

Practical Software Engineering

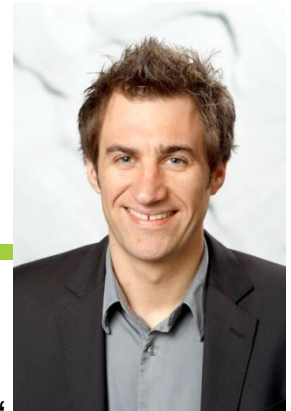


Organisation/Syllabus
(Christian Krenn, Egon Teiniker)

FH JOANNEUM
Software Design

WS 2014/15

Über mich...



Aktuell

- Teamleiter für MS-Technologien bei DCCS GmbH
- Schwerpunkte: Akquise, Requirements, Entwicklung, Promotor von „Agile“
- Vorträge u.a. zu Agilen Themen

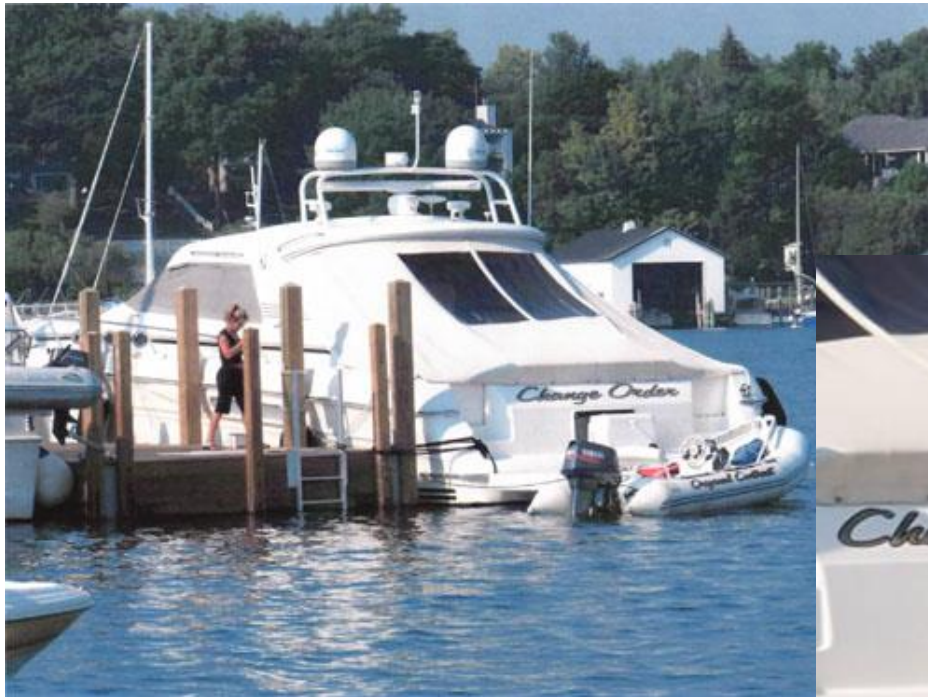
Davor

- Consulting- und Entwicklungstätigkeiten in internationalen Beratungshäusern
 - Automobilbranche, Telekom Branche
 - Von Kleinprojekten - Großprojekten (bis zu 100 Mitarbeiter)
 - Technologien: JEE, .NET, (Cobol)
- Studium der Telematik (spät berufen)

Warum mache ich das hier:

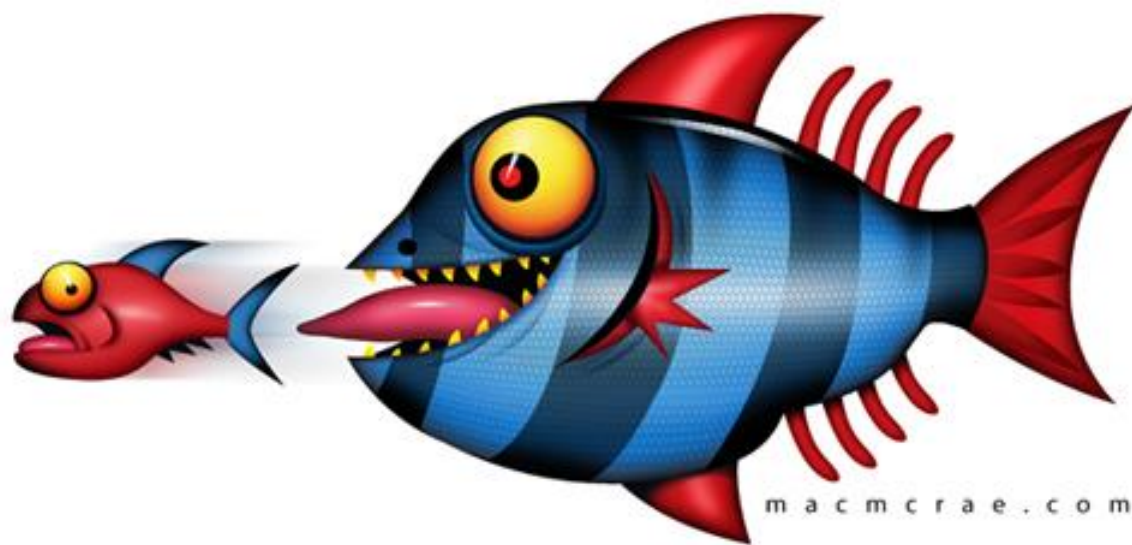
- ☹ Nicht aus monetären Gründen...
- Habe viele Projekte im Wasserfall begleitet / mitgemacht und kenne die Probleme
- Agile Entwicklungsmethoden sind Trend der Zeit
- Agil – aber nicht „Schwarz – Weiss“

Warum Agil?



Geht das nicht besser?

Agil = Anpassbar







Agil = Anpassbar

Organization

Practical Software Engineering

- Seminar: 2 SWS, 5 ECTS
- Grading
 - Project development 60%
 - Each iteration is evaluated separately
 - The final grade is calculated using the grades from the three separate iterations.
 - Class exam 40%

Note that both parts must be positive!

Organization

Seminar Software Design

Rating Scale

100	91	Sehr gut
90	81	Gut
80	71	Befriedigend
70	61	Genügend
60	0	Nicht genügend

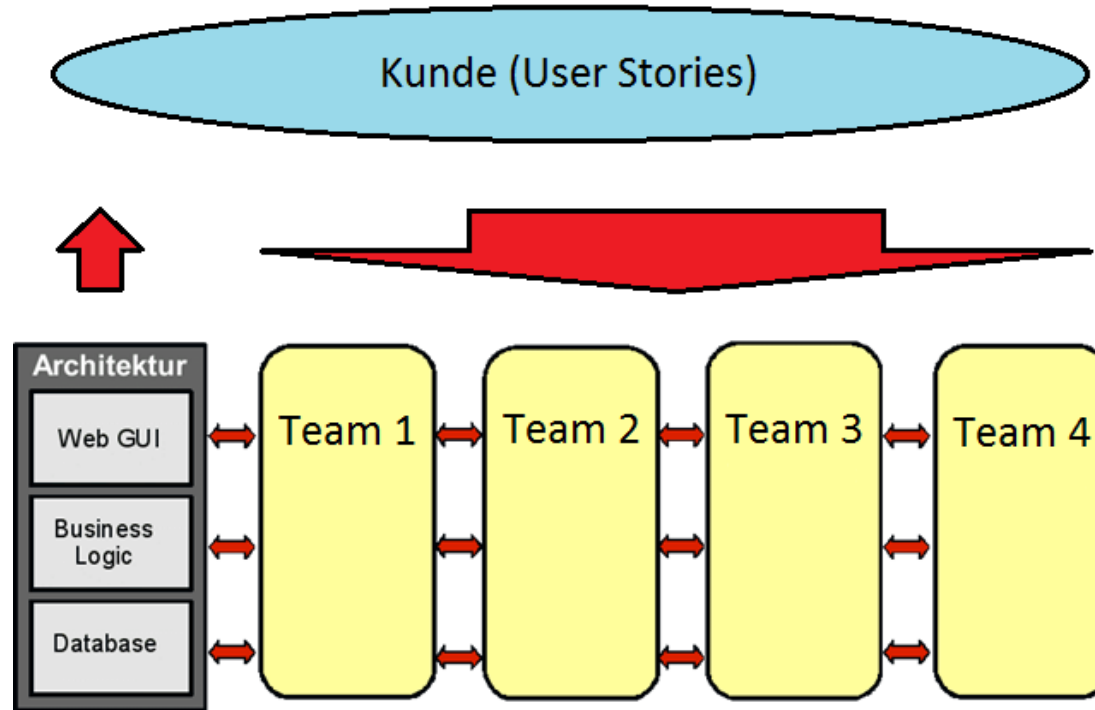
Organization

Agenda

- Open Source Platform
- Agile Software Development
- Continuous Integration
 - Source Code Repository
 - Automatic Build
 - Automatic Tests
 - Automatic Source Code Analysis
 - Automatic Documentation
- Architecture & Design

Organisation

Team Strukturen und Rollen



Organization

Team Struktur und Rollen

- Entwicklungsteams - Rollen
 - Team Manager (Planung, Controlling, Requirements)
 - 4 Entwickler (Design, Coding, Unit tests, Source Code, Dokumentation)
 - Leadentwickler (Architektur)
 - Tester (Qualität, Entwicklung)
 - Configuration Manager (Build Process, Project Struktur)
- Rollenaufteilung ist im Team zu definieren
 - **Rolle != Person**
 - **Jedes** Teammitglied entwickelt **cross-funktional** (GUI + Backend)

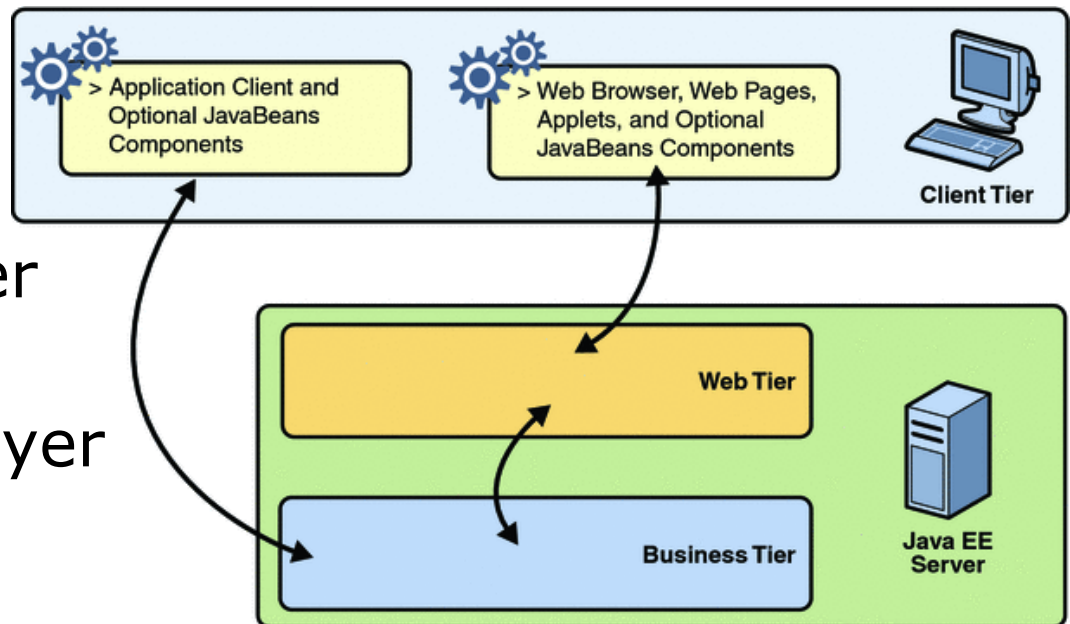
Organisation

- Gesamt: 28 Personen
- 4 Teams
 - 7 Personen
- Team Manager organisieren sich untereinander
- Leadentwickler sind federführend für Architektur verantwortlich

Technologien

Java Enterprise Platform (JEE)

- Web Clients
- Presentation Layer
 - JSF 2.0?
- Business Logic Layer
 - POJO, EJB
- Persistency Layer
 - JPA



Open Source Tools

- Java SE <http://java.sun.com>
- JBoss8 Wildfly <http://wildfly.org/>
- MySQL <http://www.mysql.com/>
- Eclipse <http://www.eclipse.org>
- Subversion <http://subversion.tigris.org>
- JUnit <http://www.junit.org>
- Maven <http://maven.apache.org/>
- Jenkins <http://jenkins-ci.org>

Google Code Plattform



The screenshot shows the Google Code project page for "fhj-ws2014-sd12-pse". The project description is "Student Project developing a Java Project the agile way". The page includes a search bar, navigation links (Project Home, Wiki, Issues, Source, Administer), and a summary section. The left sidebar contains project information: "Starred by 0 users" with a link to "Project feeds", "Code license" as "MIT License", "Labels" as "Academic, Agile, Java", "Members" as "chr.krenn, egon.teiniker", and "Your role" as "Owner". A tip box at the top right of the main content area says: "Tip: Project owners, see our [Getting Started](#) guide for steps to configure your project."

fhj-ws2014-sd12-pse
Student Project developing a Java Project the agile way

Search projects

Project Home [Wiki](#) [Issues](#) [Source](#) [Administer](#)

Summary [People](#)

Tip: Project owners, see our [Getting Started](#) guide for steps to configure your project.

Project Information

★ Starred by 0 users
[Project feeds](#)

Code license
[MIT License](#)

Labels
Academic, Agile, Java

Members
[chr.krenn](#), [egon.teiniker](#)

Your role
[Owner](#)

<https://code.google.com/p/fhj-ws2014-sd12-pse/>

TODOs (bis nächste Woche)

- Teams definieren
- Google Code Project ausprobieren
 - Google Account erstellen
 - Email Adresse zur Freischaltung mir schicken (Email: christian.krenn@dccs.at)
 - Google Code Platform ausprobieren

Termine

- User Stories
 - Abgabe: 06.10.2014, 23:55 (auf Google Drive)
 - Feedbackrunde in VO am 10.10.2014
- Iteration 1
 - Detaillierte Planung (User Stories, Aufwand, Zuordnung): bis 19.10.2014 23:55
 - Abgabe: 16.11.2014, 23:55
 - Erfahrungsberichte: 17.11.2014, 20:00
 - Kurze Präsentation (Demo) des Ergebnisses am 21.11 (+ Feedback)
- Iteration 2
 - Abgabe: 11.01.2015, 23:55
 - Kurze Präsentation des Ergebnisses online am 16.1.2015 (+ Feedback)
- Klausur:
 - Tbd, im Januar 2015: Vorschläge willkommen.

Aufgabenstellung

- Thema: SWD12 Enterprise 2.0 Portal
 - Portal-Startseite
 - Activity-Streams
 - Communities
 - Userseiten
 - Services
- Ziel: Agiler Entwicklungsprozess in 2 Iterationen
 - Erhebung der User Stories
 - Testgetriebene Implementierung mit Continuous Integration
 - Dokumentation (Tests)
 - Abnahme (= Demo + Bewertung 😊)
- Motto der Iteration: “Funktionierende Software”

Requirements (1)

- SWD12 Enterprise 2.0 Portal
 - Portal besteht aus
 - Headerbereich
 - Inhaltsbereich
 - Startseite
 - Communities-Seite
 - Services-Seite für personalisierte Services (Erweiterung)
 - Users-Seite (Liste aller User)
 - Administratorenbereich

Requirements (2)

- Seiten generell
 - Headerbereich
 - Navigation
 - Angemeldeter User (mit Link auf Userseite)
 - Inhaltsbereich
 - Untergeordnete Navigation (falls nötig)
 - Stream
 - Sonstige Detail-Inhalte

Requirements (3)

- Rollen
 - User darf generell editieren
 - Administratoren
 - Portal Administratoren (News auf Startseite)

Requirements (4)

- Startseite
 - Hat zentralen Activity-Stream (personalisiert)
 - Unternehmens-Newsbereich
 - Titel, Inhalt, Foto (später)
 - Online von/bis
 - Editieren durch Portaladmins
 - Lesen können alle
 - Dokumentenbibliothek
 - Editieren durch Portaladmins
 - Lesen/Download alle
 - Userseite (Personen des Unternehmens)
 - Services (personalisierbar) -später

Requirements (5)

- Activity Stream

- Global auf Startseite, lokal in Community
- User kann Nachricht schreiben
- Nachricht kann lokal sein (Community) oder global (alle sehen Nachricht) - Filter
- Like- Funktion
- Kommentarfunktion
- User sieht seinen gesamten Stream auf Startseite
- User sieht Community Stream in Community
- User können Nachrichten mit Tags versehen (Notation?)
- Tags entsprechen Interessen und Expertisen, Communities
- User bekommt Nachrichten entweder global oder über Communities oder über Interessen und Expertisen oder über Kontakte (Team + Netzwerk)
- Private Messages an User über Emailadresse (als Tag)

Requirements (6)

- Userinformationen
 - werden importiert (Datei-SST)
 - Vorname, Nachname
 - Emailadresse (=Logon)
 - Pwd
 - Geografische Adresse + Zimmer
 - Telefonnummer
 - Geografischer Ort
 - Abteilung/Team (Hierarchie)
 - Interessen (mehrere)
 - Expertisen (mehrere)
 - Foto (optional) – Upload
 - Status (online, offline, extern) - Später
 - Kontakte

Requirements (7)

- Userseite
 - Mein Netzwerk (Abteilung+ Kontakte)
 - Details zum User aus Netzwerk
 - Kontakt hinzufügen (Oneway keine Bestätigung)
 - Kontakt entfernen
 - Userinformationen bearbeiten

Requirements (8)

- Communities

- Übersichtseite
- Können beantragt werden (von jedem)
- Freischaltung über Administrator
- Können öffentlich oder privat sein
- Es gibt personalisierte Übersichtseite (meine, alle öffentlichen)
- Mitgliederverwaltung (self service bei Öffentlichen, Freischaltung bei privaten durch Administratoren; später durch Community Admin)
- Haben Activity Stream (wie beschrieben)
- Haben Dokumentenbibliothek (Aktivität erscheint im Stream)
- Inhalte von Community-Stream erscheinen auch auf Startseiten-Stream

Requirements (9)

- Userseite
 - Selbstsicht
 - User kann manche Infos pflegen (Zimmer, Tel, Interessen, Expertisen, Foto, Status)
 - Später: Persönliche oder öffentliche Dokumente hochladen
 - Später: Fotoupload
 - Außensicht
 - Darstellung der Infos
 - User als Kontakt hinzufügen
 - Wenn User mein Kontakt, sehe ich seine „öffentlichen“ Activities im Stream
 - Orte sollen über Google Maps dargestellt werden können (optional)
 - Privater Nachrichtenaustausch (sieht User im Stream auf der Startseite)
 - Wenn User Offline, kann ich im Nachricht per SMS senden
 - Kontakte

Requirements (10)

- Verwaltungsbereich (Administratoren)
 - Freischaltung von Communities

Requirements (11)

- Services
 - Erfassung von Dienstreisen
 - Projektverwaltung
 - Zeiterfassung
 - ...

Inhalt Iteration 1(1)

- Erstellen der User Stories (jeder beteiligt sich)
- Erheben weiterer Details (wenn nötig)
 - Rückfragen an mich
- Aufwandsschätzung und Planung für Iteration 1
 - Frage: Wieviel kann umgesetzt werden?
 - Aufwandsabschätzung (in SP)
 - Priorisierung (durch mich)
 - Zuordnung je Team

Inhalt Iteration 1(2)

- Implementierung
 - Architektur
 - Umsetzung Features test-driven
 - Aufsetzen CI (nur Build)
- Testing
 - Unit Tests
 - Testprotokoll (Abnahmetests)
- Tatsächlicher Aufwand (jeder) – Gesamtreport
 - Als Burndown Chart
- Kurzer Erfahrungsbericht (jeder)

Was ist uns wichtig

- Sie arbeiten als selbstorganisiertes Team
 - Sie als Team sind für den Erfolg verantwortlich!
 - Sie üben agil!
 - Selbsteinschätzung lernen
 - Aus Erfahrung lernen (Continuous Improvement)
- Testgetriebene Entwicklung
 - Führt zu testbarem Design und hat mehr Qualität
 - Software ist fertig, wenn Sie läuft
 - „Weniger ist mehr“ – Was sie abgeben funktioniert

Tipps

- Nehmen Sie sich nicht zuviel vor!
 - Ziel: Features sollen funktionieren
 - Definition of Done: Fertig programmiert != fertig
- Beachten Sie Ramp Up Effekt
 - Architektur braucht auch in agilen Projekten Vorlaufphase
 - Team-Building (kostet Zeit)
- Definieren Sie Spielregeln für die Kommunikation
 - Sie arbeiten und funktionieren nur als Team
 - Email ist asynchron!
- Fangen Sie frühzeitig an
 - Don't: Massen-Checkins am Abgabetag