

Web-Engineering, Web- Anwendungssysteme und Mobile Computing III

Eine Vorlesung mit integrierter Übung im Modul
„Aufbau und Nutzung von Internettechnologien
(IT)“
des Studienganges
„Crossmedia Management“

Dr. Sebastian Tramp

” *Internettechnologien bilden die Basis crossmedialer Strategien. Die Studierenden erlernen diese komplexen Technologien und erlangen **Kompetenzen zum technischen Design** multifunktionaler Web-Systeme und erweitern ihr Wissen über **Einsatzmöglichkeiten wichtiger Klassen von Web-Anwendungssystemen**. Nach erfolgreichem Abschluss des Moduls kennen die Studierenden die technischen Grundlagen der Internettechnologien. Sie verfügen über **Grundwissen im Bereich Softwaretechnologien für Webanwendungen und Mobile Computing**. Sie können das Erlernte **selbständig anwenden**.*

- Dozentenplan_IT-2_Tramp

” *Baustein IT 2 vermittelt die Softwaretechnik für die Entwicklung von Web-Applikationen. Im Rahmen des Web-Engineering werden **relevante Standards und Technologien** vorgestellt. Darüber hinaus werden **wichtige Klassen von Web-Anwendungssystemen eingeführt**. Dazu zählen insbesondere ECommerce-Systeme, Virtuelle Welten, Lehr- und Lernsysteme. Ein weiterer Schwerpunkt ist die Vorstellung von **Technologien zur Entwicklung ubiquitärer Anwendungen** wie Apps und Informationssystem für mobile Endgeräte, Tablet-PCs etc.*

- <https://www.leipzigschoolofmedia.de/.../aufbau-und-nutzung-von-internettechnologien.html>

” ***Prüfungsleistung:** Projektarbeit; Bearbeitungszeit: 4 Wochen;
Themenausgabe: 03.03.2018; Abgabetermin: 31.03.2018*

18.01.2018

- The Web / Big Picture / HTTP (Wie funktioniert das Web)
- HTML(5) (Wie erstellt man Webseiten)
- CSS (Wie styled man Webseiten)

08.02.2018

- PHP / Datenbanken
- Apache / PHP / WordPress Installation
- Management von Software-Projekten

03.03.2018

- **JavaScript**
- **Semantic Web Primer, schema.org**
- **Software Engineering Methoden**
- **Magento**
- **Entwicklung für Mobile Devices**

- [LSoM Slides \(PDF\)](#).
- Online Buch: [JavaScript: Browserübergreifende Lösungen](#) von Christian Wenz
- Referenzen: [Mozilla](#), [SelfHTML: JavaScript](#), [Kontrollstrukturen](#)
- Tutorials: [W3 Schools](#)
- jQuery: [Dokumentation](#), [API Browser](#), [jQuery UI](#)

- Resource Description Framework
 - Statement: Subject - Predicate - Object
 - Resources (IRIs) and Literals (strings + datatype + optional language tag)
- Schema and Ontology Languages
 - RDF Schema
 - Web Ontology Language (OWL) 2
- Vocabularies / Ontologies: schema.org, Goodrelations, ...



In: [wikipedia](#)

- Suchmaschinen dominierter Versuch, eine 'Ontologie für das Web' zu erstellen
- Logischerweise, kein long tail, aber wichtige erste Konzepte



- Produkte, Events, Ratings, Offers, Person, ... and more
- Hauptvorteil: Google, Bing, Yahoo und Yandex haben sich dieser Ontologie committed

- Microdata ist eine Sammlung von HTML5 Tags um Meta Daten in HTML Seiten zu integrieren.
- Microdata steht damit in Konkurrenz zu RDFa, dem W3C Standard Web Standard mit gleichem Ziel
- <http://www.w3.org/TR/microdata/>
- Microdata gilt als deutlich weniger komplex in der Benutzung

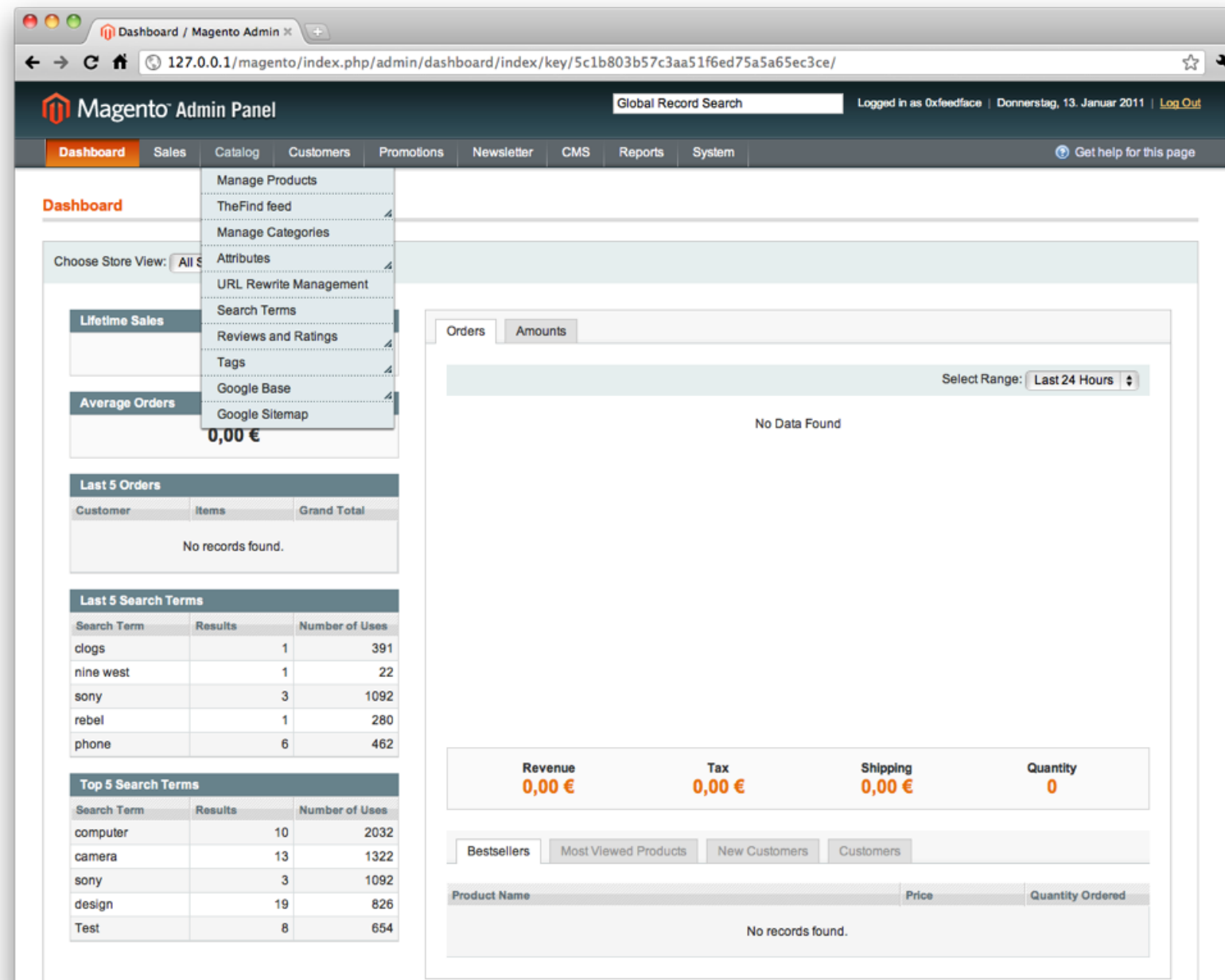
Wichtige Tags sind `itemscope`, `itemtype` und `itemprop`

```
<!-- Hinweis: Das itemscope Attribut startet ein neues Subjekt -->
<!-- Hinweis: Mit itemtype wird einem Subjekt eine Klasse zugeordnet -->
<p itemscope itemtype="http://schema.org/Person">
  Guten Tag, mein Name ist
    <!-- Hinweis: Mit itemprop werden Attribute zugeordnet -->
    <span itemprop="name">Sebastian Tramp</span>.
</p>
```

- [Google Structured Data Testing Tool](#) - Referenz für das Praktikum
- [Google Search Gallery](#) - shows how structured data can produce rich results in Google Search
- [Yandex Structured data validator](#)

Beispiel 2 zeigt, wie zusätzlich zu einfachen Literal Werten, ein benannter Link erzeugt wird (copyrightHolder), welcher eine Klasse und einen Namen bekommt.

```
<article itemscope itemtype="http://schema.org/Article">
  <h1 itemprop="name">Fahnder suchen ...</h1>
  <p itemprop="text">Lorem ipsum dolor sit amet, ...</p>
  <p>
    copyright <span itemprop="copyrightYear">2014</span> by
    <!-- Hinweis: Hier wird ein neues Subjekt angelegt welche
      Objekt von copyrightHolder ist -->
    <span itemprop="copyrightHolder" itemscope
      itemtype="http://schema.org/Organization">
      <!-- Hinweis: Das attribut name bezieht sich jetzt schon
        auf das neue Subjekt -->
      <span itemprop="name">Spiegel</span>
    </span>
  </p>
</article>
```



1. Präsentation, 2. Installation

Prinzipien, Methoden und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Softwaresystemen (aus [Balzert](#))

Dabei **u.a.** folgende Teilaufgaben wichtig:

- Anforderungserfassung
- Spezifikation
- Management und Qualität

- Trennung von funktionalen und nicht-funktionalen Anforderungen
- Möglichst formale Repräsentation wählen - möglichst wenig prosa
- Einzelne Anforderungen müssen identifizierbar sein
- Je besser die Erfassung der Anforderungen, desto besser das Verständnis des Projekts

Anforderungen welche die Funktionalität oder Dienste eines Produktes beschreiben.

Möglichst genaue Beschreibungen mit vielen Meta-Daten, welche insbesondere W-Fragen beantworten. Anforderungen können auch hierarchisch geordnet sein.

- Der Nutzer soll die gesamte Webseite durchsuchen können.
- Die Suchergebnisse sollen highlights der gesuchten Wörter enthalten.
- Gefunden Produkte sollen zuerst und mit Preis dargestellt werden

Mögliche Attribute: Nutzerkategorie, Eingabe, Ziel, ... Priorität, Risiko(!)

Achtung: Anforderungen verändern sich ständig, da sich auch das Verständnis für das Projekt ändert.

Anforderungen, welche spezielle Funktionen der Software **nicht** betreffen.

- Produktanforderungen: Usability, Zuverlässigkeit, Effizienz, ...
- Unternehmensanforderungen: Lieferung, Umsetzung, Vorgehensweise
- Externe Anforderungen: Kompatibilität, Rechtliches wie Datenschutz, Gesetze, ...

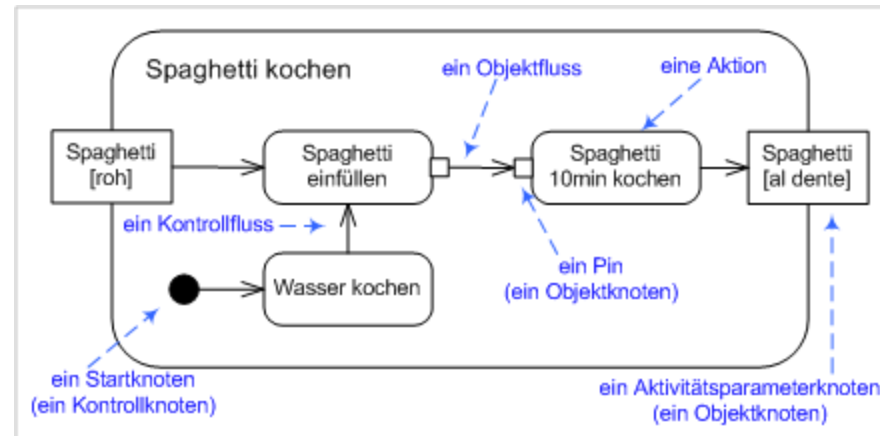
Formalisierte aber trotzdem natürlich-sprachige Anforderungserfassung. Häufige Form:

Als --Rolle-- möchte ich --Ziel/Wunsch--, um --Nutzen--

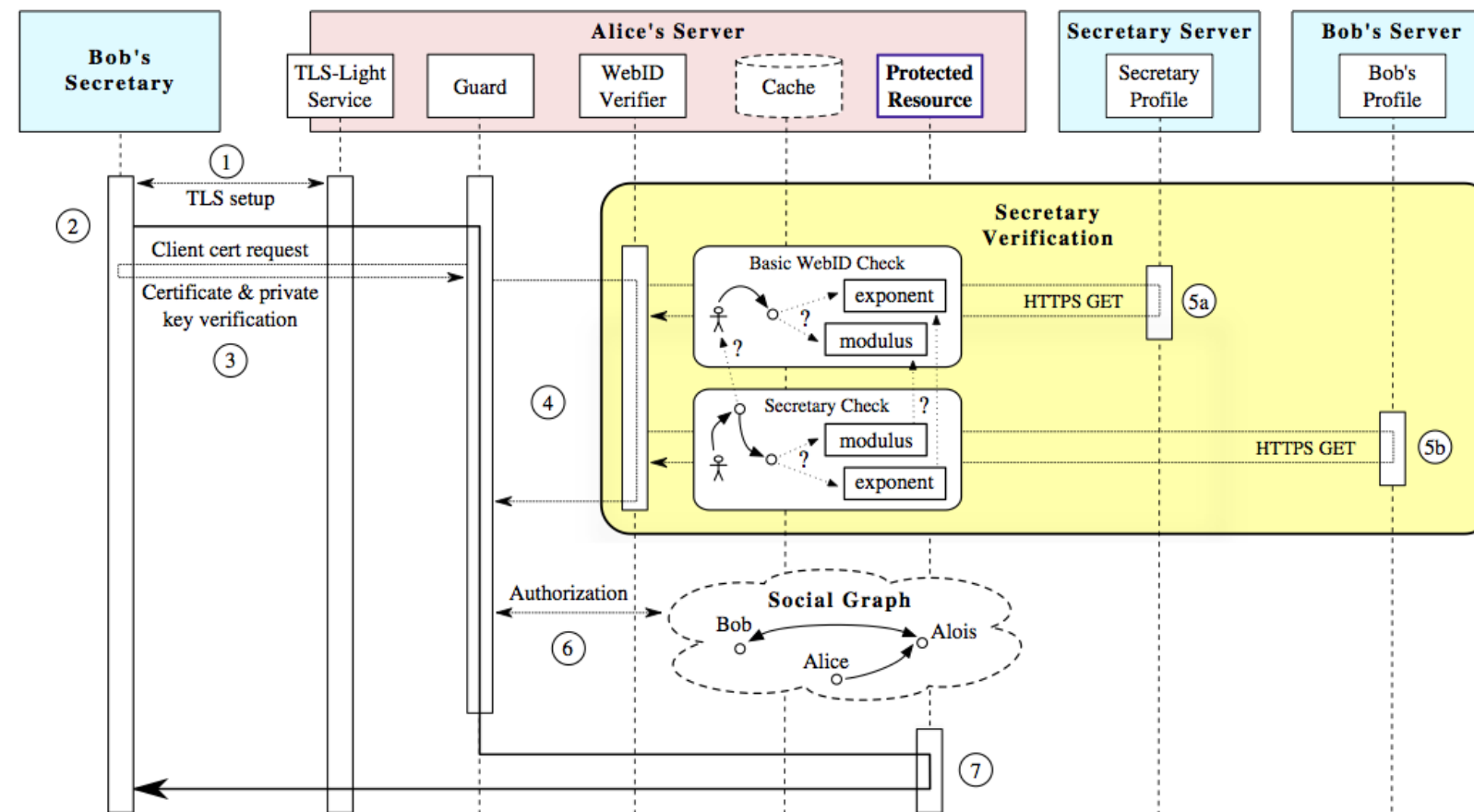
Als --Besucher der Webseite-- möchte ich --meine Suchbegriffe hervorgehoben haben--, damit --ich mich besser zurecht finde--.

Vorteil: schnelle Erfassung von Anforderungen, gut für erste Phase der Anforderungserfassung.

- Deutlich längere Variante einer User Story
- Stark Dokument-orientiert, mit Historie versehen
- Formale Verhaltens-Diagramme: Aktivitätsdiagramme, Sequenzdiagramme u.a. aus der Unified Modeling Language



Source: [wikipedia](https://en.wikipedia.org/wiki/Activity_Diagram)



In: [S.Tramp: Access Delegation](#)

- Helmut Balzert: [Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb](#)
- Ian Sommerville: [Software Engineering](#)
- Frederick P. Brooks: [The Mythical Man-Month. Essays on Software Engineering](#) - "Der Einsatz zusätzlicher Arbeitskräfte bei bereits verzögerten Softwareprojekten verzögert sie nur noch weiter."

Idee: Anforderungen und konkretes Projekt-Management wird in issues / Fälle geteilt und mit einem issue tracker verwaltet. Unmittelbare Vorteile der Nutzung eines Issue-Trackers:

- Qualitätssteigerung
- Transparenz
- Verfügbarkeit der Historie
- Dokumentation für folgende Projekte

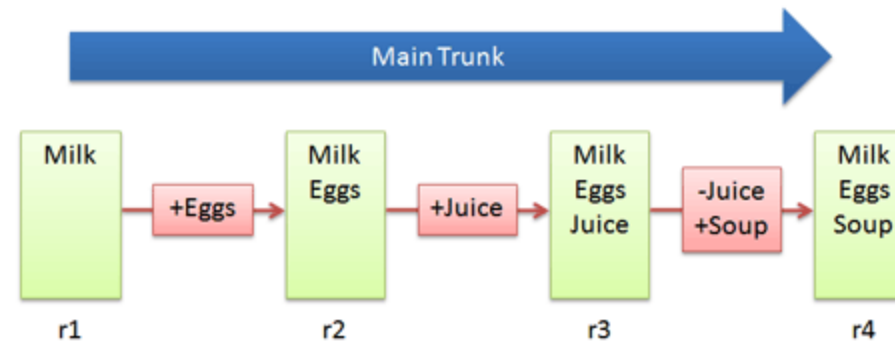
Freie ITS: gitlab, bugzilla, mantis, trac

Kommerzielle ITS / Services: JIRA, github, ...

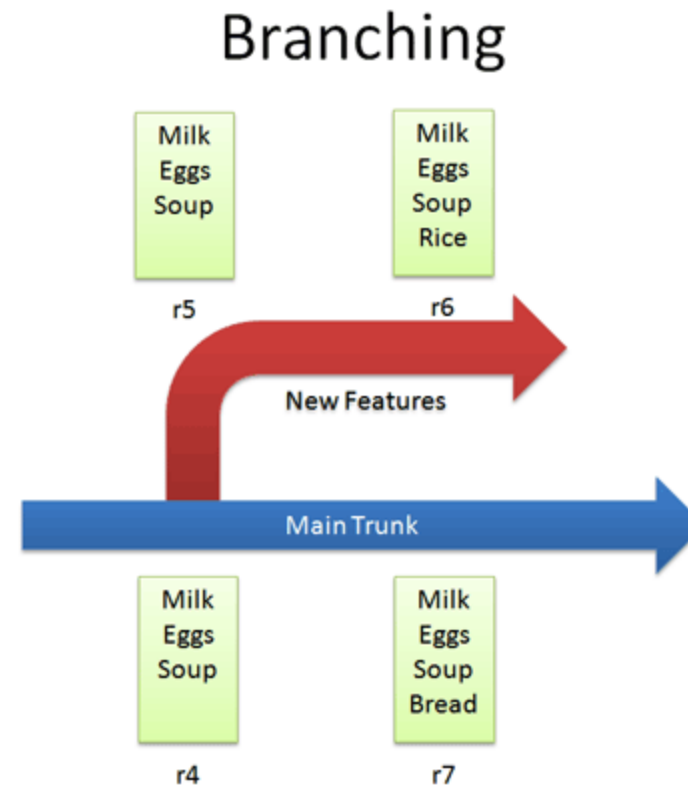
Software zur Verwaltung von Änderungen am Quellcode eines Software-Projektes.

- Wichtigstes Werkzeug bei der Entwicklung von Software
- Transparenz der Entwicklung
- Verantwortlichkeit und Dokumentation für Änderungen
- Idealerweise mit Issue-Tracker gekoppelt

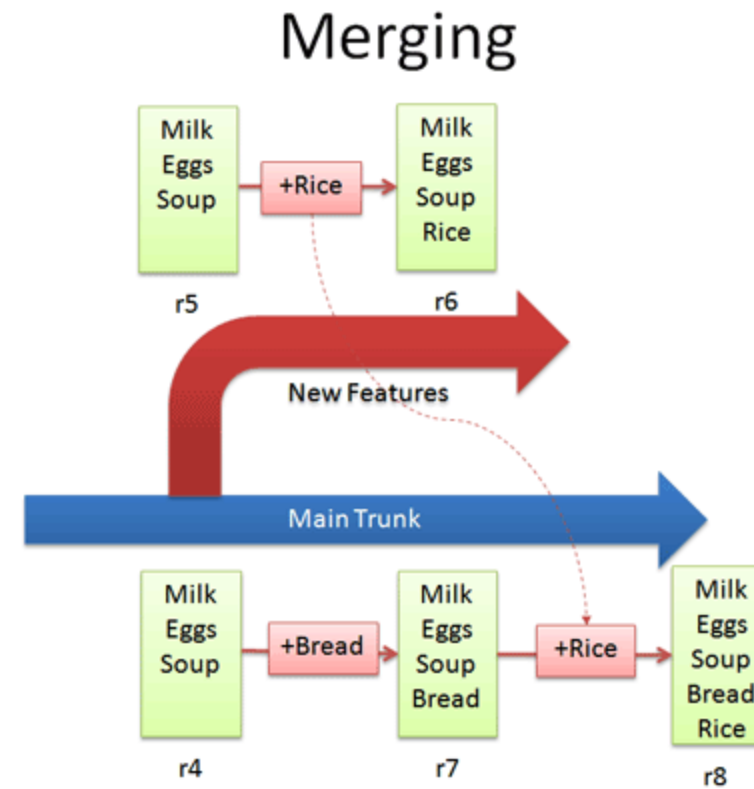
Basic Diffs



In: [A Visual Guide to Version Control](#)

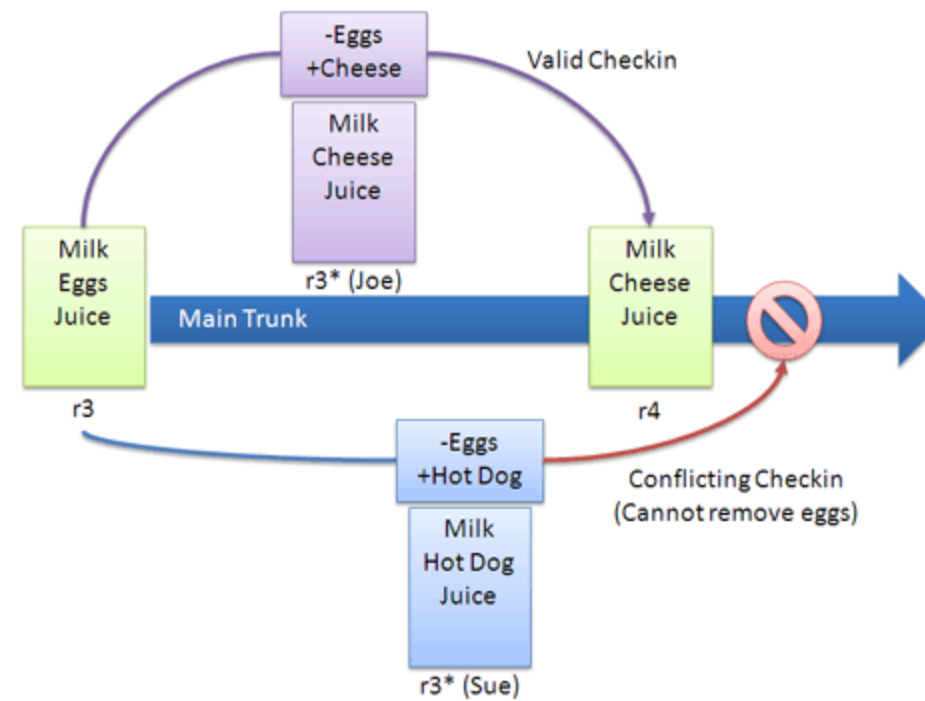


In: [A Visual Guide to Version Control](#)

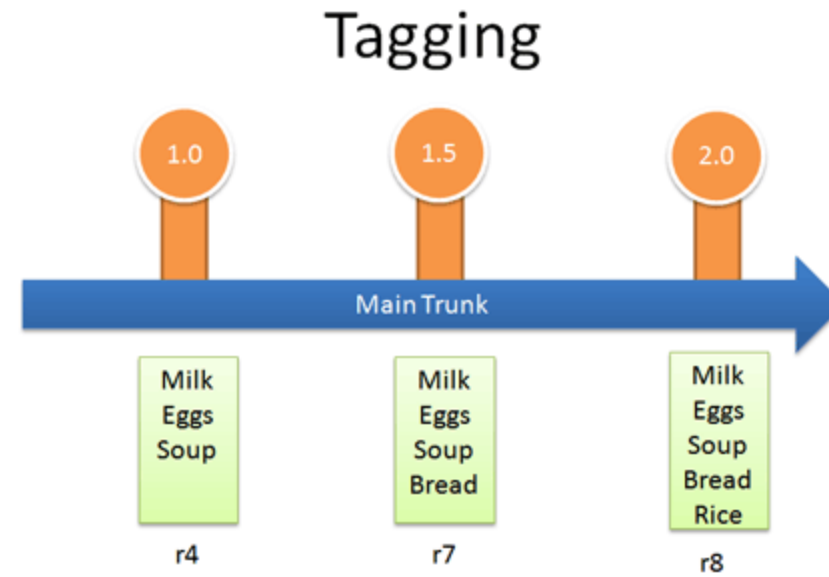


In: [A Visual Guide to Version Control](#)

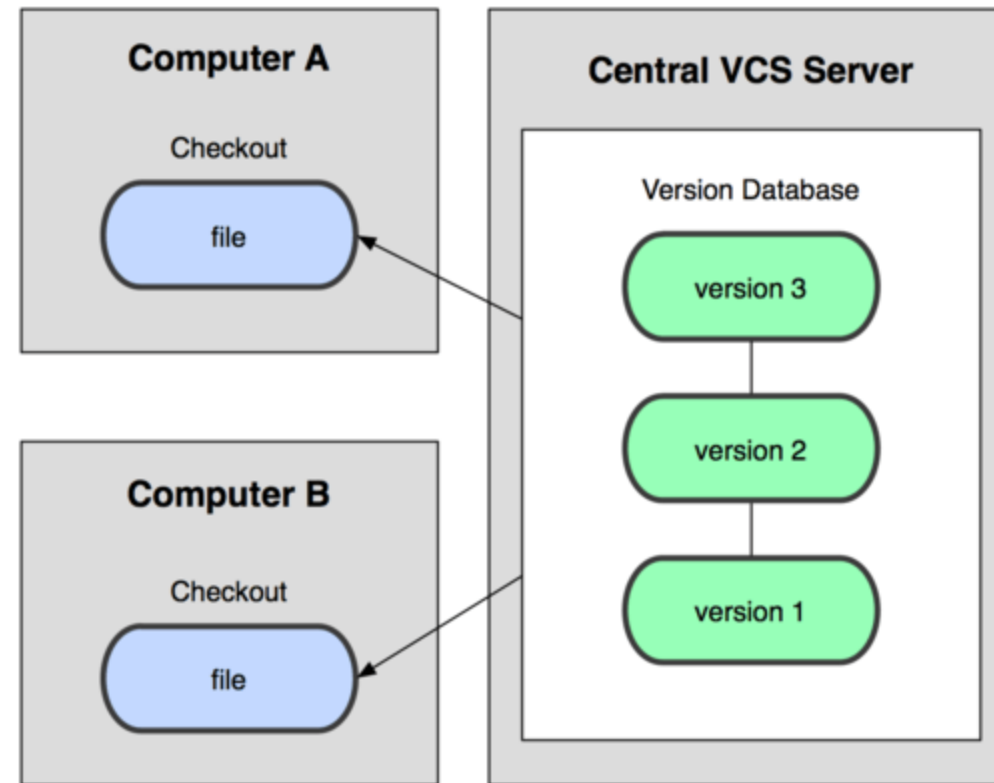
Conflicts



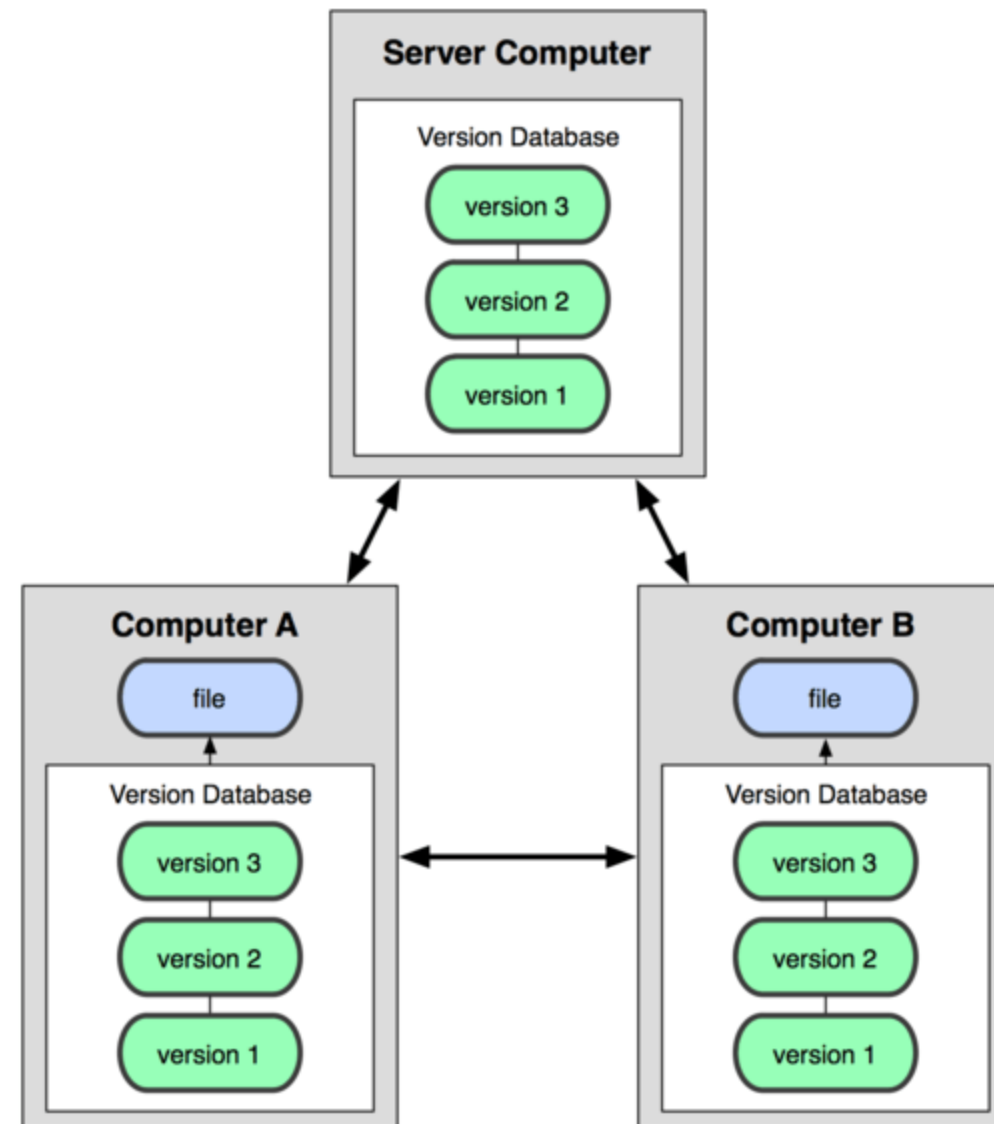
In: [A Visual Guide to Version Control](#)



In: [A Visual Guide to Version Control](#)



In: [git book](#)



In: [git book](#)

- Tests sind ein Werkzeug zur automatischen Überprüfung von Software
- Tests werden parallel zur Software mit entwickelt (z.T. davor)
- Kategorien: Funktionstests, UI Tests, Integrationstests, Lasttests ...

- Gemeinsame Codebasis mit häufiger Integration
- Kontinuierliche Test-Entwicklung und automatisierte Ausführung der Tests
- Automatisiertes Reporting und Verteilung
- Reporting schließt Tests und Code Coverage Reports mit ein

Vorstufe: Daily Builds (einmal pro nacht)

Resources

- [Media Queries](#)
- jquerymobile.com
- [Presentation](#) by Mark Dalgleish
- [Demo center for latest Version](#)
- [API Documentation](#), esp. [Data Attribute Reference](#) and [Plattform support list](#)
- [jQuery Learning Center: jQuery Mobile Chapter](#)

Im Praktikum soll jede_r Student_in das erlernte Wissen um relevante Standards des Web-Engineerings praktisch anwenden.

Folgende Technologien sollen dabei benutzt werden:

- HTML5 und insbesondere Semantisches Markup
- CSS3 und insbesondere Media Queries für ein Responsive Design
- schema.org Annotationen mit Microdata
- Optional: Web Fonts, JavaScript

Das Web Projekt soll sich dabei an den Best Practices orientieren, welche wir in der Vorlesung besprochen hatten. Das umfasst u.a. die Trennung von Layout, Inhalt und aktiven Inhalten und die Benutzung von Validatoren. Das Projekt soll sich unterschiedlichen Bildschirmgrößen anpassen und mit einem aktuellem Firefox oder Chrome Browser funktionieren.

Es müssen folgende Dokumente abgegeben werden:

1. Ein ZIP file mit dem gesamten Web Projekt (relative links, damit es überall funktioniert!).
2. Ein PDF Dokument welche das Web Projekt beschreibt und dabei auf folgende Punkte eingeht:
Hintergrund und selbst gestellte Anforderungen, Aufbau und Erläuterungen zu technische Entscheidungen, Arbeitsweise, mögliche Weiterentwicklungen des Web Projektes. Dieses Dokument darf nicht mehr als 8 Seiten im LNCS Stil umfassen (dieser Stil ist Pflicht).

Abgabe der Projektarbeit erfolgt via campus.leipzigschoolofmedia.de.

Ziel der Projektarbeit ist die exemplarische Durchführung eines Web-Projektes mit allen Basis-Facetten. Die Menge des Inhalts ist dabei nebensächlich, wichtig ist der technische Durchstich, also alle Basis Web-Standard einmal angewendet zu haben.

Folgende Inhalte sind umfänglich völlig ausreichend:

- Die erste Seite einer Tageszeitung
- Die Haupt-Inhalte einer Amazon Produktseite incl. Reviews
- Die Inhalte einer nicht allzulangen Wikipedia-Seite
- Die begleitende Webseite zum LSoM Filmprojekt

Thank you for your Attention!

My WebID:

- <http://sebastian.tramp.name>