



Verteilte Paarprogrammierung in Open-Source-Softwareprojekten

Eike Starkmann

Institut für Informatik

FU Berlin

15.10.2010

- Ziele der Arbeit
- Bezugsrahmen
 - Saros
 - Verteilte Paarprogrammierung
 - Open-Source-Softwareprojekte
- Vorgehensweise
 - Forschungsmethode
 - Innovationseinführung
 - Datenauswertung
- Projekte

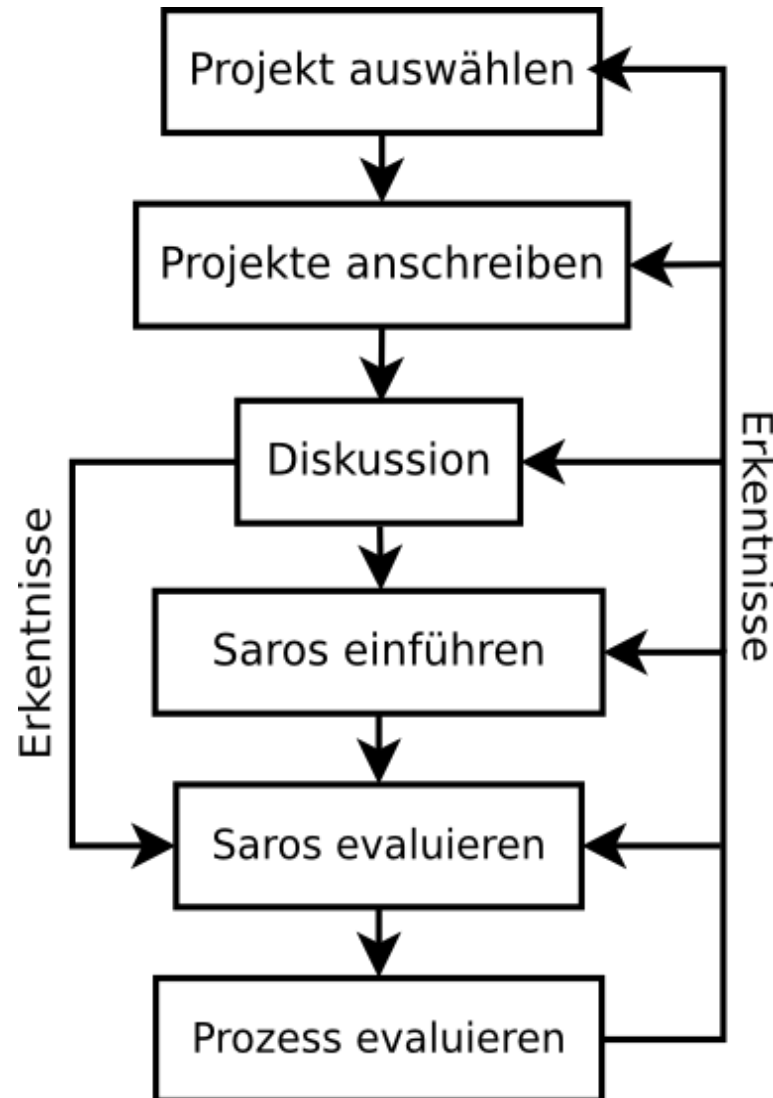
- Ergebnisse
 - Kontaktaufnahme
 - Projekte
 - Saros
- Reflexion des Untersuchungsplanes
- Ausblick

- Saros in OSS Projekten einführen
 - Kontaktaufnahme
 - Projekteigenschaften
 - Saros/DPP einsetzbar?
- Durch Saros wird Arbeitsweise in OSSP 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 den Einsatz von Saros in OSS Projekten?

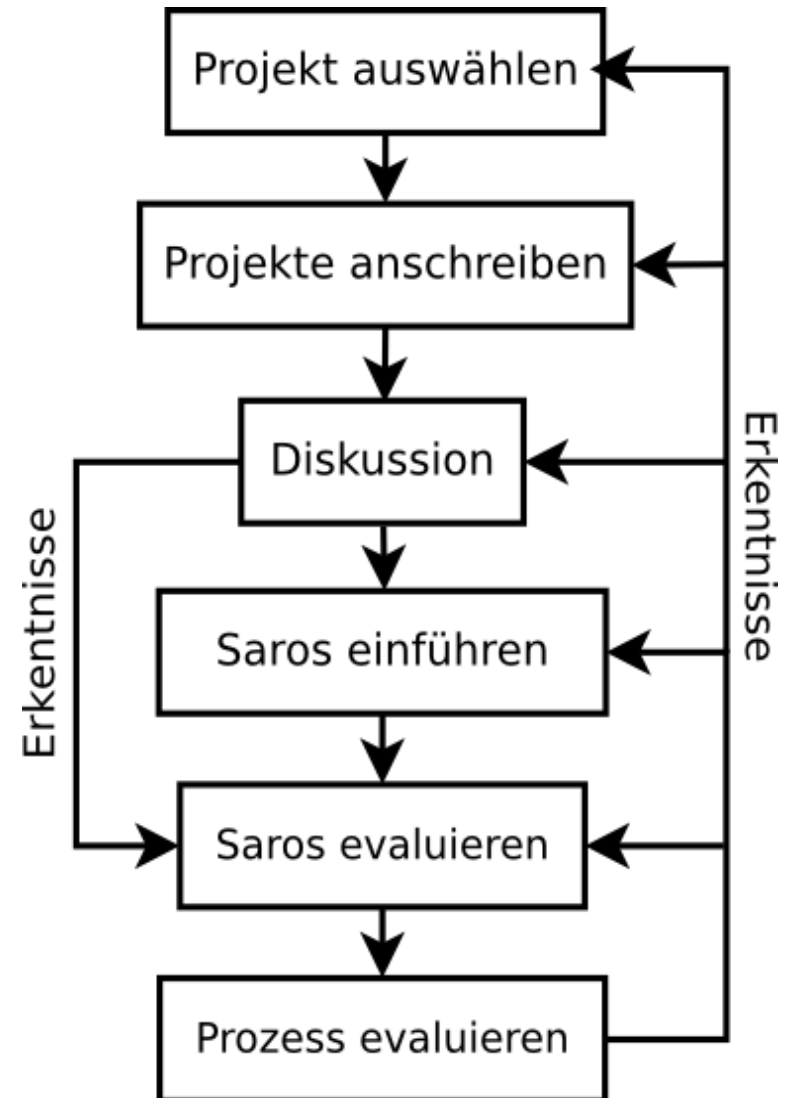
- Eclipse-Plugin
 - Plattformunabhängig
- Ermöglicht verteilte PP
 - Mehrere Personen können arbeiten gleichzeitig an einer Datei arbeiten

- Offener Quelltext
- Kommunikation per Mailingliste
- Versionsverwaltung
- Bugtracker
- Gemeindebasiert (community founded)
 - Von einer Person/kleiner Gruppe initiiert
- Gesponsert (sponsored)
 - Anfang meist nicht öffentlich
 - Produkt wird dann als OSS freigestellt
 - Sponsor ist meistens Firma

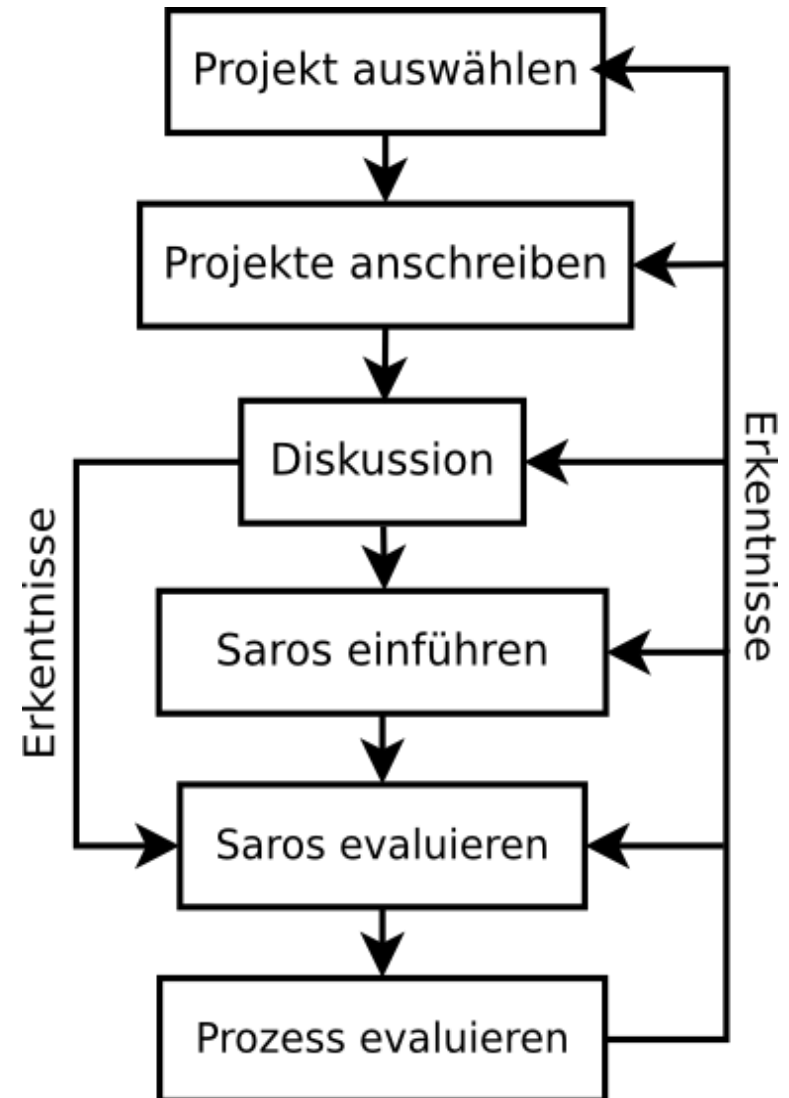
- Handlungsforschung
 - Setzt an konkreten Problemen an und lässt die Ergebnisse im Forschungsprozess in die Praxis umsetzen
 - Iteratives Vorgehen
 - Gleichberechtigter Diskurs zwischen betroffenen Personen und Forscher
- Qualitatives Experiment
 - Keine Überprüfung von vorgefertigten Hypothesen
 - Aufdecken von Strukturen eines Gegenstandes steht im Vordergrund
 - Immer wieder in veränderter Form eingreifen
 - Dialog mit dem Gegenstand



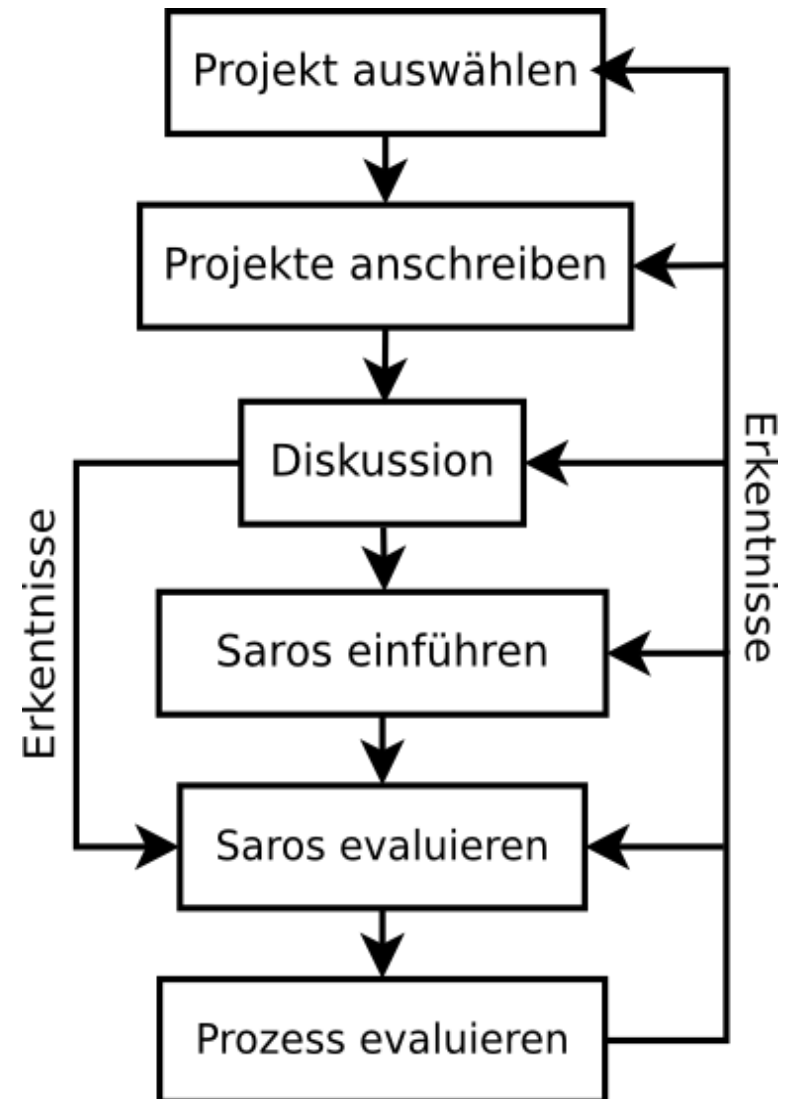
- Projekt auswählen
 - Projekte welcher Struktur eignen sich für Saros bzw. eine Kontaktaufnahme?



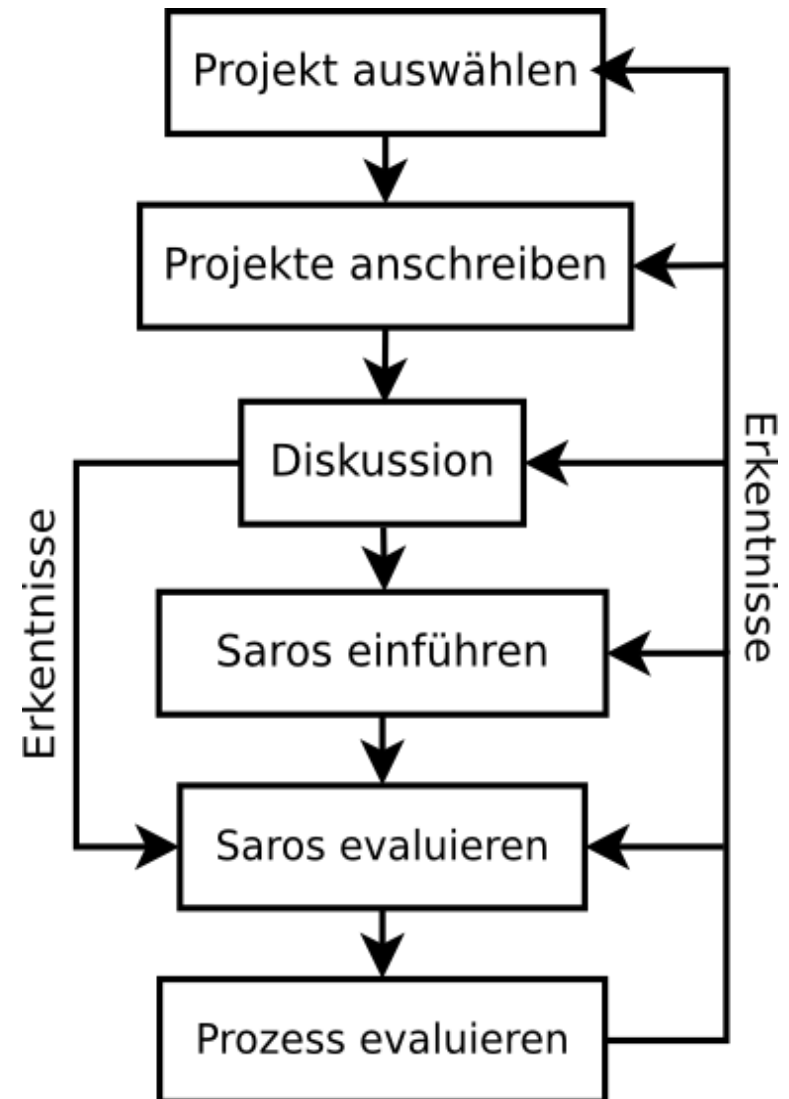
- Projekt anschreiben
 - Welche Kontaktstrategien sind erfolgversprechend?
 - Innovationseinführung



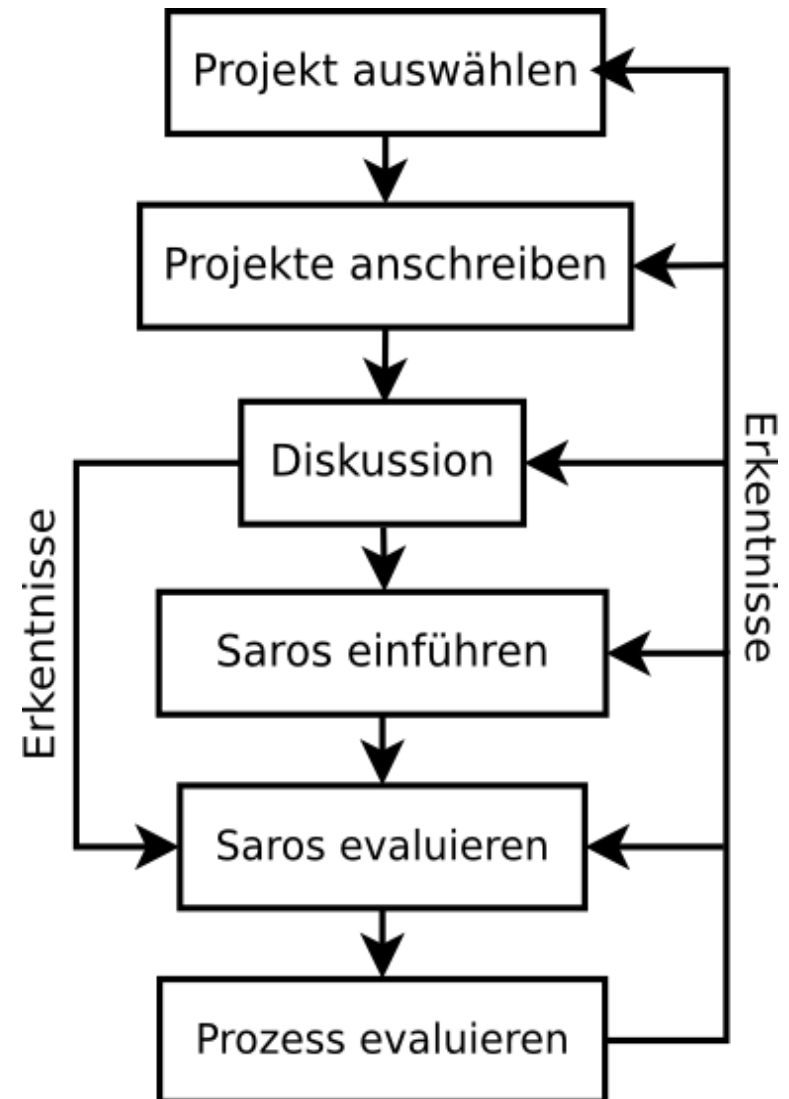
- Diskussion
 - Dialog zwischen Forscher und Entwickler (Handlungsforschung)
 - Anwendungsmöglichkeiten
 - Defekte bzw. fehlende Features von Saros



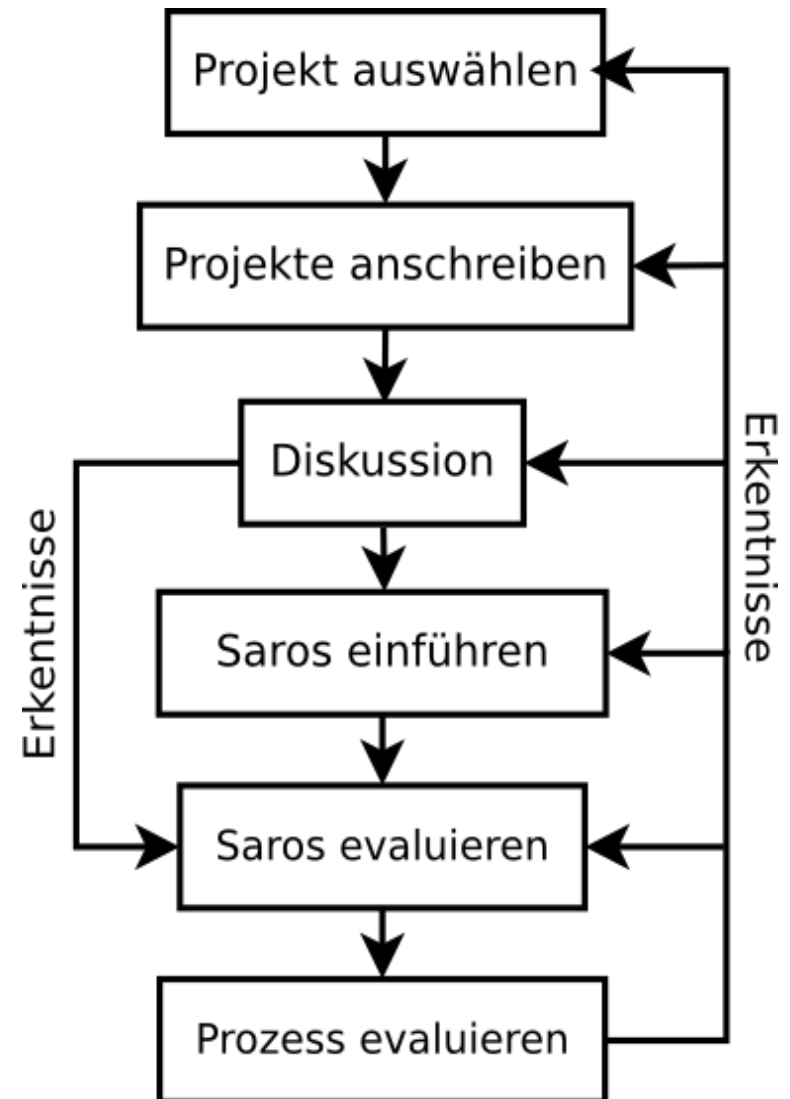
- Saros einführen
 - Saros wird von den Entwicklern getestet
 - Begleitung bzw. Einführung durch mich



- Saros evaluieren
 - Verbesserung von Saros



- Prozess evaluieren
 - Hinterfragung und Bewertung des eigenen Vorgehens
 - Erkenntnisse fließen in die nächste Iteration bzw. in das nächste Projekt



- Welche Kontaktstrategien sind erfolgversprechend?
 - Hängt von der Struktur des Projektes ab (Projektkriterien)
 - Und der gewählten Kontaktart (Kontaktaufnahme)
- Projektkriterien
 - Größe von 5 bis 50 Entwickler (Literatur)
 - Eclipse Nutzung
 - Diskussionen
 - .project Datei im Repository
 - Offenheit
 - Auswahl der Projekte hauptsächlich über sourceforge.net und apache.org

- Kontaktaufnahme
 - Entwickler kontaktieren
 - Nur den/die Maintainer kontaktieren
- Kommunikation
 - Email
 - Vorstellung meiner Person und meines Anliegens
 - Vorteile von Saros und Anwendungsmöglichkeiten in dem Projekt aufzeigen
 - Unterstützung anbieten

- Daten sind hauptsächlich Texte (Email, Chatprotokolle)
- Auswertung mit Hilfe der qualitativen Inhaltsanalyse
- Kategoriensystem zur Strukturierung der Daten und Zusammenfassung dieser
 - Akzeptanz /Ablegung
 - Technische
 - Aussage A
 - Aussage B
 - Konzeptionelle
 - Umfeld bezogene
 - Unklarheiten
 - Technische
 - Konzeptionelle

- Beispiel
 - „Anything which requires live interaction between developers won't work, we all work at different times of day“ [FR2]
 - Umfeld bezogene Ablehnung
 - Konzept unpassend für das Projekt
 - Entwickler arbeiten zu asynchron

- Typo3
 - 5x Zweig (Mailingliste)
 - 4x Zweig (Mailingliste)
 - Typo3 Entwicklung generell (Mailingliste)
 - Dev3 (Mailingliste)
 - Security Team (Maintainer)
- Drupal (Mailingliste)
- Vuze (Maintainer)
- Wordpress (Mailingliste)
- Eclipse-CS (Maintainer)
- Junit (Mailingliste)
- Freenet (Mailingliste)

- OpenBravo (Maintainer)
- OpenProj (Maintainer)
- Tomcat (Mailingliste)
- Adempiere (Forum/IRC)
- Hibernate (Mailingliste)
- Spring (Forum)

- Keine Antwort erhalten:
 - Dev3 (Mailingliste)
 - Generelle Typo3 Entwicklung Liste (Mailingliste)
 - Vuze (Maintainer)
 - OpenBravo (Maintainer)
 - OpenProj (Maintainer)
 - Tomcat (Mailingliste)
 - Eclipse-CS (Maintainer)
- Circa 50% aller kontaktierten Projekte haben geantwortet
- Vuze, OpenBravo und OpenProj sind Corporate Sponsored Open Source Projekte
- Dev3 ist nicht mehr aktiv
- Nur 5 Personen haben Saros wirklich eingesetzt

- Programmiersprache des Projektes
 - Wichtig um die evtl. benutzten Plugins mit Saros zu testen
 - Webseite angelegt mit kompatiblen Plugins
 - Bachelorarbeit führt dies fort
- 11 Projekte wurde per Mailingliste/Forum kontaktiert
 - 2 Projekte antworteten nicht
- 4 Projekte nur über die Maintainer
 - Keines dieser Projekte antwortete
 - Allerdings hatten 3 dieser Projekte keine öffentliche Kommunikation
 - 3 Projekte waren von einer Firma geleitet

- „Schlechte Erfahrungen mit Pair Programming“ war 2.häufigster Ablehnungsgrund
 - Saros sollte nicht mit distributed pair programming gleichgesetzt werden
 - In dem Kontakttext wurde nicht mehr von „distributed pair programming“, sondern von „collaborative development“ gesprochen
- 2 der kontaktierten Mailinglisten antworteten nicht
 - Diese wiesen eine hohe Aktivität auf (>19 Mails/Tag)
 - Anfragen können bei größeren Listen untergehen
- Wenn ein IRC Kanal vorhanden ist, sollte dieser während der Kontaktaufnahme ebenfalls benutzt werden.
- Gezielte Anwendungsvorschläge von Saros erhöhen die Chance auf eine Reaktion bzw. Diskussion

- Organisation
 - Von 4 von einer Firma geleiteten Projekten antwortete nur 1
 - Projekte dieser Art eignen sich weniger für diese Kontaktaufnahme
 - Je mehr Informationen über ein Projekt verfügbar sind, desto wahrscheinlicher ist ein erfolgreicher Kontakt
- Größe
 - Projekte die antworteten und wo eine Diskussion entstand hatten zwischen 23 bis 66 Entwickler

- Eclipse Nutzung
 - „Abhängigkeit von einer IDE (Eclipse)“ war 3. häufigster Ablehnungsgrund
 - Die meisten dieser Aussagen wurden von PHP Entwicklern geäußert
 - Java Entwickler sahen dies nicht als Problem an
- Arbeitsverteilung
 - „Entwickler arbeiten zu asynchron“ war häufigster Ablehnungsgrund
 - Projekt ist geografisch zu verteilt
 - Arbeit ist sehr ungleich verteilt
 - z.B. ein Hauptentwickler erledigt 99% der Arbeit
 - Prozesse wie z.B. ein BugDay können hier entgegenwirken

- Technische Akzeptanz
 - Sinnvolles Produkt
 - Interesse/sinnvoller Vorschlag 12/10 (19,3%)
 - Anwendungsvorschlag 2/2 (3,8%)
 - Vorschlag für Weiterentwicklung 3/3 (5,7%)
 - Schon nach so etwas gesucht 2/2 (3,8%)
 - Gute Lösung (Saros getestet) 3/3 (5,7%)
- Konzeptionelle Akzeptanz
 - Produktivitätssteigerung 1 (1,92%)
 - Sonstiges 1 (1,92%)
- Umfeld bezogene Akzeptanz
 - Produkt passend für das Projekt
 - Generell von Nutzen für das Projekt 1 (1,92%)
 - Anwendungsvorschlag für das Projekt 3/3 (5,7%)

Ergebnisse: Ablehnungsgründe 29/14 (55,7%)

- Technische Ablehnung
 - Kein Sinnvoller Vorschlag
 - Es gibt schon so etwas 1 (1,92%)
 - Mangelhaftes Produkt
 - Abhängigkeit von IDE 4/4 (7,8%)
 - Defekt 5/3 (9,6%)
- Konzeptionelle Ablehnung
 - Kein sinnvolle Arbeitsmethode oder Prozess
 - Schlechte Erfahrungen mit PP 5/3 (9,6%)
- Umfeld bezogene Ablehnung
 - Produkt unpassend für das Projekt
 - Fehlende Funktionalität 1 (1,96%)
 - Muss jeder selber entscheiden 1 (1,96%)
 - Einarbeitungszeit 1 (1,96%)
 - Es wird schon so etwas benutzt 1 (1,96%)

- Zu wenige Entwickler würden das benutzen 3/2 (5,7%)
- Produkt könnte nützlich sein, aber nicht für uns 3/3 (5,7%)
- Umfeld bezogene Ablehnung
 - Konzept unpassend für das Projekt
 - Entwickler arbeiten zu asynchron 6/5 (11,5%)
 - Entwickler arbeiten ohnehin im Büro zusammen 1 (1,96%)
 - Konzept könnte nützlich sein, aber nicht für uns/OSSP 1 (1,96%)

- Es wurde eine Webseite angelegt mit Anwendungsmöglichkeiten von Saros in OSSP
 - Vorurteile gegen DPP sollten so abgebaut werden
 - Weitere Möglichkeiten von Saros wurden erläutert
- Anwendungsmöglichkeiten von Saros (Entwickler)
 - Neueinsteiger anlernen
 - Präsentation und Verbreitung von Wissen
 - Kollaboratives Arbeiten um Patches in den Kern zu integrieren
 - Peer-Review
- 8 Defekte wurden gefunden
 - Einige davon so schwerwiegend, dass das Projekt Saros nicht benutzen konnte

- 5 Feature wurden vorgeschlagen
 - Save Saros sessions for later replay
 - Support Multiple Projekt in one session
 - Remember selections when sharing a part of a project
 - Whiteboard (Diplomarbeit)
 - Chat is missing (Bachelorarbeit)

- Flexibel genug um sich Projekten anzupassen
- Fast 100 Seiten Textmaterial konnten durch das Kategoriensystem gut zusammengefasst werden.
- Iteratives Vorgehen war nur sinnvoll bei Projekten die Saros auch getestet haben
 - Dort strukturiert es das Vorgehen aber zufriedenstellend
- Nur indirekte Antwort auf die Frage „Wie ist Saros in OSSP einsetzbar“
 - Anwendungsmöglichkeiten

- Defekte müssen behoben werden
 - Insbesondere innerhalb des Partial Sharing
 - Verbindungsaufbau sollte verbessert werden
- Projektstrukturen
 - Vereinfachen Projektauswahl
 - Höhere Erfolgsrate
 - Projektstation
 - Stabile Version
 - Anfangsstadium
- Kontaktaufnahme
 - Anwendungsfälle finden und daher Projekte noch intensiver untersuchen
 - z.B. in Hinsicht auf die Arbeitsverteilung innerhalb des Projektes

Vielen Dank!