

# Verteilte Paarprogrammierung in Open Source Projekten

Eike Starkmann  
Institut für Informatik  
FU Berlin  
23.07.2009

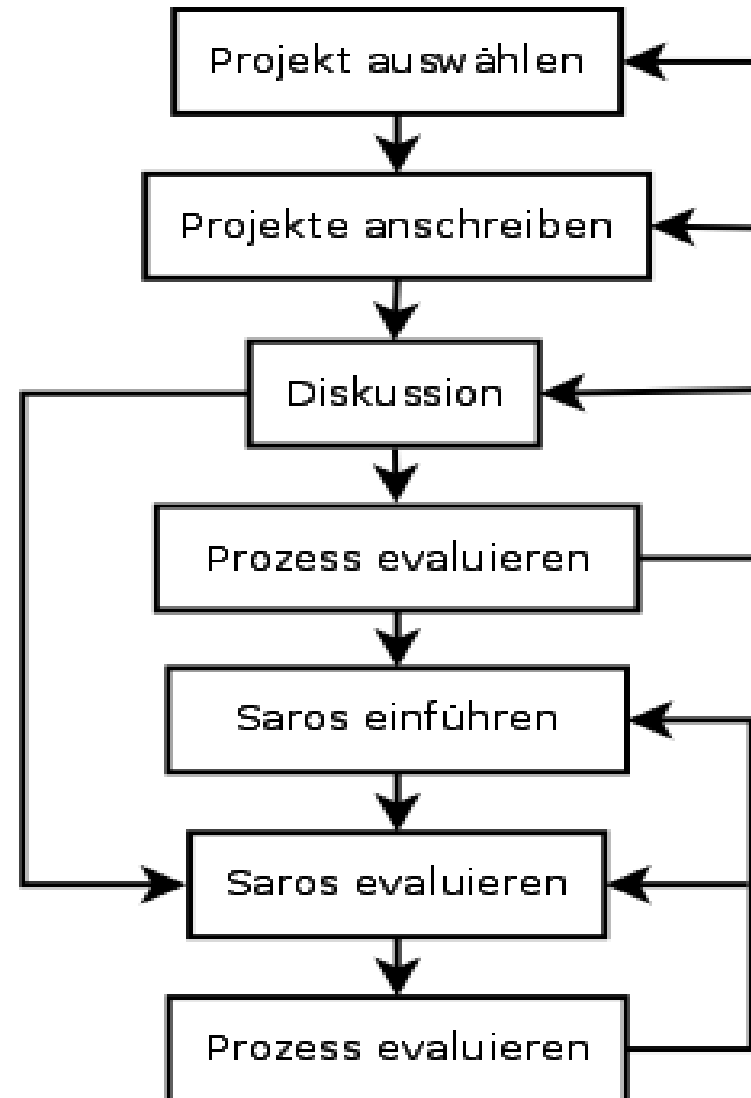
---

- Kontext und Ziele der Arbeit
- Plan der Durchführung
- Projektauswahl
- Projekt anschreiben
- Saros einführen
- Saros evaluieren
- Zeitplan
- Fragen / Anmerkungen

- Saros in OSS Projekten einführen
- Zusammenarbeit in OSS Projekten erfolgt meistens nur per email, Wiki und Versionsverwaltung.
- Dadurch wird die Zusammenarbeit erweitert und Arbeitsweise verändert. (Vorteile und Nachteile)
- Zentrale Forschungsfragen
  - Wie ist Saros in OSS Projekten einsetzbar?
    - Wie wird Saros benutzt?
    - In welchem Kontext wird Saros benutzt?
  - Wie kann Saros in OSS Projekten eingeführt werden?
  - Welche Folgen ergeben sich durch dem Einsatz von Saros in OSS Projekten?

- Theoretisches Wissen (angefangen)
  - Methodik: Action research
  - Innovationseinführung
  - Kontaktstrategien
  - Open Source Software

- Arbeitsschritte (nicht so klar trennbar):



- Projekteigenschaften
  - ca. 5-50 Entwickler
  - Eclipse muss verwendet werden
  - Offene Kommunikation (Mailingliste)
  - Projekt ist Open Source per OSI Definition
  - Projekt sollte von Saros profitieren
- Sehr passende Projekte genauer analysieren (TOP-Projekte)
  - Wie werden Entscheidungen getroffen
  - Wer sind die maintainer
  - Wie sind sie Organisiert
  - Wie oft werden releases gemacht

- Kontakt
  - Eine generische email, welche an viele Projekte geschickt werden kann (Level 0/1 Shot Gun/Targetted Advertisement)
  - TOP-Projekte speziell anschreiben und Anwendungsmöglichkeiten vorschlagen. (Level 2/3 Sniper Rifle/Inside Man)
  - Es werden die Entwickler direkt angeschrieben (BOTTOM-Ansatz)
  - Oder es wird nur der/die Maintainer angeschrieben (TOP/SPREAD-Ansatz)

- Bei Interesse Saros einführen und Hilfe leisten
- Diskussion aufbauen und Kontakt halten
- Anwendungsmöglichkeiten vorschlagen
  - Code Tag: Möglichkeit einem Projektmitglied (z.B. Maintainer) bei der Arbeit zuzuschauen
  - Sitzungen mit mir zum vorstellen von Saros.
  - Code review
  - Code sprint
  - Einen Neuling anlernen
  - ...
- Dabei können sich neue Anwendungsmöglichkeiten als auch fehlende Funktionalität festgestellt werden.



- Absehbare Probleme:
  - Änderung der Arbeitsgewohnheiten oft schwierig
  - Zeitliche Synchronisation
  - Persönliche Ängste
    - Hohe Präsenz
    - Generelle Probleme mit jemanden zusammen zu arbeiten
  - Heterogenität der Werkzeuge
  - Wie findet das Pairing statt
  - .....
- Probleme werden gesammelt

- Fragen erstellen und von Benutzern beantworten lassen (Servlet)
- Probleme in Saros identifizieren und melden
- Probleme des DPP an sich identifizieren
- Absehbare Probleme:
  - Zuordnung der Fragebögen zu den Benutzern

- Theorie (01.07.2009-15.08.2009)
- Projekte auswählen 5-8 (24.07.2009-15.08.2009)
- Projekte anschreiben (01.08.2009, dann iterativ)
  - Emails verfassen
- Diskussion (läuft immer parallel, 4 Monate)
- Saros einführen (läuft immer parallel, 4 Monate)
- Saros evaluieren (läuft immer parallel, 4 Monate)
  - Fragen stellen
  - Daten erfassen
- Arbeit schreiben (01.12.2009) - (30.01.2010)
- Problem
  - Iterativer Prozess
  - Stark abhängig vom OSS Projekt

- Fragen / Anmerkungen

# Vielen Dank!