebaut. Auf dem Computer des Benutzers wird ein Client ausgeführt. Dieser kann über verschiedene Verbindungsarten auf den Server zugreifen. Die ganzen Information über Dateien (Informationen über Zustände, Versionen, Kommentare) und die Dateien selbst liegen auf dem Server. Das Repository selbst, welches auf dem Server liegt, kann entweder in einer BerkleyDB-Datenbank oder direkt auf dem Dateisystem gespeichert werden. Derzeit gibt es drei Zugriffsvarianten:

- HTTP / HTTPS
- SVN (subversioneigenes Übertragungsprotokoll)
- FILE (lokaler Zugriff)

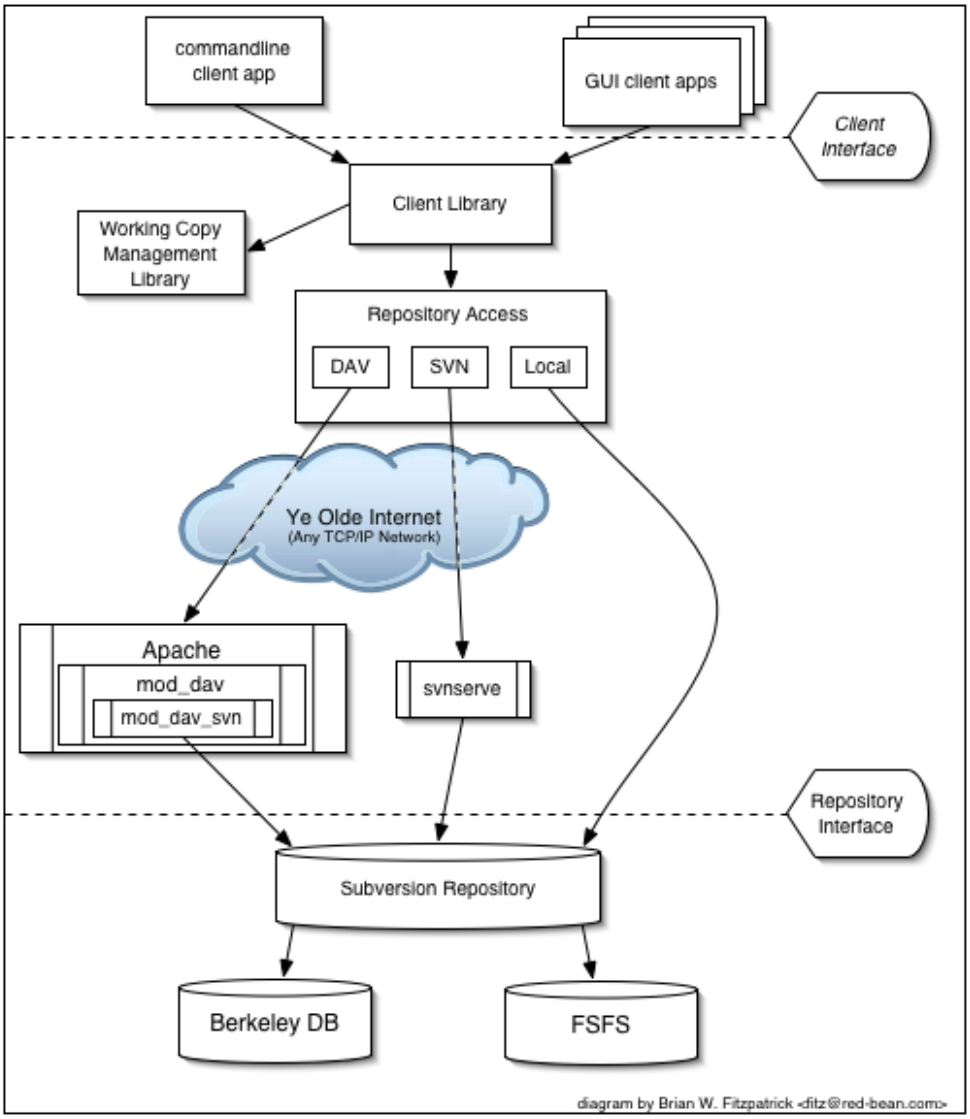

Abbildung 2: Subversion Architektur

1.6 Die Komponenten von Subversion

Subversion besteht aus verschiedenen Softwareteilen:

- svn
  - Kommandozeilenprogramm

- svnversion
  - Schreibt die Revisionsnummer der Arbeitskopie in die Standartausgabe
- svnlook
  - Werkzeug um ein Projektarchiv zu untersuchen
- svnadmin
  - Werkzeug um ein Projektarchiv zu erstellen, zu verändern oder zu reparieren
- mod_dav_svn
  - Plugin für den Apache-HTTP Server um ein Projektarchiv über ein Netzwerk verfügbar zu machen
- svnserve
  - Serverprogramm; Andere Möglichkeit um Projektarchiv über ein Netzwerk verfügbar zu machen (SSH)
- svnsync
  - Tool, um Revisionen eines Projektarchivs in ein anderes Projektarchiv zu überspielen.

1.7 Begriffe in SVN

Bei Subversion sind folgende Begriffe wichtig:

- Revision
- Changeset
  - Ein Changeset ist eine Zusammenfassung aller Änderungen einer Version. Bei einem Changeset gibt es folgende Notations:
    - U = Updated
    - D = Deleted
    - A = Added
- Delta / Diff
  - Unter Delta / Diff versteht man die Differenzen zwischen zwei Versionen. In SVN werden immer nur die Unterschieden zwischen zwei Versionen festgehalten.
- Merge
  - Unter Merge versteht man das Zusammenführen von verschiedenen Änderungen in zwei Versionen einer Datei.

- Branch
  - Ein Branch ist eine Kopie einer Ursprungsversion.
- Tag
  - Mit einem Tag kann man die einzelnen Versionen beschriften.

1.8 Terminal - Befehle für SVN

Neuen Baum auf den lokalen Rechner kopieren:

- svn checkout [Pfad] [Lokaler Name]

Lokalen Baum aktualisieren:

- svn update

Lokale Dateien hinzufügen und löschen:

- svn add file...
- svn remove file...

Übersicht über lokale Änderungen:

- svn status

Lokale Änderungen ins Repository übertragen:

- svn commit -m "Beschreibung der Änderungen-

1.9 Ist SVN noch aktuell?

SVN wird durchaus auch heutzutage noch verwendet. Jedoch ist seine größte Konkurrenz GIT, welches mittlerweile weiter verbreitet ist. Es gibt aber einen großen Unterschied zwischen GIT und SVN: GIT ist dezentralisiert und SVN zentralisiert. Deshalb wird SVN auch nicht so schnell verschwinden, da beide Systeme ihr Vor- und Nachteile haben.

2 Trello

Trello ist eine web-basierte Projektmanagementsoftware. Diese Software wurde von US-amerikanischen Unternehmen Fog Creek Software entwickelt. Trello nutzt MongoDB, Node.js und Backbone.js. Dieses Projekt wurde mit JavaScript verwirklicht.

Abbildung 3: Logo von Trello

2.1 Geschäftsmodell

Geschäftsmodell von Trello basiert auf dem Freemium-Prinzip: Die Grundfunktionen sind kostenlos verfügbar. Jedoch Extrafunktionen können nur mit einen Gold-Account genutzt werden.

2.2 Geschichte

Im Sommer 2010 beginnt Fog Creek Software mit einem potenziellen Produkt her zum zu entwickeln. Im Januar 2011 wird ein Prototyp vorgeschlagen, der zur Lösung einiger hochrangiger Planungsprobleme dienen soll. Sein Name lautet Trellis. Kurz danach beginnt die Entwicklung in Vollzeit.

Nach abgeschlossener Betaphase startet Trello im September 2011 auf der TechCrunch Disrupt mit Apps für Internet und iPhone. Ab hier wurde der Name Trello gewählt.

Der Dienst wurde am 13. September 2011 gegründet und ist nur in English verfügbar. Aktuell hat Trello 7 Millionen Benutzer (Stand: Januar 2015). Der Entwichlung dieses Dienstes fand nach dem Vorbild der Produktionsplanungsmethode Kanban statt. Dies kann über ihre Webanwendung trello.com oder über ihre App die für iOS, Android und Windows 8 verfügbar ist.

Im Sommer 2012 wurde Taco, der Husky von Fog Creek-Mitbegründer Joel Spolsky, Maskottchen von Trello. Trello erreicht die Marke von 500.000 Mitgliedern und führt die Trello-Android-App ein.

Abbildung 4: Maskottchen Taco

2.3 Nutzung

Ihr Dienst läuft über ihre sogenannten Boards. Hier können Listen bearbeitet und modifiziert werden. Es können Checklisten, Anhänge und mit einem festgelegten Termin versehen werden. Den Zugriff auf diese sogenannten Boards kann beliebig vielen Nutzern zur Verfügung gestellt werden. Dies kann durch ein einfaches Drag and Drop gemacht werden. Diese können sich bei der Planung und der Kommunikation des Projekts beteiligen. Hier können an die Bretter verschiedene Probleme angesprochen und zusammen gelöst werden. Der Dienst bietet den Nutzern die Möglichkeit Kommentare zu hinterlassen, Dateien hochzuladen, Checklisten erstellen, Termine und viele weitere Features. Daten können nicht nur vom Computer selbst sondern acuh von Google Drive, Droppbox oder OneDrive hinzugefügt werden. Um das ganze noch aufzupeppen sind „emojis“ auch natürlich verfügbar. Um auf den laufenden über sein Projekt zu sein, wird man in der App selbst benachrichtigt, über E-Mail, Desktop Benachrichtigungen über den Browser oder über „push notifications“.

Hier ein kleines Video dazu: https://goo.gl/ehSahR

3 Bonita BPM

3.1 Was ist Bonita BPM?

Bonita PBM ist eine Open Source Software Suite für Prozessmodellierung, Geschäftsprozessmanagement (BPM) und Arbeitsablaufverwaltung (Workflow-Engine). Damit ist es möglich Prozesse zu automatisieren, dern Ergebnisse zu messen und diese Prozesse anschließens zu verbessern. Die Suite besteht aus 3 Teilen:

- *Bonita Studio*: ein Studio in dem der Benutzer die Geschäftsprozesse grafisch bearbeiten kann. Im Studio kann auch die vom Web aus erreichbare Geschäftsanwendung generiert werden. Eine weitere Funktion des Bonita Studios ist es die Geschäftsprozesse mit anderen Standards und Technologien zu designen.
- *Bonita Portal*: ein Protal das dem Endbenutzer und den Administratoren mittels seiner Webseite eine übersicht über die Aufgaben und Prozessse zur Verfügung stellt.
- *Bonita Execution Engine*: eine Java Engine um programmiertechnisch mit den Prozessen zu interagieren

3.2 Wer braucht BonitaBPM?

- *Angestellte:* Mit Bonita BPM haben Angestellte einen überblick darüber, welche Arbeiten anstehen und in welcher Reihenfolge diese abzuarbeiten sind.
- *Teamchefs:* Mit der Hilfe von Bonita BPM können Teamchefs Aufgaben auf ihr Team verteilen.
- *Manager:* Durch die Aktivitätsanalyse die mit Hilfe von Diagrammen erstellt werden kann, können Manager die Produktivität der Firma erhöhen.

3.3 Funktionalitäten

Bonita BPM bringt verschiedene Funktionen um das Geschäftsprozessmanagement (BPM) von Projekten zu planen, zu entwickeln, ausführen und zu überwachen:

- *Live Updates der Prozesse*: Neue Versionen von Prozessen können in der Produnktionsumgebung einfach aktualisiert werden
- *Versionskontrolle der Prozesse*: Während der modellierung der Prozesse können verscheidene provisorische Versionen der Prozesse gespeichert und verwaltet werden

- *Simulation der Prozesse*: Bonita BPM bietet die Möglichkeit dir Prozesse mit verschiedenen Parametern wie Kosten, Dauer, Ressourcenkonsum usw. zu simulieren.
- *Zuweisen und filtern von Aktivitäten*: Mit den eingebauten Instrumenten und Filtern ist es möglich einer oder mehrerer Personen dynamisch und effizient Aktivitäten zuzuweisen.
- *Zentrales Repository*: Im zentralen Firmenrepository ist es möglich alle Prozesse zu speichern, organisieren und archivieren

3.4 Weitere Eigenschaften

Bonita BPM macht es möglich das Geschäftsprozessmanagement mit weiteren Bereichen eines Unternehmens zu verbinden. Zum Beispiel mit dem Kundenbeziehungsmanagement oder dem Ressourcenplanungsmanagement. Des weiteren bietet die Software Suite einen Kommentarfeed zu jedem spezifischen Fall um die Kommunikation innerhalb des Unternehmnens zu fördern. Bonita BPM steht in den Sprachen Englisch, Französich und Spanisch zur Verfügung. Die Interfaces können durch ein eigenes Werkzeug aber in jede beliebige Sprache übersetzt werden. Es ist weiters möglich Prozesse in Echtzeit zu ändern oder diese zu analysieren.

4 DokuWiki

4.1 Was ist DokuWiki?

DokuWiki ist eine freie Wiki-Software,die in der Programmiersprache PHP geschrieben und unter der GPL 2 lizenziert wurde. Dokuwiki wurde von Andreas Gohr 2004 ins Leben gerufen. Um Inhalte und Metadaten zu speichern, werden einfache Textdateien genutzt und keine SQL-Datenbanken, wie bei anderen Wiki-Engines. Um die Wikiquellseiten gut leserlich zu halten, werden Inhalt und Metadaten von Wikiseiten bei DokuWiki strikt getrennt. Anfangs zur einfachen Dokumentation von Projekten gedacht, wird Dokuwiki mittlerweile aufgrund seiner Einfachheit und Funktionen für eine Vielzahl von Anwendungen eingesetzt. Auf Basis einer übersichtlichen Struktur lassen sich mit Erweiterungen (Plugins) weitere Funktionen hinzufügen, etwa für Blogs, Mediendaten oder Arbeitsgruppen.

4.2 Merkmale

4.2.1 Versionsverwaltung

Die Versionverwaltung speichert alle Versionen einer Wikiseite. Es ist möglich, ältere Versionen mit der aktuellen Version zu vergleichen. Außerdem wird verhindert, dass mehrere Benutzer gleichzeitig eine Seite verändern können.

4.2.2 Zugriffskontrolle

Die Zugriffsrechte lassen sich für Kombinationen von Benutzern, Gruppen und Namespaces vergeben. Die Einstellung ist via Webinterface (Usermanager) oder manuell per Konfigurationsdatei möglich (Access Control List).

4.2.3 Add-ons

DokuWiki hat einen einfachen Add-on-Mechanismus.Dadurch ist es möglich Erweiterungen (Plugins) in PHP zu schreiben. Es gibt inzwischen eine ganze Reihe an Erweiterungen (¿300). Über den Plug-in-Manager können diese über die Web-Oberfläche in das eigene Wiki integriert und verwaltet werden.

4.2.4 Templates

Das Aussehen des Wikis kann der Administrator über Templates festlegen. Es wurden inzwischen unterschiedliche Templates von der Entwicklergemeinde zur Verfügung gestellt.

4.2.5 Internationalisierung

Als Standard-Zeichencodierung wird UTF-8 verwnedet. Somit sind auch Sprachen wie Chinesisch, Thai oder Hebräisch darstellbar. Das Wiki selber kann momentan in 39 Sprachen konfiguriert werden.

4.2.6 Caching

Um den Server des Wikis zu entlasten, speichert ein Cache geparste Seiten. Bei einem erneuten Aufruf der Seite werden die gespeicherten Daten geliefert, anstatt die Wikiseite nochmals zu parsen.

4.2.7 Volltextsuche

DokuWiki hat eine Volltextsuche integriert, mit der in dem gesamten Wiki nach Stichwörtern gesucht werden kann.

4.2.8 WYSIWYG-Editor

Der Wiki-Philosophie einer einfachen Markup-Syntax entsprechend hat DokuWiki in der Grundausstattung keinen WYSIWYG-Editor. Diese Funktion kann aber über ein Plugin nachgerüstet werden; alternativ gibt es eine Quickbuttonleiste ähnlich MediaWiki.

4.2.9 Datenspeicherung

DokuWiki speichert alle Daten (aktuelle und alte Seiteninhalte, Indizes, Caches) in Textdateien. Dadurch ist keine separat laufende Datenbank (etwa MySQL) notwendig.

4.2.10 Versionierung/Synchronisation

Jede Wiki-Seite wird in einer Textdatei im Verzeichnis dokuwiki-JJJJ-MM-TT/data/pages gespeichert, der Name der Datei bleibt trotz Versionierung gleich. Vorherige Versionen befinden sich unter dokuwiki-JJJJ-MM-TT/data/attic. Es erfolgt kein Umbenennen/Neuanlegen der Originaldatei (z. B. Revision00011, Revision00012). Dies macht Dokuwiki ideal für Synchronisations-Tools mit beidseitigem Abgleich und diff-Funktion wie Unison.

4.2.11 Portable Version

Für Windows-Rechner ist DokuWiki auch als portable Version zusammen mit einem portablen Apache Webserver für die Verwendung auf einem USB-Stick vorhanden.

4.2.12 HTML5

Seit Release 2012-10-13 “Adora Belle” parst DokuWiki HTML5-Seiten.

Weitere Information über DokuWikie und ähnlicher Software findet man auf http://www.wikimatrix.org/.

Gradle Thomas Rizzolli Mai 2015

5 Gradle

5.1 Was ist Gradle

Gradle ist ein Build-Management Tool wie Maven, dass in Java geschrieben wurde. Anders als Maven verwendet Gradle Groovie um die zu bauenden Projekte zu beschreiben. Im Gegensatz zu Maven, das XML verwendet, sind Gradle-Skripts direkt ausführbarer Code.

Gradle wurde für Builds von Projekten konzipiert, die aus einer Vielzahl von Unterprojekten besteht. Da Builds umfangreicher Projekte sehr viel Zeit in Anspruch nehmen, unterstützt Gradle sowohl inkrementelles(stück für stück) als auch Bauen der Software durch parallel ablaufende Build-Prozesse. Dadurch können nur Teile eines Projektes gebaut werden, die geändert wurden, ohne das Ganze Projekt neu bauen zu müssen. Des Weiteren können Tests parallel zum laufenden Build gestartet werden. Dadurch wird die Geschwindigkeit des Bauens des Projektes um ein Vielfaches erhöht

Viele bekannte Frameworks verwenden Gradle, wie zum Beispiel Hibernate, Groovy, Grails und Spring Security. Auch Android ist seit Mitte 2013 hinzugekommen. Auch sogenannte "nativer" Systeme, welche nicht auf der Java-Plattform basieren, werden unterstützt. Dazu gehören die Programmiersprachen C++, C, Objective C und Assembler.

5.2 Aufbau

In Gradle braucht man in den meisten Fällen nur die build.gradle Datei. Gradle verfügt jedoch Über 2 weitere Dateien, die aber nicht unbedingt erforderlich sind.

- build.gradle - enthällt die Definition der Tasks und der ganzen Abhängigkeiten des Projektes
- settings.gradle - hier werden die teilnehmenden Unterprojekte eines Multiprojekts definiert.
- gradle.properties - enthällt eine Liste von Eigenschaften für eine projektspezifische Gradle-Initialisierung eines Builds

Weiteres wird in Gradle zwischen Tasks und Eigenschaften(Properties) unterschieden.

Tasks sind ausführbarer Code, der über die Kommandozeile aufgerufen wird, indem man dort "gradle [den auszuführenden Task]" eingibt.

Durch Properties werden die Eigenschaften des Builds, wie z.B. Pfad der Klassen und Abhängigkeiten definiert werden. Eigenschaften können in zwei verschiedenen Weisen definiert werden:

1. `sourceSets.main.java.srcDir "src/main/java"`

2.
```
sourceSets{
      main{
          java{
              srcDir "src/main/java"
          }
      }
  }
```

Beide Schreibweisen funktionieren, wobei meistens eine Kombination von 1. und 2. bevorzugt wird.

5.3 Einfache Grundtasks

Das Build-Script wird in der "build.gradle" geschrieben.
Zum Anfang ein einfaches Hello-World Programm:
In der build.gradle:

```
task helloworld {
    String hallo = 'Hallo'
    String welt = ' Welt!'
    println hallo + welt.toUpperCase()
}
```

5.4 Erstellen eines einfachen Java-Builds

Wie bei den Tasks, werden auch Build-Scripts in der build.gradle definiert:

```
apply plugin: "java"

sourceSets {
    main.java.srcDir "src/main/java"
}

jar {
    manifest.attributes "Main-Class": "net.tfobz.test.Test"
}
```

- Als erstes wird apply plugin: java gesetzt, damit Gradle mit Java-Projekten überhaupt umgehen kann.
- Durch sourceSets.main.java.srcDir , kann der Pfad für die Java-Packages gesetzt werden. Wird diese Eigenschaft nicht gesetzt, dann wird standartmäsig "src/main/java" als Pfad gewählt.
- Die Eigenschaft jar.manifest.attributes ergänzt die Manifest-Datei mit den gesetzten Attributen(in diesem Fall der Pfad der Main-Klasse).

5.4.1 JUnit-Tests mit Gradle durchführen

Die Tests werden in Gradle, ähnlich wie in Maven, getrennt vom Quellcode abgelegt. Standardmäßig ist der Pfad "src/main/test" eingestellt. Man kann diesen Pfad aber, wie es auch im vorherigen Beispiel gemacht wurde, ändern.

Folgendes Beispiel soll zeigen, wie man JUnit in seinem Projekt integrieren kann:

```
repositories {
    mavenCentral()
}

sourceSets {
    main.java.srcDir "src/main/java"
    test.java.srcDir "src/main/test"
}

dependencies {
    testCompile 'junit:junit:4.12'
}

test {
    testLogging {
        events 'started', 'passed', 'failed'
    }
}
```

- Die Funktion mavenCentral() in der Eigenschaft repositories holt die erforderlichen Bibliotheken für JUnit(Internetverbindung erforderlich).
- In sourceSets.test.java.srcDir kann der Pfad der JUnit Testklassen geändert werden.
- In der Eigenschaft testCompile in dependencies wird die zu verwendende JUnit-Version angegeben.
- In der Eigenschaft test.testLogging.events werden die anzuzeigenden JUnit-Meldungen angegeben. Wird z.B. 'failed' weggelassen, so werden die gescheiterten JUnit-Tests nicht in der Konsole ausgegeben (Build schlägt aber trotzdem fehl).

5.4.2 Java Projekt eclipsfähig machen

Um ein Gradle-Projekt eclipsfähig zu machen braucht man nur das Plugin "eclipse" hinzuzufügen.

```
apply plugin: "java"
apply plugin: "eclipse"
```

- Gibt man in der Kommandozeile den Befehl “gradle eclipse” ein, werden die benötigten Dateien für Eclipse erzeugt. Das Projekt an sich wird aber weder compiliert noch getestet.
- Mit dem Befehl “gradle cleanEclipse” werden die erzeugten Dateien wieder entfernt.

6 Microsoft Dynamics AX

6.1 Einführung

Microsoft Dynamics® AX ist eine objektorientierte ERP-Lösung von Microsoft® für mittelständische Unternehmen und Großunternehmen, die es ermöglicht, Wandel und Veränderungen aus dem Geschäftsumfeld aufzunehmen und mitzugestalten und so den Geschäftserfolg voranzutreiben. Es handelt sich um eine leistungsstarke Lösung, mit der man in kürzester Zeit einen nachhaltigen Nutzen erzielen kann. Dank der 36 verfügbaren Landesversionen eignet sich die Software insbesondere für den Einsatz in multinationalen Organisationen. Microsoft Dynamics AX 2012 stellt den Beginn eines neuen Produktivitätszeitalters bei Unternehmenslösungen dar. Es handelt sich um die neueste Version der erfolgreichen ERP Software Reihe und beeindruckt mit einem deutlichen Zuwachs an Funktionalität, einer völlig neuen Agilität und einer Benutzeroberfläche, die überzeugt und Produktivität beim Anwender fördert.

6.2 Ausgangssituation zur Entwicklung der ERP Software

Zur Entwicklung dieser ERP Software wurden relevante aktuelle Business und IT-Anforderungen einbezogen: steigender globaler Wettbewerb, verbunden mit signifikantem Kosten- und Preisdruck, Organisationsänderungen, aber Prozessoptimierungen, das Entstehen von komplexen internationalen Netzwerken oder erhöhte Anforderungen durch steigende gesetzliche Kontrolle sind wichtige Trends, der sich Unternehmen stellen müssen.Es wird absolute Transparenz für schnelle und richtige Entscheidungen gefordert. Viele Organisationen haben aber zunehmend Schwierigkeiten, mit diesen Trends bzw. Beweggründen Schritt zu halten, da ein altes und überholtes ERP System weder die nötige Flexibilität noch Prozessorientierung aufweist, um hier effizient agieren zu können.

6.3 Die vier Prinzipien von AX

6.4 Leistungsstartk

Folgende Eigenschaften tragen zur hohen Performanz der Microsoft ERP Software bei:

- Umfassende ERP Software-Funktionen sowohl für administrative als auch für operative Unternehmensbereiche
- Reports können durch direkte Integration der Business Intelligence Funktionen des SQL-Servers rasch und effektiv verarbeitet werden.

- Tasks eines Benutzers werden in einem zentralen Aufgabenbereich zusammengeführt und gegebenenfalls einer funktionalen Eingabewarteschlange zugewiesen.

6.5 Agil

Die Geschäftstätigkeit des Unternehmens kann durch eine Sammlung von einheitlichen Organisationsmodellen, überwacht, analysiert, bewertet und gegebenenfalls modifiziert werden. Das Microsoft Dynamics ERP System löst zwei essentielle Fragen auf überzeugende Art und Weise: Wie gut bildet das ERP System die reale Welt ab und wie schnell lässt sie sich an das Unternehmen anpassen?

6.6 Einfach

Das Prinzip der Einfachheit spiegelt sich bei Installation, Implementierung, Anpassung, der Anwendungsoberfläche sowie bei Updates wieder, wobei die Grundlage zur Einfachheit die vertraute Benutzeroberfläche bildet.

6.7 Global

Eine globale Ausrichtung der aktuellen Microsoft ERP Software wird durch folgende Funktionen erreicht:

- Ab dem AX 2009-Release wurden Lokalisierungen für 38 Länder zur Verfügung gestellt.
- Zentrale Stammdatenhaltung für Produkte und Geschäftspartneradressen wurden mit dem aktuellsten AX 2012 Release hinzugefügt.

7 Activiti

Activiti ist ein freies Workflow-Management-System mit dem man Businessprozesse definieren und ausführen kann. Es ist in Java geschrieben und damit Plattform unabhängig. Es ist ausgerichtet auf Geschäfts Leute, Entwickler und System Administratoren. Die Activiti Engine ist leight weight und auf Java Entwickler ausgelegt.

7.1 Übersicht

In folgender Grafik sind alle Teile der Activiti Software in den nächsten Kapitel werden die wichtigsten erklärt.

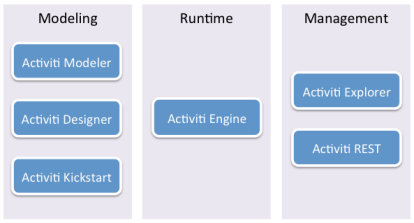

Abbildung 5: Teile der Activiti Software

7.2 Activiti Engine

Die Activiti Engine ist das Herzstück des Projekts. Es ist eine Business Process Model and Notation 2 Engine mit API für Java. Es gibt dem Entwickler die Möglichkeit mit Listener bei bestimmten Ereignissen eigenen Java Code einzubauen um so mehr technische Details ins Diagramm zu bringen. Des weiteren können eigene Aktivitäten für das Diagramm definiert werden. Die ganze Engine ist von Anfang an darauf ausgelegt Cloud fähig zu sein.

7.3 Activiti Explorer

Der Activiti Explorer ist eine Webanwendung mit der das System Verwaltet werden kann. Sie ermöglicht alles vom erstellen neuer Aufgaben bis zum Abschließen jener. Es können Arbeiter verwaltet werden und deren Aufgaben ebenfalls. Für User zeigt die Webanwendung ihre aktuellen Aufgaben an und welche sie als nächstes machen müssen. Die Anwendung führt dabei History wer wann was macht.

7.4 Activiti Modeler

Der Activiti Modeler ist eine Webanwendung die es ermöglicht Grafische BPMN 2.0 Diagramme zu erstellen die ausgeführt werden können von der Activiti Engine.

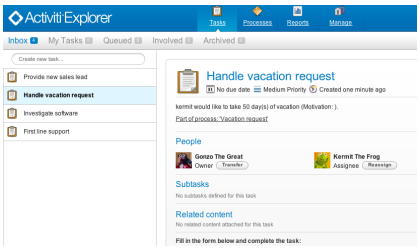

Abbildung 6: Der Activiti Explorer

7.5 Activiti Designer

Der Activiti Designer ist im Grunde das selbe wie der Activiti Modeler, es ist aber ein Eclipse plugin. Entwickler können damit direkt in Eclipse BPMN Diagramme für ihre Programme erstellen.

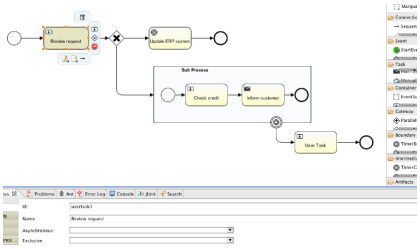

Abbildung 7: Der Activiti Designer

8 ANT

Apache Ant(Ameise), „Another Neat Tool“, ist ein in Java geschriebenes Buildmanagementprogramm. Ein Buildmanagementprogramm(BMP) ist ein Programm, welches zum automatisierten Erzeugen von ausführbaren Computerprogrammen aus existierenden Quelltexten, Bibliotheken und sonstigen Dateien verwendet wird. Ein weiteres Beispiel für ein BMP ist das weitverbreitete MAKE. Ant ist Open Source, startete als Teil des sogenannten Jakarta-Projekts und ist nun ein Apache-Top-Level-Projekt.

8.1 Entwicklung

Entwickelt wurde die erste Version von James Duncan Davidson, der 1999 ein Werkzeug wie make für Java benötigte. Davidson gilt außerdem als Vater von Tomcat. Für ihn steht der Name „ANT“ dafür, dass es als kleines Programm, genau wie die kleinen Ameisen, Großes leisten kann.

8.2 Funktion

Wie auch bei MAKE wird ANT durch eine XML-Datei, die so genannte Build-Datei, gesteuert. Standartmäßig wird diese Build-Datei build.xml genannt. In der Build-Datei wird ein project definiert welches das Wurzelelement der XML-Datei darstellt. Zu einem Software-Projekt sollte immer nur eine Build-Datei und ein ANT-Projekt gehören, um Umordnung zu vermeiden. ANT wird von vielen Java-Werkzeugen unterstütz und lässt sich ganz einfach in eigenen Projekte integrieren.

8.3 Begriffe

Targets (Ziele): V ergleichbar mit Funktionen in Programmiersprachen welche von außen über die Kommandozeile oder die Entwicklungsumgebung gezielt aufgerufen werden können.

Abhängigkeiten: Beim Aufrufen eines Targets löst Ant Abhängigkeiten auf und arbeitet die Targets entsprechend ab.

Tasks(Aufgaben): Sie sind vergleichbar mit Befehlen in Programmiersprachen. Tasks können auch selbst erstellt und beliebig erweitert werden.

8.4 Syntax

Da es sich bei der Build-Datei um eine XML-Datei handelt, hängt ihre Bedeutung nicht von Tabulatorzeichen, Leerzeichen oder Pfadtrennzeichen ab, die auf unterschiedlichen Betriebssystemen unterschiedlich definiert sind. Dies ist insbesondere eine Verbesserung gegenüber den von MAKE benutzten Makefiles.

8.5 Häufig verwendete Tasks

Ant enthält über 150 Tasks, wobei man auch eigene Tasks in Java selbst programmieren kann. Diese Liste enthält einige Standarttasks von Ant:

javac zum Kompilieren von Quellcode. copy zum Kopieren von Dateien. delete zum Löschen von Dateien oder Verzeichnissen. mkdir zum Erstellen von Verzeichnissen. junit für automatisierte (JUnit-)Tests. move zum Umbenennen von Dateien oder Verzeichnissen. exec zum Ausführen von System-Programmen. zip zum Zippen, also zum Komprimieren von Dateien. cvs zum Durchführen von CVS-Operationen. mail zum Versenden von E-Mails. replace zum Ersetzen von Text in Dateien.

8.6 Beispiele

```
<?xml version="1.0"?>
 <project name="Demo" basedir="." default="build">
  <property name="build.classes" value="bin" />
  <property name="build.lib" value="lib" />
  <property name="java.dir" value="." />
  <property name="name" value="Wikipedia-Demo" />
  <property name="manifest" value="manifest" />

  <path id="classpath">
      <pathelement location="." />
   </path>

  <!-- Anwendung bauen  -->
  <target name="build" depends="clean" description="Baut die komplette Anwendung">
    <!-- Verzeichis anlegen -->
    <mkdir dir="${build.classes}"/>

    <!-- Quelltext kompilieren -->
    <javac srcdir="${java.dir}"
           destdir="${build.classes}"
           debug="false"
           deprecation="true"
           optimize="true" >
      <classpath refid="classpath" />
    </javac>
```

9 GitHub

9.1 Einführung

Github ist im Grunde eine Website, auffindbar unter http://github.com. GitHub erlaubt es, online Dateien anhand Versionsverwaltungssystems Git abzuspeichern. Daher auch der Name.

Meist wird GitHub für Softwareprojekte benutzt und normalerweise auch in Gruppen zu mehreren Personen. Das ist jedoch nicht bedingt. Viele Benutzer nutzen GitHub auch z.B. als Backup eigener Dateien oder einfach um Dateien (welche auch gar nichts mit Software zu tun haben) online anderen Nutzern zur Verfügung zu stellen.

9.2 Geschichte GitHubs

Erschienen ist GitHub erstmals im Februar 2008. Github, Inc., welche die Website nun betreibt, hieß damals noch Logically Awesome. Entwickelt wurde es von Chris Wanstrath, PJ Hyett und Tom Preston-Werner, wobei sie Ruby on Rails und Erlang benutzten. Eigentümer der Website ist die GitHub, Inc., welche seit 2007 besteht. Ihren Sitz hat sie in San Francisco.

Am 24. Februar 2009 wurde bekanntgegeben, dass innerhalb eines Jahres eine Zahl von 46.000 öffentlichen Repositories erreicht wurde. Davon wurden allein 17.000 im Januar desselben Jahres erstellt. Geforkt wurden zu dieser Zdeit 6.200 Repositories und gemerget wurden 4.600. Einige Monate darauf, am 5. Juli 2009 wurde dann die 100.000-User-Marke überschritten. Im selben Monat wurde weiters bekanntgegegeben, dass der Dienst nun 135.000 Repositories hostet, wovon 90.000 öffentlich sind.

Die Seite wuchs weiterhin ziemlich schnell: Am 25. Juli 2010 wurden bereits eine Million Repositories auf GitHub gehostet. Nicht ganz ein Jahr später, am 20. April 2011 waren es dann bereits doppelt so viel. 2013 wurde die 10-Millionen-Marke überschritten.

Im Folgejahr wurde die Seite in Russland aufgrund von Suizid-Referenzen und in Indien aufgrund von Repositories der IS blockiert. Indien machte die Website jedoch im Jahr darauf wieder der Öffentlichkeit zugänglich.

Am 26. März 2015 griff die chinesische Regierung die Website mit einer groß ausgelegten DDoS-Attacke (Distributed Denial of Service) an. Grund war ein Projekt, welches die New York Times für chinesische Leser bereitstellte. Es hostete von der chinesischen Regierung gesperrte Webseiten, sodass diese in China trotzdem lesbar wurden. Die chinesische Suchmaschine Baidu wurde so manipuliert und mit Schadcode versehen, dass Besucher der Website Teil der DDoS-Attacke wurden.

9.3 GitHub

GitHub ist, ähnlich Bitbucket, SourceForge und Google Code, ein Online-Hosting-Service. Anhand des Versionskontrollservices Git kann man Dateien

auf GitHub laden, aktualisieren und holen. GitHub wird zum größten Teil für Open-Source-Software-Projekte benutzt.

Im Gegenteil zu beispielsweise SourceForge ist nicht das Projekt die zentrale Sammlung von Code, sondern ein vom User erstelltes Repository. Jeder User kann beliebig viele Repositories sammeln. Sofern diese öffentlich gemacht werden, sind sie kostenfrei. Private Repositories sind kostenpflichtig.

Ein Repository ist, wie bereits gesagt, eine Sammlung von Code. Ein Repository hat immer einen Besitzer, welcher mehrere Mitarbeiter festlegen kann. User, welche nicht Mitglied eines Repository sind, können von Git nicht direkt in dieses pushen. GitHub erlaubt es, Repositories zu „forken“. Ein Fork ist eine Kopie eines Repository, mit welcher gearbeitet werden kann, ohne dass das ursprüngliche Repository verändert wird. Änderungen können mit Git von und nach einem Repository geladen werden. Möchte man dem ursprünglichen Ersteller eines geforkten Repository bitten, Änderungen in seinem originalen Repository aufzunehmen, so erstellt man einen Pull Request. Der Ersteller kann diesen dann annehmen oder ablehnen.

Neben einfachen Quellcode Hosting bietet GitHub auch weitere projektspezifische Dienste an:

- README-Dateien Jedes Projekt kann eine README-Datei beinhalten, welche sich im Root der Projektstruktur befinden muss. Diese mit Markdown formatierte Datei kann GitHub dann auf der Projektseite anzeigen. Die Datei wird häufig zur Vorstellung des Projekts und für Installation/Benutzung sowie Regelung zur Mitarbeit benutzt.

- Issue Tracking Weiters stellt GitHub einen eigenen Issue Tracker für jedes Repository bereits. Dieser hat fundamentale Funktionalität: Issues können erstellt, diskutiert und geschlossen werden. Issues können Tags gegeben werden, um sie beispielsweise als Frage, Diskussion, Bug oder Feature markieren zu können (die Tags können beliebige Namen haben). Genauso können Issues Milestones zugewiesen werden. Auch können GitHub-User von anderen beauftragt werden, ein Issue zu lösen.

- Projekt Wiki Jedes Repository kann ein eigenes Wiki besitzen. Dieses kann vom Besitzer und den Mitarbeitern eines Repository bearbeitet werden. Es wird meist dazu genutzt, allgemeneie Funktionalitäten des Programms oder der Bibliothek zu beschreiben, sodass der Einstieg für einen Neuling einfacher ist.

- Projekt Website GitHub erlaubt das Hosten einer kleinen Website. Diese ist unter http://projektname.github.io erreichbar. Sie ist für eine kurze Vorstellung des Projekts gedacht.

9.4 GitHub Enterprise

GitHub Enterprise ist für Großbetriebe gedacht, welche GitHub firmenintern hosten möchten und nur ihre eigenen Projekte darauf nutzen wollen. GitHub Enterprise ist kostenpflichtig, der Einstiegspreis liegt bei 5.000$ für 20 Personen.

9.5 Gist

Gists werden bereitgestellt, um kleine Code-Schnipsel schnell und einfach anzuzeigen. Anders als bei ähnlichen Diensten wie Pastebin können nicht nur einzelne Dateien geteilt werden, sondern auch mehrere Dateien bis hin zu ganzen Applikationen. Auch verwendet GitHub im Hintergrund nach wie vor Git, so dass auch die Gists unter einem Versionskontrollsystem laufen.

10 MediaWiki

MediaWiki ist ein freies Softwarepaket zum Betrieb eines Wikis, das ursprünglich für die Wikipedia geschrieben wurde. Es ist das Wikisystem aller von der Wikimedia-Stiftung (Wikimedia Foundation) betriebenen Wikis sowie vieler anderer Wikis.

10.1 Erklährung für wiki:

Ein Wiki (hawaiisch für „schnell“), seltener auch WikiWiki oder WikiWeb genannt, ist ein Hypertextsystem für Webseiten, deren Inhalte von den Benutzern nicht nur gelesen, sondern auch online direkt im Webbrowser geändert werden können (Web-2.0-Anwendung). Das Ziel ist häufig, Erfahrung und Wissen gemeinschaftlich zu sammeln (kollektive Intelligenz) und in für die Zielgruppe verständlicher Form zu dokumentieren. Die Autoren erarbeiten hierzu gemeinschaftlich Texte, die ggf. durch Fotos oder andere Medien ergänzt werden (Kollaboratives Schreiben, E-Collaboration). Ermöglicht wird dies durch ein vereinfachtes Content-Management-System, die sogenannte Wiki-Software oder Wiki-Engine. Die bekannteste Anwendung von Wikis ist die Online-Enzyklopädie Wikipedia, welche die Wiki-Software MediaWiki einsetzt.

10.2 Mediawiki allgemein

MediaWiki ist eine freie Server-basierte Software, die unter der GNU General Public License (GPL) lizenziert ist. Es wurde entworfen, um auf einer großen Server-Farm eine Website zu betreiben, die Millionen Seitenzugriffe pro Tag erhält. MediaWiki ist eine äußerst leistungsfähige, skalierbare Software und eine funktionsreiche Wiki-Implementierung, die PHP verwendet, um Daten zu verarbeiten und anzuzeigen, die in einer Datenbank wie MySQL gespeichert sind. Auf den einzelnen Webseiten wird MediaWikis Wikitext-Format verwendet, so dass Anwender ohne Kenntnisse von XHTML oder CSS sie einfach bearbeiten und gestalten können. Wenn ein Benutzer eine Bearbeitung auf einer Seite anlegt, schreibt MediaWiki es in die Datenbank, aber ohne die vorherigen Versionen der Seite zu löschen, so dass einfache Zurücksetzungen im Falle von Vandalismus oder Spam möglich sind. MediaWiki kann auch Bild- und Multimedia-Dateien verwalten, die im Dateisystem gespeichert werden. Für große Wikis mit vielen Benutzern, unterstützt MediaWiki Caching und kann leicht mit Squid-Proxy-Server-Software gekoppelt werden.

Seiten können leicht geändert werden, es können vorübergehend belanglose Sätze veröffentlicht werden, und eine Seite kann vorübergehend vollständig in einem Wiki zerstören werden. Dazu benötigt man keine Programmierkenntnisse.

Mediawiki ist eine freie Software, deswegen wird seitens der Entwickler, keinerlei Garantie gegeben oder irgend eine Gewährleistung jedweder Art gestellt.

10.3 Worin MediaWiki nicht so gut ist.

Weil es für offene Inhalte entwickelt ist, ist es oft nicht passend für Anwendungen, bei denen man den Zugang zu Teilen des Wikis beschränken will. (Though powering an entirely closed site, l ike an internal company wiki, is not a problem.) MediaWiki ist entworfen, um Seiten mit hohem Zugriff wie Wikipedia zu managen. Es wurde für diesen Zweck optimiert und kann für kleinere Server weniger optimal sein, bei denen Plattenplatz oder RAM größere Einschränkungen sind als Bandbreite. MediaWiki ist keine typische Forum- (BBS-) oder Blog-Software, falls Sie genau das suchen.

10.4 Syntax

```
kursiver Text : ''kursiv''
fetter Text : '''fett'''
fett und kursiv : '''''fett & kursiv'''''
durchgestrichener Text : <strike> durchgestrichener Text </strike>
Überschriften verschiedener Ebenen : == Ebene 2 ==

                                      === Ebene 3 ===

                                     ==== Ebene 4 ====
(hierbei haben die Überschriften unterschiedliche Größen)

Horizontale Linie :  ----

Bullet-Liste :
                * Beginne jede Zeile
                ** Mehr Sternchen ergeben tiefere
                *** und tiefere Einrückung.
                * Zeilenwechsel <br />ändern die Einrückung nicht.
                *** Eine ausgelassene Einrückung erzeugt einen Leerraum.
                Jeder andere Zeilenanfang beendet die Liste.

Nummerierte Liste:
                    # Beginne jede Zeile
                    ## Mehr Zeichen ergeben tiefere
                    ### und tiefere
                    ### Einrückung.
                    # Zeilenwechsel <br />ändern die Einrückung nicht.
                    ### Eine ausgelassene Ebene ergibt einen Leerraum.
                    # Leerzeilen

                    # beenden die Liste und beginnen eine neue.
                    Jeder andere Zeilenanfang beendet die Liste ebenfalls.
```

```
Definitionsliste :
                  ;Begriff 1
                  : Beschreibung 1
                  ;Begriff 2
                  : Beschreibung 2-1
                  : Beschreibung 2-2

Text einrücken :
                  : Einfache Einrückung
                  :: Doppelte Einrückung
                  ::::: Mehrfache Einrückung
```

11 Odoo

11.1 Was ist Odoo - mehr als OpenERP

OpenERP heißt - wie bekannt - seit einigen Tagen Odoo. Was bedeutet das und warum hat es diese Namensänderung gegeben?

11.2 Was bedeutet Odoo?

Soweit wir wissen, hat das Wort Odoo keine wirkliche Bedeutung. Einige behaupten, dass Odoo das Kürzel von On Demand Open Object ist. Andere leiten es - mehr oder weniger direkt – von On Demand (SaaS) Offer from OpenERP ab. Die offizielle Verlautbarung zum Namen Odoo besagt, dass Firmen mit mehr „O“ im Namen einen höheren Marktwert haben und deshalb sind 3 Os ein gutes Omen!

Ganz sicher aber gibt es einen sehr präzisen und praktischen Grund für die Einführung des Namens Odoo, nämlich den Wunsch, die Entwicklung vom einfachen und reinen ERP zu einem umfassenderen Angebot von Dienstleistungen und Tools für Unternehmen zu verdeutlichen. Dieses wird in der Modalität SaaS geliefert bzw. in der Version open source community edition zur Verfügung gestellt. Odoo ist, wie es Fabien Pinckaers, Gründer von OpenErp in einem Post geschrieben hat, eine Suite von Business Apps und kein einfaches ERP.

11.3 Website Builder und e-Commerce: Echte Neuheiten von Odoo.

Wer schon Erfahrung mit den vorhergehenden Versionen von OpenERP haben sollte, dem werden die beiden echten Neuheiten von Odoo auffallen, d.h. der Website Builder (ein integriertes, wenn auch noch ausbaufähiges CMS) und der mit dem ERP integrierte e-Commerce Motor (die Online-Verkäufe sind also direkt mit der Lieferantenverwaltung, den Kunden, dem Einkauf, dem Verkauf, dem Lager, dem Rechnungswesen usw.) verbunden. Alle anderen Funktionen (Projektmanagement, Personalverwaltung usw.) waren schon in OpenERP vorhanden, und zwar als Module (ab einem gewissen Zeitpunkt dann als App, für die Version SaaS), die bei Bedarf installiert werden konnten.

11.4 Odoo wird weiterhin mit Open Source Lizenz ausgegeben.

Odoo kann weiterhin frei genutzt werden. Sie müssen nur das Paket von der Webseite herunterladen, können Odoo aber auch in Modalität SaaS benutzen, wobei Sie direkt die Server von OpenErp SA verwenden (für nur zwei Abnehmer ist die Dienstleistung kostenfrei). In beiden Fällen benutzen Sie exakt die gleiche Software. Die neue Aufgabe von Odoo ist es, dem Unternehmen ein einziges Tool zu geben, das auf alle Firmenbedürfnisse antwortet (ERP aber auch Webseiten

und e-Commerce), ohne verschiedene Software zu benutzen, die untereinander integriert werden müssen.

Wenn man die historische Entwicklung von OpenErp nach Odoo verfolgt, dann hat bei OpenERP die Erneuerung des herkömmlichen ERP schon ab der Version 7.0 begonnen. Dabei wurden Möglichkeit der Zusammenarbeit im Sinne der Social Media eingefügt, um die Leistungen dort zu steigern, wo das Teilen von Geschäftsinformationssystemen (E-Mail, File-sharing...) verhindert wurde.

Im Klartext: Odoo ist genau das, was die OpenERP Version 8 gewesen wäre, allerdings mit den neuen Zusatzmodulen. So hat man also den Übergang zu einer noch offeneren und auf integrierter Zusammenarbeit beruhenden Lösung.

11.5 Hinweis für die Entwickler.

Ein wichtiger technischer Hinweis für die Entwickler: Mit dem Übergang zur Version 8 erfolgt die Transition von Launchpad nach GitHub. Die Terminologie ändert sich: Es wird „master“ anstelle von „trunk“ verwendet. Der Übergang zu GitHub deshalb, weil Git schneller und GitHub besser organisiert ist. Außerdem bleibt Odoo, obwohl dem Namenswechsel das Wort Open zum Opfer gefallen ist, in jedem Fall Open, weil für den Download der verfügbaren Community Edition weiterhin die Lizenz GNU Affero General Public License bzw. GNU AGPL verwendet wird.

11.6 Was ist die Aufgabe von Odoo?

Wenn wir also das „warum“ von Odoo kurz zusammenfassen wollen, dann können wir mit Sicherheit sagen, dass Odoo dem Unternehmen ein einziges und integriertes Tool gibt, das allen Firmenbedürfnissen entspricht (ERP, aber auch Webseitengestaltung und e-Commerce), ohne dass unterschiedliche Software verwendet wird, die anschließend integriert werden muss.

12 GIT Versionskontrollsystem

12.1 Geschichte

Torvalds wünschte sich ein verteiltes System, das wie BitKeeper genutzt werden konnte und die folgenden Anforderungen erfüllte:
1. Unterstützung verteilter, BitKeeper-ähnlicher Arbeitsabläufe
2. Sehr hohe Sicherheit gegen sowohl unbeabsichtigte als auch böswillige Verfälschung
3. Hohe Effizienz
Ein bereits existierendes Projekt namens Monotone entsprach den ersten zwei Anforderungen, das dritte Kriterium wurde jedoch von keinem bestehenden System erfüllt. Torvalds entschied sich dagegen, Monotone an seine Anforderungen anzupassen, und begann stattdessen, ein eigenes System zu entwickeln. Einer der Hauptgründe für diesen Schritt war die Arbeitsweise, für die Monotone nach Torvalds Ansicht optimiert ist. Torvalds argumentierte, dass einzelne Revisionen von einem anderen Entwickler in den eigenen Entwicklungszweig zu importieren zu Rosinenpickerei und „unordentlichen“ Repositories führen würde. Wenn hingegen immer ganze Zweige importiert werden, wären Entwickler gezwungen aufzuräumen. Dazu seien „Wegwerf-Zweige“ notwendig. Gits Gestaltung verwendet einige Ideen aus Monotone sowie BitKeeper, aber keinen Quellcode daraus. Es soll ausdrücklich ein eigenständiges Versionsverwaltungssystem sein.

12.2 Eigenschaften

12.2.1 Nicht-lineare Entwicklung

Sowohl das Erstellen neuer Entwicklungszweige (branching), als auch das Verschmelzen zweier oder mehrerer Zweige (merging) sind integraler Bestandteil der Arbeit mit Git und fest in die Git-Werkzeuge eingebaut. Git enthält Programme, mit deren Hilfe sich die nicht-lineare Geschichte eines Projektes einfach visualisieren lässt und mit deren Hilfe man in dieser Geschichte navigieren kann. Ein Branch stellt nur eine Reference, eine Textdatei mit einer Commit-ID, dar, die in einem Repository im Verzeichnis .git/refs/heads liegt und auf einen bestimmten Commit verweist. Über dessen Elterncommits, lässt sich die Branchstruktur rekonstruieren. Durch diese Eigenschaften lassen sich weiterhin sehr große und effiziente Entwicklungsstrukturen realisieren, bei denen jedes Feature und jeder Entwickler einen Branch oder ein eigenes Repository haben, aus dem der Maintainer dann Commits über Merge oder Cherry-pick (Nutzen einzelner Commits) in den Hauptzweig des Projekts (master) übernehmen kann.

12.2.2 Kein zentraler Server

Jeder Benutzer besitzt eine lokale Kopie des gesamten Repositorys, inklusive der Versionsgeschichte (history). So können die meisten Aktionen lokal und

ohne Netzwerkzugriff ausgeführt werden. Es wird nicht zwischen lokalen Entwicklungszweigen und Entwicklungszweigen entfernter Repositories unterschieden. Obwohl es keinen technischen Unterschied zwischen verschiedenen Repositories gibt, gilt die Kopie, auf die von einer Projekt-Homepage aus verwiesen wird, häufig als das „offizielle Repository“, in das die Revisionen der Entwickler übertragen werden.

12.2.3 Datentransfer zwischen Repositories

Daten können neben dem Übertragen auf Dateisystemebene (file://) mit einer Reihe verschiedener Netzwerkprotokolle zwischen Repositories übertragen werden. Git besitzt ein eigenes sehr effizientes Protokoll, das den TCP-Port 9418 nutzt (git://), allerdings nur zum Fetchen und Clonen genutzt werden kann, also dem Lesen eines Repositorys. Ebenso kann der Transfer über SSH (ssh://), HTTP (http://), HTTPS (https://) oder über (weniger effizient) FTP (ftp://) erfolgen. Die Übertragung in das „offizielle Repository“ eines Projekts erfolgt häufig in Form von Patches.] Für Projekte, die auf Webseiten wie GitHub oder BitBucket gehostet werden, kann eine Änderung einfach durch das Pushen eines Branches vorgeschlagen werden, der dann bei Bedarf durch ein Merge in das Projekt integriert wird.

12.2.4 Speichersystem und Dateiversionierung

Im Gegensatz zu CVS, wo jede Datei eine eigene Revisionsnummer besitzt, speichert Git bei einem Commit das gesamte Verzeichnis ab. Wenn eine Datei in einem Commit nicht geändert wird, ändert sich auch die Checksumme nicht und sie muss nicht nochmals gespeichert werden. Die Objekte liegen im Projekt unter .git/objects.

13 Apache Maven

13.1 Einleitung

Maven ist ein Build-Management-Tool der Apache Software Foundation und basiert auf Java. Mit ihm kann man insbesondere Java-Programme standardisiert erstellen und verwalten. Der Name Maven kommt aus dem Jiddischen und bedeutet so viel wie SSammler des Wissens“.

Abbildung 8: Maven Logo

13.2 Design

Maven basiert auf einer Plugin-Architektur, die es ermöglicht, Plugins für verschiedene Anwendungen auf das Projekt anzuwenden, ohne diese explizit installieren zu müssen.
Die Bandbreite reicht von Plugins, die es ermöglichen, direkt aus Maven heraus eine Webanwendung zu starten, um sie im Browser zu testen, über welche, die es ermöglichen, Datenbanken zu testen oder zu erstellen, bis hin zu solchen, die Web Services generieren. Die dafür nötigen Tätigkeiten beschränken sich häufig nur darauf, das gewünschte Plugin zu ermitteln und einzusetzen.

13.3 Lebenszyklus

Der Standard-Lebenszyklus Maven geht von einem Zyklus aus, der bei der Softwareerstellung häufig durchlaufen wird. Dabei wird nicht unterstellt, dass jedes Softwareprojekt alle Phasen des im Folgenden dargestellten Zyklus verwendet:

- archetype (Scaffolding): Damit kann ein Template für ein Softwareprojekt erstellt werden. Abhängigkeiten werden aufgelöst und bei Bedarf heruntergeladen.
- validate (Validieren): Hier wird geprüft, ob die Projektstruktur gültig und vollständig ist.
- compile (Kompilieren): In dieser Phase wird der Quellcode kompiliert.
- test (Testen): Hier wird der kompilierte Code mit einem passenden Testframework getestet. Maven berücksichtigt dabei in späteren Zyklen, dass Testklassen normalerweise nicht in der auszuliefernden Software vorhanden sind.
- package (Verpacken): Das Kompilat wird – ggf. mit anderen nichtkompilierbaren Dateien – zur Weitergabe verpackt. Häufig handelt es sich dabei um eine Jar-Datei.
- integration-test (Test der Integrationsmöglichkeit): Das Softwarepaket wird auf eine Umgebung (anderer Rechner, anderes Verzeichnis, Anwendungsserver) geladen und seine Funktionsfähigkeit geprüft.
- verify (Gültigkeitsprüfung des Software-Pakets): Prüfungen, ob das Softwarepaket eine gültige Struktur hat und ggf. bestimmte Qualitätskriterien erfüllt.
- install (Installieren im lokalen Maven-Repository): Installiere das Softwarepaket in dem lokalen Maven-Repository, um es in anderen Projekten verwenden zu können, die von Maven verwaltet werden. Dies ist insbesondere für modulare Projekte interessant.
- deploy (Installieren im fernen Maven-Repository): Stabile Versionen der Software werden auf einem fernen Maven-Repository installiert und stehen damit in Umgebungen mit mehreren Entwicklern allen zur Verfügung.

13.4 Unterstützte Entwicklungsumgebungen

Für die gängigsten Entwicklungsumgebungen (z. B. Eclipse, IntelliJ IDEA oder NetBeans) sind Plugins vorhanden, über die sich Maven direkt aus der Entwicklungsumgebung heraus bedienen lässt. Zusätzlich sind Maven-Plugins vorhanden, die Dateien erzeugen, welche den Import eines reinen Maven-Projekts in die Entwicklungsumgebung ermöglichen

13.5 Teilprojekte

Die Entwicklung von Maven ist in verschiedene Teilprojekte untergliedert.

- Maven 1.x pflegt die älteren Versionen von Maven.
- Maven 2 entwickelt die aktuelle Produktlinie von Maven weiter.
- Plugins entwickelt die meisten Maven-Plugins.
- Shared Components stellt Softwarekomponenten bereit, die von den anderen Teilprojekten verwendet werden können.
- Ant Tasks ermöglicht es, Maven-Funktionalität aus Ant-Skripten heraus zu verwenden.
- Doxia ist ein Framework zum Generieren von Content aus den Formaten Almost Plain Text (APT), Confluence, DocBook, FML (FAQ Markup Language), LaTeX, Markdown, Rich Text Format (RTF), TWiki, XDoc und XHTML.
- SCM (Source Code Management) entwickelt Software für die Anbindung von Apache an verschiedene Systeme zur Versionsverwaltung wie CVS oder Subversion.
- Surefire entwickelt ein Testframework für Maven.
- Wagon stellt eine Abstraktionsschicht für Kommunikationsprotokolle wie "Dateizugriff", HTTP oder FTP bereit.

13.6 Versionsgeschichte

Die erste Version, Maven 1.x, wurde im Jahre 2003 eingeführt und am 13. Juli 2004 als Version 1.0 fertiggestellt. Die Umsetzung passierte jedoch sehr schnell, sodass einige Eigenheiten nicht bedacht wurden. Beispielsweise gibt es Performanceprobleme und zu viele Konfigurationsdateien und -angaben.

Deshalb wurde das Konzept überarbeitet und seit dem Jahre 2005 parallel begonnen, Maven 2.x zu entwickeln, welches in Version 2.0 am 19. Oktober 2005 fertiggestellt wurde.

Maven 1.x wird nicht mehr weiterentwickelt und beschränkt sich auf Support und Bugfixes. Die letzte Version von Maven 1.x (Version 1.1) wurde am 25. Juni 2007 ausgeliefert.

Die Entwicklung von Maven 3.0 begann im Jahr 2008. Nach acht Alpha-Releases wurde die erste Beta-Version von Maven 3.0 im April 2010 veröffentlicht. Besonderes Augenmerk lag auf der Kompatibilität zwischen Maven 2 und 3.

14 JIRA

Abbildung 9: JIRA Logo

14.1 Was ist JIRA?

JIRA ist eine von Atlassian entwickelte Projektmanagementsoftware die unter anderem auch Issuetracking und Statusverfolgung bereitstellt. JIRA wird nicht nur im Bereich der Softwareentwicklung verwendet da es auch Anforderungs- und Aufgabenmanagement ermöglicht.

14.2 Geschichte

JIRA wird seit 2002 von Atlassian Inc. entwickelt und betreut. Der Name kommt vom japanischen Wort **gojira** für Godzilla um Bezug auf den bekannten Issuetracker Bugzilla zu nehmen.

14.3 Eigenschaften

JIRA ist in Java programmiert und ist somit plattformunabhängig. Die GUI wird über ein Web-interface bereitgestellt und ist daher leicht erreichbar. JIRA ist nur für offizielle non-profit organisationen und open source Projekte frei verfügbar. Wenn eine Lizenz erworben wird, erhält man eine Kopie des Quellcodes und darf diese für eigene Änderungen verwenden, aber auf keinen Fall weitergegeben oder verkauft werden.

14.4 Lizenzen

JIRA ist für öffentliche Open Source Projekte und non-profit Organisation frei verfügbar. Staatliche sowie politische, bildungs oder religiöse Organisationen müssen die Lizenz allerdings auch kommerziell erwerben.

14.5 Nutzer

JIRA wirbt mit bis zu 25.000 aktiven Nutzern in über 122 Ländern. Bekante Firmen die JIRA verwenden sind Beispielsweise: NASA, BMW, Cisco, Adobe, ...

15 Bugzilla

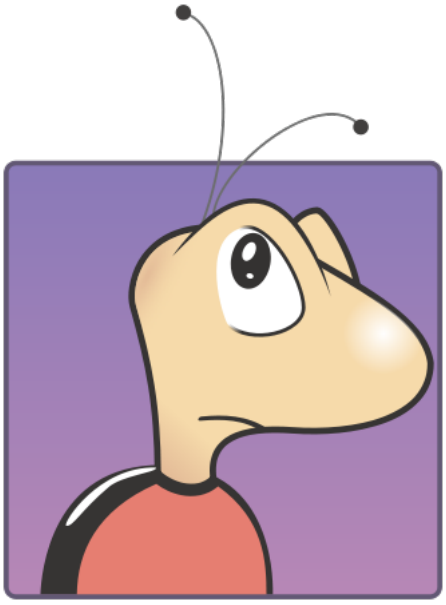

Abbildung 10: Logo von Bugzilla

15.1 Allgemeines

Bugzilla ist ein Bugtracker, welcher zur Dokumentation und Erfassen von Programmfehlern zuständig ist. Der Name setzt sich aus Bug und Mozilla zusammen, was ins Deutsche übersetzt „Käfer Internet-Software“ heißen würde. Das Programm ist in Perl geschrieben und steht unter der Mozilla Public License und ist somit kostenlos. Perl wurde 1987 von Larry Wall entworfen und ist eine Programmiersprache, welche im Gegensatz zu anderen Programmiersprachen viele Freiheiten bietet. Die Mozilla Public License (MPL) ist eine Copyleft-Lizenz. Zur Benutzung von Bugzilla wird ein Webbrowser benötigt, da Bugzilla wie eine Webseite funktioniert und über HTML aufgebaut ist. Sie ist englisch, jedoch können weitere Sprachpakete dazu installiert werden. Es kann zwischen drei verschiedenen Datenbanksystemen gewählt werden: MySQL, PostgreSQL und Oracle. Somit kann Bugzilla auf jedem Betriebssystem laufen, jedoch muss eine Perl-Distribution, ein Webbrowser und ein Datenbanksystem vorhanden sein. Bugzilla wird von viele Open-Source-Projekten verwendet um beispielsweise Fehlermeldungen oder Wünsche von Benutzern zu sammeln.

15.2 Geschichte

Bugzilla wurde von Netscape entwickelt um Softwareverfolgungen auszuführen, wo der Quelltext noch nicht offengelegt war. Seit 2007 gibt es die Version 3.0, wo auch die Unicoce-Unterstützung integriert wurde.

15.3 Funktionsweise

15.3.1 Bugs reporten

Zuerst muss man die Suchbegriffe in das Suchfeld eingeben. Hier findet man nun Bugs, für die bereits eine Lösung gefunden wurde und weist dann auf die alte Frage hin. Kennt man sich bereits etwas besser mit Latex aus, so kann man auf die unübersichtliche Query Page gehen. Wenn man weiß, welche Komponente für den Bug verantwortlich ist, kann man nun beispielsweise auf das OS beschränken oder ein gewisses Programm auswählen, welches zum Bug führte. Falls man einen Bug aber gar nicht finden kann, ist es wichtig dass man den Bug sorgfältig beschreibt und meldet. Dazu gibt es den Bugzilla Helper, der einen hilft den Bug zu melden und einige Hinweise gibt, worauf man achten muss. Wichtig ist hier vor Allem die Version, welche verwendet wird. Zum Unterpunkt „Summary“ kann man „Buzzwords“ hinschreiben. Somit werden die vorher benutzten Suchbegriffe für den Bug auch in diesen hineingeschrieben und vereinfachen später die Suche. Nun soll man eine genaue Problembeschreibung angeben und es soll selbst reproduzierbar sein, damit der Helfer genauestens Bescheid weiß. Beim letzten Schritt ist anzugeben, was genau passiert ist und was eigentlich passiert hätte sollen. Je besser der Report beschrieben wird, umso leichter kann er bestätigt werden.

15.3.2 Bearbeiten anderer Bugs

Da man bei Bugzilla einen Account erstellen muss, darf man meistens nur die eigens gemeldeten Bugs bearbeiten. Einige der anderen Nutzer können mehrere ändern und man muss erst an einigen Bugs mitarbeiten um alle bearbeiten zu können.

15.3.3 Selber gemeldete Bugs

Sobald man einen Bug nun gemeldet hat erhält er den Status UNCONFIRMED. Einige Helfer schauen meistens so gut wie möglich zu helfen: Verbesserungen an der Zusammenfassung des Problems vornehmen, andere Komponenten angeben, usw. Sobald er dann von einem dieser bestätigt wird erhält er den Status NEW. Wenn jetzt jemand das Problem behandelt, bekommt der Bug den Status ASSIGNED. Falls der Bug nicht reproduzierbar ist, bekommt er den Status WORKSFORME und wenn der Bug schon bereits gemeldet wurde, so erhält er den Status DUPLICATE. Es kann auch sein, dass es kein Bug ist. Somit wird der Status zu INVALID. Wenn einige Bugs nicht behoben werden wollen, so bekommen sie den Status WONTFIX. Sobald man die Lösung bestätigt,

wird der Status von RESOLVED zu VERIFIED. Wenn jemand die Meldungen nicht genau versteht bekommt man meistens Fragen. Diese bekommt man weil beispielsweise, weil der Helfer nicht genau versteht, wie der Fehler zustande kommt.

15.3.4 Bugs, die andere Nutzer gemeldet haben

Manchmal kann man auch zu anderen Bugs etwas beitragen, jedoch sollten diese Beiträge hilfreich sein. Hier kann z.b. helfen, dass man den Bug mit einer anderen Version nachvollziehen oder auch nicht nachvollziehen kann.

16 IRC

Internet Relay Chat (oder kurz IRC) bezeichnet ein vollkommen textbasiertes Chat-System. Hier treffen sich die Teilnehmer in sogenannten Channels und können in diesen Channels miteinander schreiben. Jeder Benutzer kann jederzeit einen neuen Channel erstellen. Außerdem ist es möglich sogenannte Querys zu erstellen, welche die Kommunikation zwischen nur zwei Gesprächsteilnehmern ermöglicht.

16.1 Geschichte

Die erste Version eines solchen Chat-Netzwerks entstand 1985 im BITNET, einem Netzwerk von Großrechnern in den USA. Damals noch unter dem Namen Relay Chat. Zwei Studenten von der Universität Oulu, in Finnland übertrugen dieses System 1988 auf das Internet. Die erste Version dieses Chat-Netzwerks erlangte schnell Beliebtheit, sodass bald weitere Unabhängige IRC Netze entstanden.

16.2 Allgemein

Ein IRC-Netzwerk besteht aus mehreren miteinander verbundenen Servern, welche auch Relay-Station genannt werden. Jeder Chat Teilnehmer verbindet sich immer mit einem Server und klingt sich somit in das Netz ein. Ein besonderes Merkmahl des IRC-Netzwerks ist, dass zwischen zwei Teilnehmern des Chats immer nur eine Verbindung besteht. Dies war historisch sehr bedeutsam, da früher die Leitungskapazität sehr begrenzt war. So wird jede Nachricht die von den Benutzern, die mit einem Server verbunden sind, nur über einen Kanal an einen weiteren Server und somit an deren Teilnehmer geschickt. Dieser Server schickt seine Nachricht daraufhin an den nächsten Server weiter. Diese Funktionsweise sorgt für eine sehr geringe Menge an Datenverkehr, jedoch fehlt gleichzeitig jede Art von Redundanz. So kann es beim Ausfall eines Servers passieren, dass sich ein Netz in zwei kleinere Netze aufspaltet zwischen denen keine Verbindung mehr besteht.

16.3 Technische Eigenschaften

IRC ist ein auf IP und TCP basierendes Protokoll. Die gesamte Kommunikation besteht aus Textnachrichten mit einer maximalen Länge von 512 Zeichen. Diese Textnachrichten enthalten einen Absender, einen Befehl und je nach Befehl eventuelle Befehlsparameter. Auf einen Befehl kann eine Antwort des Servers folgen, dies ist jedoch von Befehl zu Befehl und von Server zu Server unterschiedlich. Im Laufe der Zeit entwickelten sich außerdem verschiedene Erweiterungen zum IRC Protokoll, welche unter anderem den Aufbau der Nachrichten ändern.

16.4 Verschlüsselung

Ob und wie Nachrichten verschlüsselt werden ist von Netz zu Netz unterschiedlich. In den meisten Fällen wird entweder gar nicht verschlüsselt oder eine über SSL/TLS verschlüsselte Verbindung benutzt. Verschiedene Erweiterungen benutzen hier ebenfalls verschiedene Verschlüsselungen.

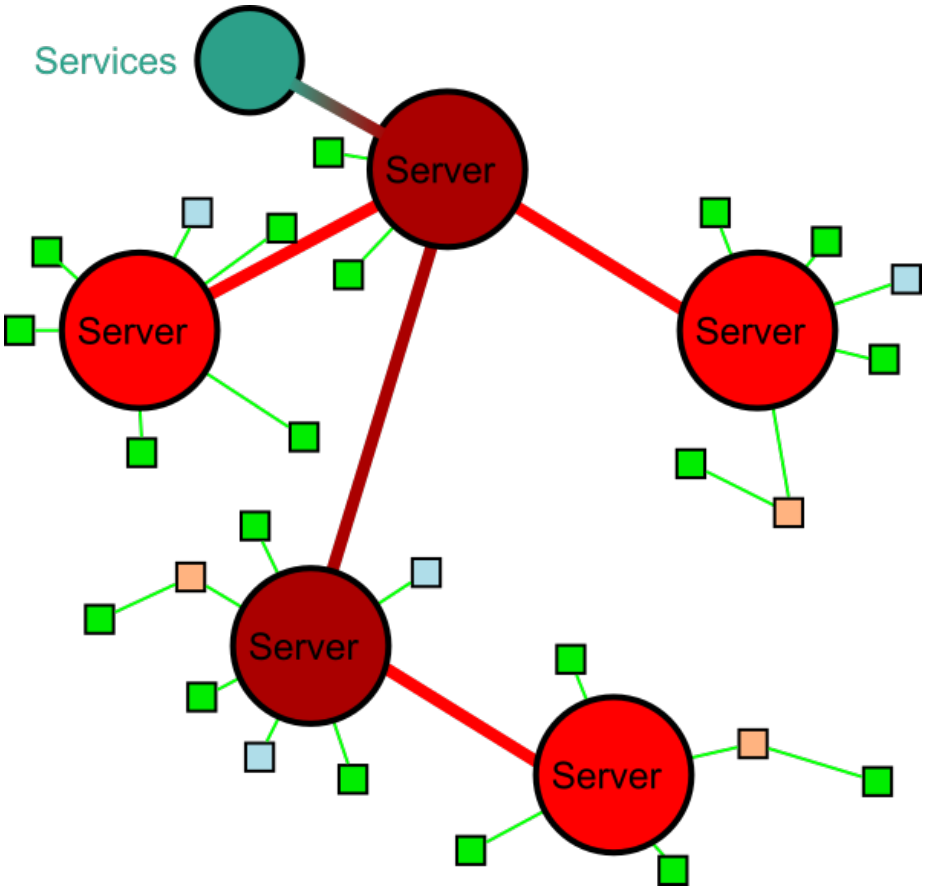

Abbildung 11: Schematischer Aufbau eines IRC-Netzes.

17 SourceForge

17.1 Allgemeines

SourceForge ist eigentlich ein englisches Wort und bedeutet soviel wie „Quellentextschmiede“. Es ist ein Repository in Form einer Website. Mit SourceForge ist es möglich Softwareprojekte online zu erstellen und zu verwalten. Was dabei zu beachten ist, ist, dass eine Registrierung optional ist. Dass bedeutet, man kann öffentliche Projekte auch ohne einen Account einsehen. SourceForge wird betrieben vom US-amerikanischen Unternehmen „Dice Holdings“. Außerdem war es bis zur Version 3 eine freie Software, wurde daraufhin allerdings proprietär und kommerziell vertrieben. Viele bekannte Open Source Projekte sind auf SourceForge vertreten, wie zum Beispiel Hibernate, Moodle oder auch FileZilla. Des Weiteren bietet SourceForge im Gegensatz zu vielen Konkurrenten, mehr als nur ein System zur Versionsverwaltung an.

Dazu zählen:

- CVS
- SVN
- Bazaar
- Git
- Mercurial

SourceForge wird von vielen Software Entwicklern genutzt, weil es neben der Hauptaufgabe als Online-Repository noch weitere Vorteile bietet. Beispielsweise kann jedes Projekt über ein eigenes Wiki verfügen, in dem wichtige Informationen über das Projekt aufgelistet sind. Ein anderer Punkt ist, dass SourceForge für jedes einzelne Projekt auch die Möglichkeit bietet, auf eine eigene MySQL-Datenbank zuzugreifen.

17.2 Einschränkungen

Der Zugriff auf SourceForge wurde vor einiger Zeit in China gesperrt, aufgrund des Projektes „Goldener Schild“ . Das soll eine Anlehnung an die Chinesische Mauer sein. Dieses Projekt ist für die überwachung der Zensur im chinesischen Internetverkehr zuständig. Allerdings wurde diese Sperrung nach bereits einem Jahr wieder aufgehoben. 2008 wurde der Zugang aber wieder gesperrt, weil vermutet wird, dass sich ein SourceForge-Programmierer negativ über die chinesische Regierung geäußert hat.

Des Weiteren wurde SourceForge in einigen sogenannten „Schurkenstaaten“ gesperrt. Das sind Staaten, die auf der Sanktionsliste der USA sind. Einige betroffene Länder sind Nordkorea, Syrien, Sudan und Kuba. Aufgrund der Reaktion der Community wurde diese Einschränkung aber ein wenig gelockert. Es ist nun möglich, dass der Administrator jedes Projektes selbst bestimmen kann, ob er den Zugang zum Projekt in diesen Ländern freigeben möchte.

17.3 Kritikpunkte

In letzter Zeit wurde SourceForge stark kritisiert, weil sie auf sogenannte „Drive-By-Installer“ zurückgriffen. Diese sorgen dafür, dass bei der Installation von SourceForge weitere Installationen von Adware von Drittanbietern vorgeschlagen wird. Ein Beispiel für eine solche Adware ist die „Ask-Toolbar“. Das ganze wurde wahrscheinlich gemacht, um an zusätzliches Geld zu kommen, allerdings war die Community damit nicht zufrieden und deswegen hat SourceForge in letzter Zeit stark an aktiven Mitgliedern verloren.

17.4 ähnliche Projekte

SourceForge war eine Art Vorreiter auf dem Gebiet von Online-Repositories. Heute gibt es allerdings jede Menge Konkurrenten, die teilweise auch schon erfolgreicher sind als SourceForge. Dazu zählt Github, ein Online-Repository, dass mit dem Versionsverwaltungssystem Git arbeitet. Weitere Beispiele sind Bitbucket, dass Mercurial zur Versionsverwaltung benutzt, Codeplex, was eine Hosting Website von Microsoft für Open Source Projekte ist und Freecode, ein Katalog für Open Source Projekte.

18 Mercurial

Abbildung 12: Logo von Mercurial

Mercurial ist ein verteiltes, programmunabhängiges Versionskontrollsystem.

18.1 Geschichte

Angekündigt wurde es von Matt Mackall auf der Linux-Kernel-Mailingliste am 19. April 2005. Diese Ankündigung war die Folge, dass die Firma BitMover, die z. B. für den Linux-Kernel als Versionskontrollsystem eingesetzte Software BitKeeper nicht mehr in einer kostenlosen Version bereitstellte. Zur selben Zeit startete Linus Torvalds das Projekt Git, das sich in der Folgezeit besser etablieren konnte als Mercurial.

18.2 Implementierung

Es ist nahezu in vollständig in Python entwickelt, weshalb es nicht unbedingt empfehlenswert wäre, Mercurial zu benutzen. Lediglich eine diff-Implementierung, die mit binären Dateien umgehen kann, ist in C entwickelt, da das in Python nicht möglich ist. Nichtsdestotrotz sind Effizienz, Skalierbarkeit und robuste Handhabung von Text- und Binärdateien einige Punkte, die bei der Entwicklung als Schwerpunkte festgelegt wurden.

Ähnlich wie Git ist Mercurial kein zentralisiertes Versionskontrollsystem, d.h. man kann ein Repository klonen kann und auf einer lokalen Kopie darauf arbeiten. Auf dieser Kopie kann man ganz normal auf den die Funktionen von Mercurial verwenden. Ebenso ist die Fähigkeit des Erstellens und Zusammenfügens von Entwicklungszweigen ein fester Bestandteil dieses Versionskontrollsystems. Des Weiteren kann man einfach und schnell Unterschiede zwischen zwei unterschiedlichen Versionen anzeigen lassen. Zudem ist es möglich bei der Version Sekunden anzugeben, die man zurückspringen will.

18.3 Zugriff

Man kann sowohl über eine grafische Oberfläche als auch über die Komandozeile auf die Funktionen von Mercurial zugreifen. Eine grafische Oberfläche wird häufig bei Microsoft Windows, Gnome/Nautilus (jeweils TortoiseHg) und bei Mac OS X (MacHg und Murky). Bei gängigen Entwicklungsumgebungen wie Netbeans, Eclipse, Android Studio oder der Qt Creator erlauben es, ein externes Plugin zu installieren und das User-Interface der Entwicklungsumgebung ist es möglich, auf die Funktionen von Mercurial zuzugreifen.

18.3.1 Terminal

Über die Kommandozeile kann man über folgende Befehle auf Mercurial zugreifen:

- **Clonen**: hg clone <URL>
- **Dateien hinzufügen**: hg add <Datei>
- **Änderung**: hg revert <Datei>
- **Änderungen bestätigen**: hg commit -m <Änderungstext>
- **Repository auf den aktuellen Stand bringen**: hg update
- **Branch mischen**: hg merge
- **Versionsgeschichte des Repository erkunden**: hg log -v

Wie beim bereits im Unterricht behandelten GIT, muss man bei der Bestätigung von Änderungen eine Beschreibung der Änderung anfügen. Wenn das automatische Mischen nicht gelingt, muss man manuell die Konflikte lösen.

18.4 Anwendungs-Beispiele

Einige namhafte Firmen haben und bei bekannten Projekten wurde Mercurial eingesetzt. Dazu gehören Facebook, Mozilla (Firefox, Thunderbird), SourceForge, Google Inc. (Google Chrome, Google Code), Atlassian (Bitbucket), Microsoft (Codeplex), Oracle (OpenJDK), Xen, NetBeans IDE, Python und ClearCanvas.

18.5 Fazit

Zusammenfassend kann man sagen, dass Mercurial ähnlich dem Versionskontrollsystem Git ist, es gibt keine signifikante Unterschiede. Der größte Unterschied zwischen beiden Systemen ist der Performanceverlust durch den Python-Interpreter, weshalb es besser ist, Git als Versionskontrollsystem zu verwenden.

19 Bitbucket

Bitbucket ist ein webbasierter Hosting-Dienst für Software-Entwicklungsprojekte, der die Versionsverwaltungssysteme Git und Mercurial unterstützt.

19.1 Geschichte

Der Dienst wurde ursprünglich als reines Mercurial-System von Jesper Nøhr entwickelt und 2010 von dem australischen Unternehmen Atlassian gekauft und am 3. Oktober 2011 um Unterstützung für Git erweitert.

19.2 Eigenschaften

In Bitbucket wir nicht das Projekt als Sammlung von Quelltexten verwaltet, sondern die Repositorys der Nutzer. Gleichzeitig wird das Erstellen (Branchen) und Wiedervereinigen (Mergen) von Abspaltungen (Forks) besonders propagiert. Bei Bitbucket ist es wie gesagt möglich, den eigenen Code zentral zu verwalten und zu versionieren. Dabei stehen zwei Versionierungs-Plattformen für die eigenen Repositories zur Verfügung: Git und Mercurial. Zusätzlich ist auch das Folgen von anderen Bitbucket-Nutzern und Teams möglich.

Bitbucket ist ein Web-basierter Online-Dienst, d.h. er läuft auf einem Browser und damit überall. Die Repositories befüllt man wie gehabt mit Git oder Mercurial von einem lokalen System aus. Bitbucket bietet zudem dank einer API auch verschiedene Apps an. Auf Android existiert beispielsweise die App Bitbeaker.

19.3 Kosten

Bitbucket hat ein Premium-Modell. Die Basis-Nutzung ist kostenlos. Diese Basis-Nutzung umfasst eine beliebe Anzahl an öffentlichen und privaten Repositories mit allen dazu nötigen Features, wie Issues, Wikis etc. Teams können aus bis zu 5 Mitgliedern bestehen, welche sich Repositories teilen. Diese Anzahl kann man bis auf 8 Mitglieder erhöhen. Wer mehr Mitglieder in seinem Team verwalten möchte, muss den Dienst bezahlen. Bei 10 Nutzern sind es 10$/Monat, bei 25 Nutzern sind 25$/Monat usw. Beim Unlimited-Plan bezahlt man 200$/Monat.

19.4 Bitbucket vs Github

- Bitbuckets offensichtlichster Vorteil ist die Möglichkeit, kostenlos eine unbegrenzte Zahl an privaten Repositories mit bis zu fünf Beteiligten zu hosten.
- Bitbucket unterstützt nicht nur Git sondern auch Mercurial.

- In Github ist es möglich öffentliche kostenlose Repositorys zu erstellen.

19.5 Verwendungen

2014 arbeiteten über 330.000 Teams aus über 2,5 Millionen Entwicklern mit Bitbucket, 200 Terabyte Code wurden gehostet. Zu den Unternehmen, die Gebrauch von Bitbucket machen, zählen neben Atlassian:

- DHL
- PayPal
- Teslaa Motos
- The New Yorks Times

Auch die Open-Source-Projekte Eigen und OGRE sind bei Bitbucket gehostet.

19.6 Fazit

Bitbucket ist ein webbasierter Hosting-Dienst wie Github. Im grunde genommen ergänzen sich die beiden Programme jedoch recht gut, den bei Bitbucket kann man unbegrenzt viele private Repositorys erstellen und bei Github hingegen kann man unbegrenzt viele öffentliche Repositorys erstellen.