

# Teamserver

## Werkzeugunterstützung für kleine Projekte

Gerd Beneken

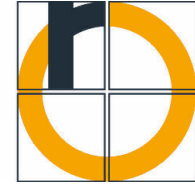
GI 2008, 3. Workshop „Vorgehensmodelle in der Praxis“

09.09.2008

# Agenda

---

1. Projekt SE-KMU
2. Software-Entwicklung in KMU
3. Projekt Teamserver
4. Bestandteile: Werkzeuge für kleine Projekte
5. Teamserver: Integrationsstufen
6. Zusammenfassung



# 1. Projekt SE-KMU

Thema:

- *Software-Entwicklung und Software-Einführung in kleinen und mittleren Unternehmen*

Fragestellung:

- *Was macht KMU erfolgreich?*

Projekthalte:

- Stärken/Schwächen-Analyse der Ist-Prozesse
- Erarbeitung eines Leitfadens für SW-Entwicklung und SW-Einführung in KMU

Untersuchte KMU:

- 11 Unternehmen, ca. 2 - 20 Entwickler, 5 - 100 MA
- i.d.R. Produkt + Projektgeschäft
- Methodik: Persönliche Interviews mit Fragebogen als Rahmen

## 2. Software-Entwicklung in KMU

### Einige Schlaglichter aus SE-KMU

---

- Prozesse:
  - Prozesse gelebt, gewachsen, (erfolgreich)
  - Prozesse kaum dokumentiert, Wissen in den Köpfen
  - Kaum Dokumente während der Projektdurchführung
  - Projektergebnis: Laufende Software + Benutzerdoku
  - Schwerpunkt: Persönliche Kommunikation
- Werkzeuge:
  - Zum Teil umfassende Werkzeugunterstützung (z.B. MS-TFS)
  - Zum Teil nur IDE ohne Versionsmanagement
  - Typische Werkzeuge wie Issue-Tracker teilweise unbekannt
  - Kaum automatisierte Tests, automatisierte QS, autom. Build
  - Kaum Modellierung, kaum Spezifikation, „nur“ Codierung

### 3. Teilergebnis SE-KMU Rosenheimer Projekt „Teamserver“

- Ziel: Werkzeugunterstützung für kleine Projekte
  - Möglichst wenig Aufwand für Verwaltung / Installation / Konfiguration (*Neue Projekte auf Knopfdruck*)
  - Kostenlos/kostengünstig: möglichst freie Werkzeuge
  - Umfassend: PM, QS, KM, REQM, ...
  - Technikneutral: Java, C#, C++, Delphi, ...
  - Schüsseltätigkeit: Programmierung, Test
  - Zielgruppe: Gesamtes Entwicklungsteam
- Wie: Studentische Projekte
  - Jeweils 3-5 Studierende konzipieren einen Teil
  - Test des Teamserver: Studentische Projekte im Fach „Software-Engineering 2“

## 4. Was braucht ein (kleines) Projekt mindestens?

- Mailverteiler
- Gruppenlaufwerk / Projektablage
- Projekthandbuch / Projekt-Wiki
- Versionsmanagement / Konfigurationsmanagement
- Issue-Tracker
- Automatisiertes Build / Continuous Integration Infrastr.
- Möglichkeit für Projektberichte und -analysen

Ziel: Neue Projekte per Knopfdruck

## 5. Teamserver Integrationsstufen

---

- **Stufe 0: Oberflächen Integration**
- Stufe 1: Datenintegration
- Stufe 2: Strukturiertes Messen
- Stufe 3: Prozessunterstützung
- ...

Wichtig: **Nutzungskonzept**

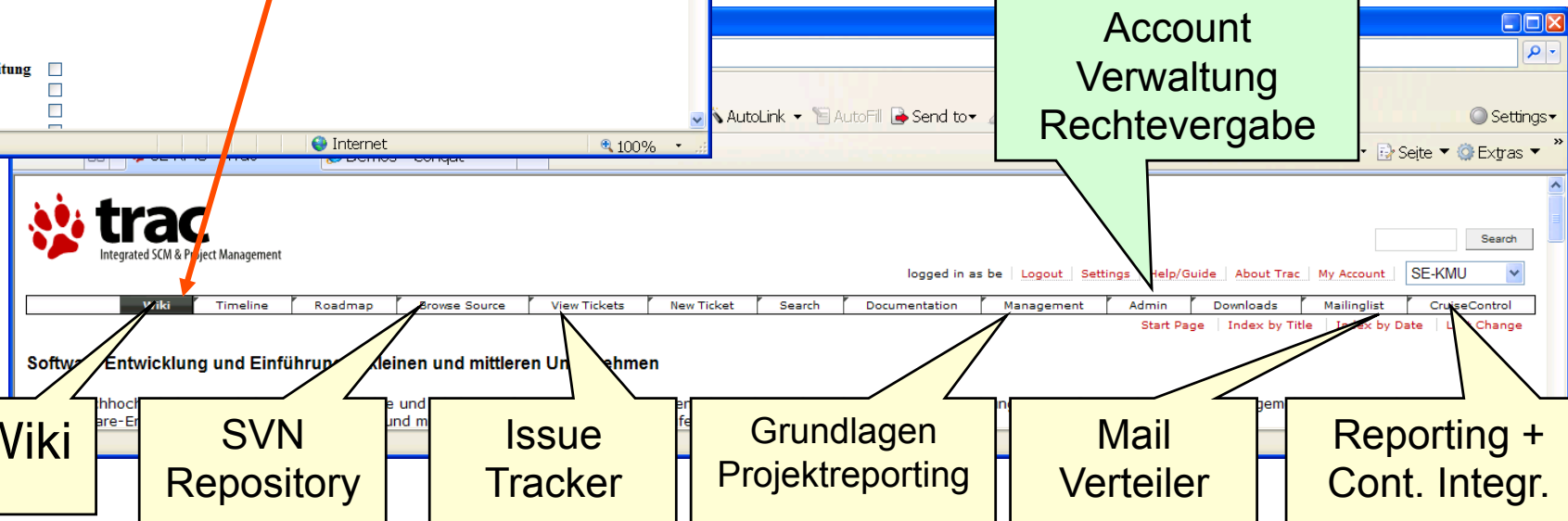
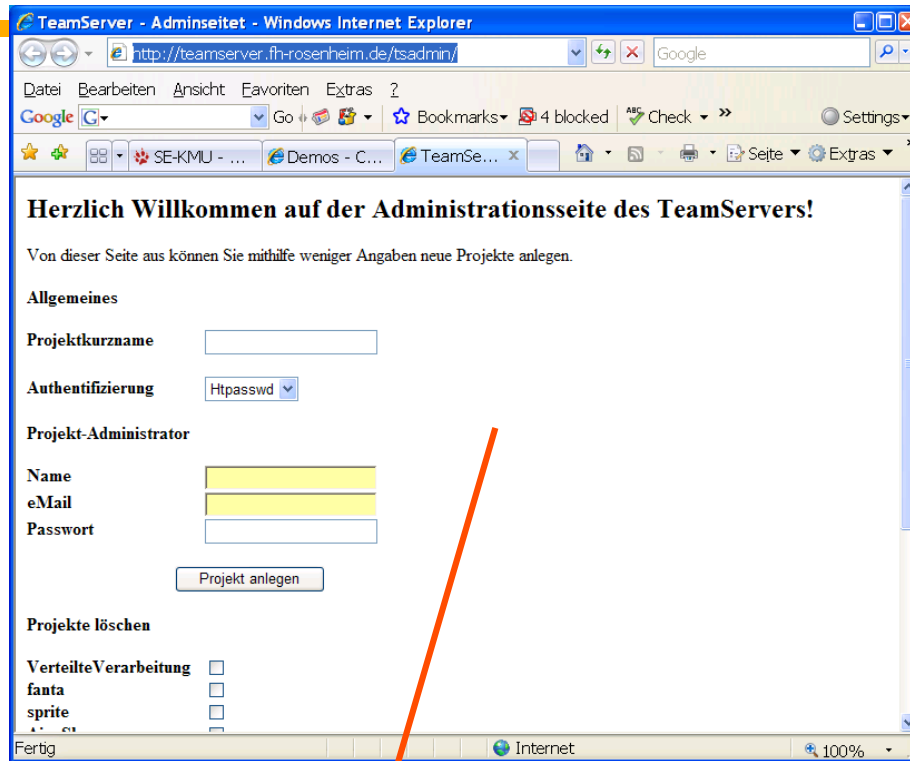
## 5. Stufe 0 – *Aktueller Stand* Oberflächen Integration der Werkzeuge

- Mailserver: Mailman
- (Gruppenlaufwerk: WebDav)
- Projekthandbuch/-wiki: Trac Wiki
- Versionsmanagement: Subversion
- Konfigurationsmanagement: Subversion + Trac Wiki
- Issue Tracker: Trac
- Autom. Build: Ant + CruiseControl
- Berichte/Analyse: ConQAT (TU-München)
  
- Infrastruktur: VM-Ware, Linux
- Client: Eclipse / Browser

## 5. Stufe 0 – *Aktueller Stand* Trac als Integrationsplattform

Plattform: Linux auf VMWare  
Integration über

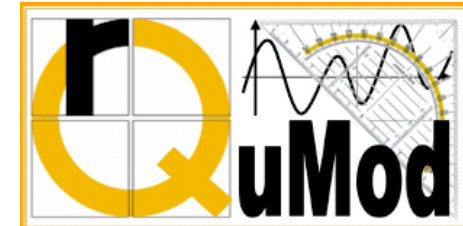
- Shell Scripte
- Trac Plugins (Python)
- Zukunft: eigene Plugins



## 5. Stufe 1 Datenintegration

- Lose Kopplung der Datenquellen
  - HTML-Links im Wiki  
(Dateien im SVN-Repository/Gruppenlaufwerk, Tickets im Issue Tracker, Queries über Tickets, Tags/Branches im Repository, Mails im Mailarchiv ...)
  - Checkin-Hooks im SVN-Repository  
(z.B. Ticket-ID als Checkin-Kommentar)
  - Berichts/Analyse-Werkzeug, zapft alle Datenquellen an  
(Repository, Mailarchiv, Issue-DB, Build-Reports, QS-Reports)
  - Nightly Build erzeugt auch Berichte / Analysen / Doku.
  - Projektmitarbeiter == Mailverteiler, Mail ggf. == Ticket
- Aber: Kein gemeinsames Datenmodell/Metamodell, keine gemeinsame Datenbank

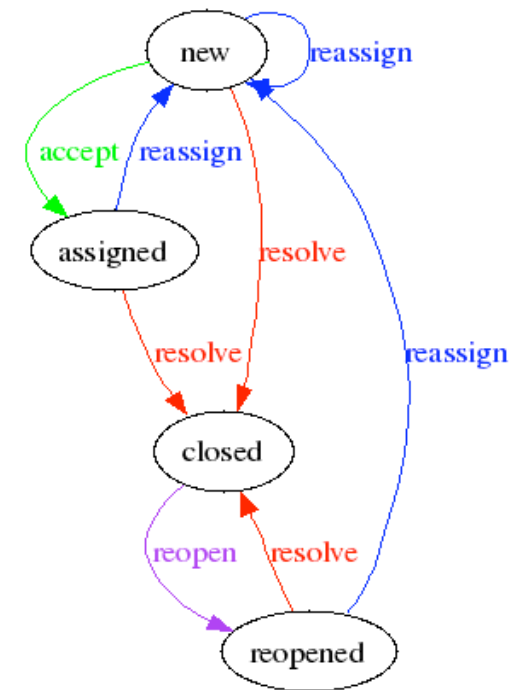
## 5. Stufe 2 Strukturiertes Messen



- PM: Projektcontrolling über Tickets des Issue-Trackers
  - Ticket-Typ „User Story“ im Issue-Tracker, möglichst Hierarchische Tickets: User Story + Tasks
  - Fertigstellungsgrad über abgeschlossene User Stories/Tasks und Restaufwände für offene User Stories zu einer Iteration
  - Messung von Bugfixing-Raten, Bearbeitungszeiten für Tickets
  - Messung von Schätzgenauigkeiten der Entwickler
- QS: Softwarecontrolling (RoQuMod)
  - Strukturiertes Erheben relevanter Metriken für Code Qualität, Testqualität, Architektur-Einhaltung, ...
  - Entwicklung und Verbesserung eines Frühwarnsystems

# 5. Stufe 3 Prozessunterstützung über Rollen + Ticketworkflows

- „Alles ist ein Ticket“
  - Anforderungen, Risiken, Arbeitspakete, Fehler, ...
- Für jeden Ticket-Typ ein eigener Workflow/Prozess
  - Rollen / Rechte Modell, flexible Workflows
  - Steuerung der Bearbeitung über Workflows + Rollen
  - Unterstützung über Repository (Commit-Hooks erzwingen gültiges Ticket, Kein Checkin ohne Auftrag)
- Generierung der wichtigsten Meilensteine und Tickets gem. Tailoring des Vorgehensmodells
- Generierung von Dokumenten über Query auf Tickets ...



Ticketworkflow Trac 0.10

## 6. Zusammenfassung

---

- Teamserver aktueller Stand
  - Ziel: Unterstützung kleiner Projekte, insbesondere Code-Erstellung und Test
  - Integration typischer Entwicklungs-Werkzeuge (Server): Versionsmanagement, Issue-Tracker, Autom. Build, ...
  - Zusätzlich: Mailserver, Gruppenlaufwerke, ...
- Ausblick
  - Strukturiertes Messen
    - Projektcontrolling mit Informationen aus Issue-Tracker
    - Software-Controlling über Qualitätsmodell und kont. Messung
  - Umfassende Prozessunterstützung über Ticket-Workflows im Issue-Tracker

# Backup

# Was braucht ein (kleines) Projekt? Mailverteiler



- Ziel: **Informationsverteilung und –archivierung**
- Verteilung
  - **Gesamtverteiler** enthält alle internen ProjektmitarbeiterInnen
  - Spezielle Verteiler für Projektgremien / Teilprojekte (Steuerkreise, Taskforces, ...)
- Archivierung
  - Mails werden vom Mailverteiler gespeichert
  - Mails sollen zu Issues des Issue-Trackers werden können (z.B. über CC an den Issue-Tracker)
- Teamserver: Mailman
  - Administration/Moderation über Web-Interface
  - Ansicht der Mails über Thread, Datum, Autor, ...

# Was braucht ein (kleines) Projekt? Projekthandbuch / Projekt-Wiki



- Ziel: „Handbuch“ für Projektmitarbeiter, das alle wichtigen Informationen enthält
  - Kurzbeschreibung des Projektes
  - Projektorganisation, grobe Projektplanung
  - Liste der Ansprechpartner
  - Beschreibung der Projektablage (wo findet man was?)
  - Anpassung des verwendeten Vorgehensmodells, ggf. auch das Vorgehensmodell selbst (=Überblick, Vorlagen, Checklisten)
- Wichtig: Aktualität, Richtigkeit der Informationen
- Teamserver: Trac-Wiki (alternativen: DokuWiki, MediaWiki)
  - Inhalte können schnell über WebBrowser angepasst werden

SE-KMU - Trac - Windows Internet Explorer

http://teamserver.fh-rosenheim.de/trac/sekmu

Google

trac  
Integrated SCM & Project Management

logged in as be | Logout | Settings | Help/Guide | About Trac | My Account | SE-KMU

Wiki | Timeline | Roadmap | Browse Source | View Tickets | New Ticket | Search | Documentation | Management | Admin Mailinglist | Downloads CruiseControl

Start Page | Index by Title | Index by Date | Last Change

## Software Entwicklung und Einführung in kleinen und mittleren Unternehmen

Die Fachhochschule Rosenheim und sechs kleine und mittlere Software-Häuser aus Rosenheim und Umgebung haben unter der Leitung von Prof. Dr. Roland Feindor ein gemeinsames Forschungsprojekt „Software-Entwicklung und -Einsatz in kleinen und mittleren Unternehmen“ aus der Taufe gehoben.

Mit finanzieller Unterstützung der Bayerischen Forschungsstiftung wird erforscht, wie sich die Methoden bei der Software-Entwicklung und dem Software-Einsatz bei kleinen und mittleren Unternehmen von den in Großunternehmen eingesetzten Methoden unterscheiden und worin die „Erfolgsgeheimnisse“ der Kleinen liegen.

Das fördert natürlich die Konkurrenzfähigkeit der Unternehmen und damit auch die Sicherheit der Arbeitsplätze in der Region.

In den kommenden zwei Jahren wird von Diplomanden der FH Rosenheim und von den beteiligten Unternehmen Forschungsarbeit für das Projekt geleistet. Studenten analysieren in den Unternehmen die Abläufe, vergleichen und bewerten sie und forschen nach weiteren Verbesserungsmöglichkeiten.

### Informationen zum Projekt

- [Ziele](#)

### Weitere Informationen

- [Verwandte Projekte](#)

### Quick Start:

[All active Tickets](#) (detailed)  
[My Tickets](#) (detailed)

### My Own Tickets:


- #6 Abschlussbericht erstellen
- #5 Abschlussbericht SE-KMU

### Latest Tickets:

- #6 Abschlussbericht erstellen
- #5 Abschlussbericht SE-KMU
- #4 Nachdokumentation und Kommentierung des SE-Checks
- #3 Ist Aufnahme der Software-Einführungsprozesse
- #2 Ist Aufnahme der Software-Entwicklungsprozesse
- #1 Eigenschaften kleiner und mittlerer Unternehmen

Edit this page | Attach file | Delete this version | Delete page

# Darstellung eines einfachen Entwicklungsprozesses im Wiki

 **Prozessperspektive**

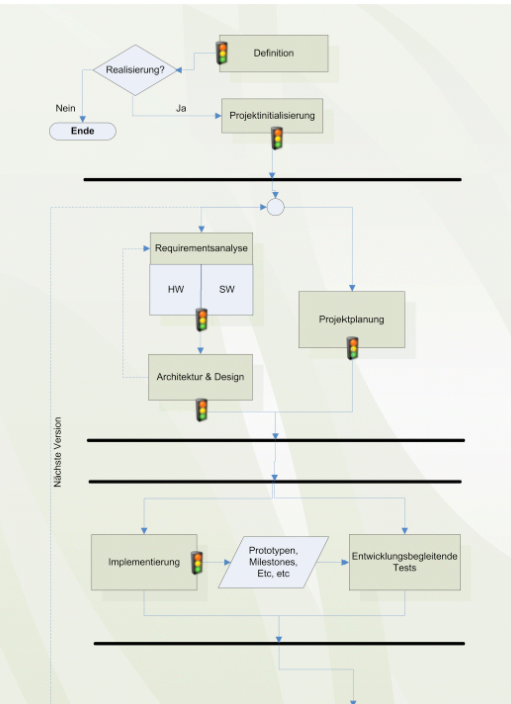
**Aufbau des Software-Entwicklungsprozesses** [\[bearbeiten\]](#)

Der KSEP ist in 2 Bereiche untergliedert:

1. [Prozessstrecken](#) und
2. eigenständige Prozesse

Klicken Sie einfach auf eine [Prozessstrecke](#) oder einen eigenständigen Prozess über den Sie mehr Informationen haben möchten. Sie werden eine detaillierte Dokumentation des gewählten Abschnittes erhalten.

**Die Prozessstrecken** [\[bearbeiten\]](#)



Nummer	Prozessstrecke	Freigabe
E0	Einstiegspunkt: neues Projekt	-
PS1	Definition	19.7.2007
PS2	Projektinitialisierung	28.9.2007
E1	Einstiegspunkt: neue Baseline	-
PS3	Requirementsanalyse	nein
PS4	Projektplanung	nein
PS5	Architektur & Design	nein
PS6	Implementierung	nein
PS7	Entwicklungsbegleitendes Testen	nein
PS8	Test & Integration	nein
PS9	Release	nein
PS10	Produktion	nein
PS11	Kundenabnahme	nein
PS12	Maintenance	nein
PS13	Projektabschluss	nein

- Darstellung des Prozesses
- Vorlagen, Checklisten, Beispiele zum Download
- Verschiedene Perspektiven
  - Prozesssicht
  - Rollensicht
  - Normensicht
- Möglichst: Generiert aus Versionsverwalteten Dokument(en)

Aus: Diplomarbeit Andy Phillip

# Was braucht ein (kleines) Projekt? Versionsmanagement



- Ziel: **Zentrales Versionsmanagement**
  - Verwaltung von Quelltexten, Buildscripten, Dokumenten ...
  - Arbeiten an gemeinsamer Codebase, ...
  - Wiederherstellung bereits ausgelieferter Software-Releases
  - Erstellung von Branches für BugFixing an Releases
  - ...
- Teamserver: Subversion (SVN)
  - „Nachfolger“ von CVS
  - Versioniert Verzeichnisse statt Dateien, hat Transaktionen
  - Branching / Tagging einfach (Lazy Copy)
  - Zugriff: z.B. WebDAV, TortoiseSVN (Explorer), Eclipse, ...

## Was braucht ein (kleines) Projekt? Versionsmanagement - Client



TortoiseSVN

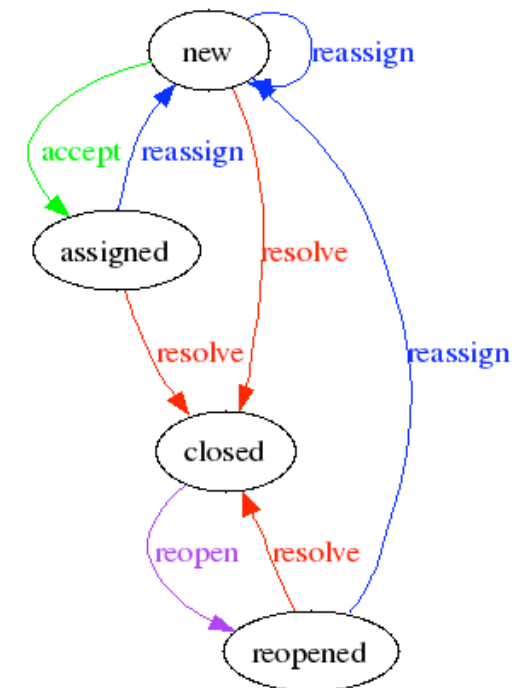
The screenshot displays the TortoiseSVN client interface. On the left, a Windows Explorer window shows a directory structure for '04\_Anforderungsmanagement'. The main area shows a file explorer with various files and folders. Overlaid on this are three windows:

- Log Messages - C:\Daten\Rosenheim\seu**: A window showing a list of revisions. The selected revision is 403.
- Enter Log Message - C:\Daten\Rosenheim\se1**: A dialog box for committing a file. The commit message is: `Behebung des Fehlers #1235, HashMap korrigiert und zusätzliche Fehlerbehandlung in Methode "wakaluba()"`. The file 'Tach.txt' is listed as modified.
- Statistics**: A window showing a line graph titled 'Commits by date' with a y-axis from 0 to 18 and an x-axis labeled 'week' with dates from 5/07 to 3/08.

# Was braucht ein (kleines) Projekt? Issue-Tracker



- Ziel: **Zentrales Änderungs/Problem/Anforderungs-Management**
  - Anforderung, Fehler, Risiko, Aufgabe, ... = Issue/Ticket
  - Verwaltung aller Informationen in einem Issue-Tracker/BugTracker
  - Wichtig: Workflow/Zustandsmodell für Issues/Tickets
- Issues/Tickets/Tasks
  - Haben eine Beschreibung, Einen Autor, einen Bearbeiter, eine Priorität, eine Deadline
  - Können einem Meilenstein zugeordnet werden
  - Können im PM verfolgt werden: Termine, Aufwand (Plan, Ist), ...
  - Haben einen Bearbeitungs-Workflow
- TeamServer: Trac (Altern.: Mantis, Bugzilla, ...)
  - Trac wird auch als „Integrationsoberfläche“ verwendet
  - Wichtig: Nutzungskonzept
  - Ähnliche Idee: „Agilo for Scrum“



#3 (Ist Aufnahme der Software-Einführungsprozesse) - SE-KMU - Trac - Windows Internet Explorer

http://teamsserver.fh-rosenheim.de/trac/sekmu/ticket/3

trac  
Integrated SCM & Project Management

logged in as be | Logout | Settings | Help/Guide | About Trac | My Account | SE-K

Wiki | Timeline Documentation | Roadmap Management | Browse Source CruiseControl | **View Tickets Admin** | New Ticket Downloads

## Ticket #3 (new task)

### Ist Aufnahme der Software-Einführungsprozesse

Opened 2 months ago  
Last modified 2 months ago

Reported by:	be	Assigned to:	solms
Priority:	blocker	Milestone:	
Keywords:		Cc:	
Tracforge_linkmap:		Estimated Number of Hours:	0
	0		Yes
Total Hours:	0	Deadline:	

Description

Aus den vorhandenen Diplomarbeiten und zusätzlichen Interviews sollen die Software-Einführungsprozesse in den untersuchten Firmen herausgearbeitet und dokumentiert werden. Folgende Informationen sind interessant:

- Dokumentationsgrad (ist der Prozess dokumentiert oder ist er nur in den Köpfen?)
- Rollen / Projektorganisation in den KMU: Wer ist für was verantwortlich? Gibt es die klassischen Rollen wie Projektleiter, Architekt, Tester und Entwickler?
- Aktivitäten: Was wird konkret getan?
- Artefakte / Produkte: Welche Zwischenergebnisse (Dokumente, Code, Testspezifikationen, Pläne, ... ) werden erzielt?
- Workflow: Gibt es einen definierten Prozess? Wenn ja, wie sieht der aus?

Die Aussagen sollten über einen Fragebogen, Interviews oder den Text in den Diplomarbeiten sein.

Attachments

Internet

#3 (Ist Aufnahme der Software-Einführungsprozesse) - SE-KMU - Trac - Windows Internet Explorer

http://teamsserver.fh-rosenheim.de/trac/sekmu/ticket/3

### Add/Change #3 (Ist Aufnahme der Software-Einführungsprozesse)

Comment (you may use WikiFormatting here):

**B I A** wysiwyg  textarea

Change Properties

Summary: Ist Aufnahme der Software-Einführungsprozesse

Type: task

Description: **B I A** wysiwyg  textarea

verantwortlich? Gibt es die klassischen Rollen wie Projektleiter, Architekt, Tester und Entwickler?  
 \* Aktivitäten: Was wird konkret getan?  
 \* Artefakte / Produkte: Welche Zwischenergebnisse (Dokumente, Code, Testspezifikationen, Pläne, ... ) werden erzielt?  
 \* Workflow: Gibt es einen definierten Prozess? Wenn ja, wie sieht der aus?

Reporter: be

Priority: blocker

Milestone:

Keywords:

CC:

Tracforge\_linkmap:

Estimated Number of Hours: 0

Add Hours to Ticket: 0

Billable?:

Total Hours: 0

Deadline:

Action

leave as new

accept ticket

resolve as: fixed

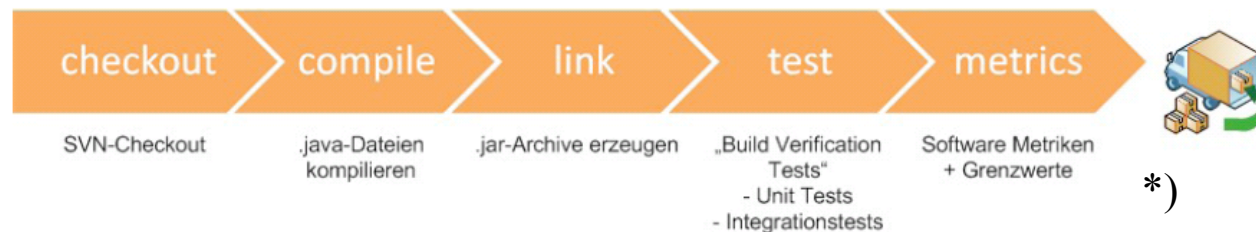
reassign to: be

Preview Submit changes

Internet

# Was braucht ein (kleines) Projekt? Automatisiertes Build

- Ziel: **Automatisierung der Übersetzung der Software**
  - Unabhängig von der Entwicklungsumgebung
  - Nach jedem Checkin ins SVN (=Continuous Integration)
  - Einmal pro Nacht (=Nightly Build)
- Bei jeder Übersetzung
  - Generieren von Quelltexten, Übersetzen der Quelltexte
  - Durchführung der automatisierten Tests (UnitTests, ...)
  - Durchführung automatisierter Metriken
  - Erstellung von Berichten
  - Bereitstellung des Build-Ergebnisses



- Zusätzlich: Wiederherstellung und autom. Build ausgelieferter Releases
- Teamserver: CruiseControl + ANT (Alternativen: Continuum + Maven, ...)

The screenshot displays two overlapping web browser windows. The background window shows the Trac project page for 'CruiseControl - SE-KMU - Trac'. The foreground window shows the 'CruiseControl Build Results' page for a specific build.

**Trac Project Page (Background):**

- URL: <http://teamserver.fh-rosenheim.de/trac/sekmu>
- Project: **trac** (Integrated SCM & Project Management)
- Build from: **03/31/08 19:42:08 build.1**
- Unit Tests: (0) - No Tests Run. This project doesn't have any tests.
- Modifications since last successful build: (21)
- Powered by Trac 0.10.4 By Edgewall Software.

**CruiseControl Build Results (Foreground):**

- URL: <http://build.inf.fh-rosenheim.de:8080/buildresults/flyby?log=log2006121515533>
- Project: flyby
- Status: **BUILD COMPLETE - build.1**
- Date of build: 12/15/2006 15:53:37
- Time to build: 47 seconds
- Last changed: 12/15/2006 15:26:04
- Unit Tests: (158) - All Tests Passed
- Modifications since last successful build: (11)
- Deployments by this build: (1)

Change Type	Build	File Path	Timestamp
modified	build	/flyby/src/test/de/fhro/se2/flyby/application/stammdaten/dt/TestKennzeichen.java	12/15/2006 15:26:04
modified	build	/flyby/src/test/de/fhro/se2/flyby/application/stammdaten/dt/TestKennzeichen.java	12/15/2006 15:02:42
modified	build	/flyby/config/common.xml	12/15/2006 12:05:25
modified	build	/flyby/config/common.xml	12/15/2006 11:56:37
added	build	/flyby/config/checkstyle-noframes-sorted.xml	12/15/2006 11:56:37
modified	build	/flyby/config/common.xml	12/15/2006 11:48:41
modified	build	/flyby/config/common.xml	12/15/2006 11:43:38
modified	build	/flyby/config/build.properties	12/15/2006 11:43:38
added	build	/flyby/config/checkstyle.xml	12/15/2006 11:43:38
modified	build	/flyby/config/build.xml	12/15/2006 11:43:38
modified	build	/flyby/config/buildcc.xml	12/15/2006 11:43:38

# Was braucht ein (kleines) Projekt? Reporting / Software Controlling



- Ziel: **Flexible Projektberichte erstellen**
  - Fortschrittsberichte / Fertigstellungsgrad / Fixing-Rate
  - Fehlerberichte (Fehlerzahl, gefixte Fehler, ...)
  - Qualitätsberichte (Metriken, Coding Style Verletzungen, ...)
- Erstellt im automatisierten Build
- Flexibel konfigurierbar  
(Datenquellen, (Vor-)Verarbeitung, Visualisierung)
- TeamServer: ConQAT (TU-München, Open Source)
  - Flexible Konfiguration der Verarbeitung über Graphen
  - Daten aus Build, Metriken, Repository, IssueTracker über **Adapter**
  - Visualisierung über Grafiken, Tabellen, ..., flexibel erweiterbar

