

# Saros

## Entwicklung und Implementierung eines Saros-Super-Servers

Mehmet Cetinkaya

---

Freie Universität Berlin, Arbeitsgruppe Software Engineering

Februar 2020



1 Saros

2 Entwicklung

3 Offene Probleme

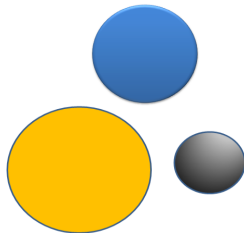
4 Ziele der Arbeit

5 Offene Fragen

6 Literaturverzeichnis



Ein Werkzeug zur verteilten  
Paarprogrammierung  
[https://github.com/  
saros-project/saros](https://github.com/saros-project/saros)



- ▶ Eine Praktik der extreme Programming (XP) [1]
- ▶ XP ist eine Methode der agilen Softwareentwicklung
- ▶ Lösen **einer Aufgabe** von zwei Softwareentwicklern an **einem Computer**
- ▶ Vorteile:
  - ▶ Besserer Entwurf
  - ▶ Weniger Fehler
  - ▶ Eingebauten Code-Review
  - ▶ Vereinigung von Wissen
- ▶ Nachteile:
  - ▶ Hohe Personalkosten
  - ▶ Bei großen Leistungsunterschieden zwischen Partnern

- ▶ Saros ein Werkzeug zur DPP
- ▶ Saros wurde 2006 von Riad Djemili [2] FU entwickelt
- ▶ Stetige Weiterentwicklung von Saros auch auf andere IDE's
- ▶ Unterschiede zu PP:
  - ▶ Softwareentwickler sind örtlich voneinander getrennt
  - ▶ Softwareentwickler sind über eine aktive Netzwerkverbindung miteinander verbunden
  - ▶ Ermöglicht das verteilte Editieren

- ▶ Initiator einer Saros-Sitzung (Host) hat eine privilegierte Rolle
- ▶ Ist Zentraler Akteur im Netzwerk
- ▶ Aufgaben:
  - ▶ Einladen von Teilnehmern
  - ▶ Teilen von Projekten
  - ▶ Dient als Kontaktpunkt für alle Kommunikationen innerhalb einer Sitzung
- ▶ Nachteil:
  - ▶ Bei Ausfall des Hostes wird auch die Sitzung beendet
- ▶ Idee Entwicklung Host-unabhängiger Sitzung, mittels eines Sitzungs-Servers

- ▶ **2014** erster Server-Prototyp Nils Hauke Bussas [3]
  - ▶ Implementierte einen Server-Prototypen als Teil des Saros-Eclipse-Plugins
  - ▶ Grundlegende Funktion *JoinSessionRequests*
  - ▶ Teilen von Projekten nicht möglich
- ▶ **2015** Refactoring des Saros-Codes Arndt Lasarzik [4]
  - ▶ Aufteilung der Saros-Codebasis in *Saros/C*, *Saros/E* und *Saros/I*
  - ▶ Abkapselung des Saros-Kern-Codes von den IDE spezifischen Teilen
  - ▶ Erleichterung für Wartung Weiterentwicklung
  - ▶ Minimierung von Code Redundanz

- ▶ **2016** Entwicklung eines Servers von Denis Washington [5]
  - ▶ Refactoring des Saros-kerns
  - ▶ Erweiterung des ersten Prototypen um das *Non-Host Project Sharing*
  - ▶ Hacken: Nie Vollständig in die Code-Basis integriert
- ▶ **2017** Integration des Servers in die Code-Basis (Saros/S) von Michael Krummrei [6]
  - ▶ Leider nicht fertig gestellt
  - ▶ Mittlerweile veraltete Code-Teile mussten überarbeitet werden
  - ▶ *Non-Host Project Sharing* noch nicht eingepflegt



- ▶ **2019** Fertigstellung des Servers für den Produktiveinsatz von Victor Brekenfeld [7]
  - ▶ Fertigstellung der noch ausstehenden Patches
  - ▶ Überarbeitung des Codes (den Aktuellen Stand gebracht)
  - ▶ Kompatibilitätsprobleme mit der Codebasis behoben
  - ▶ Implementierung einer Konsole für das Teilen von Projekten und das Einladen von Teilnehmern
  - ▶ Die verwendung des Sitzungs-Servers ist nun möglich

- ▶ **Der Server steht für eine Sitzung**
  - ▶ Möglichkeit soll geschaffen werden mehrere Sitzungen zu starten
- ▶ **Rechteverwaltung:** jeder Klient kann Projekte mit dem Server teilen
  - ▶ Implementierung einer Rechteverwaltung auf Sitzungs- und Serverebene
- ▶ **Wiederherstellung des Sitzungssatus:** Bei einer beendeten Sitzung ist dies nicht möglich
  - ▶ Sitzungsserver speichert keine Zustände ab
  - ▶ Nach einem Neustart wird eine neue leere Sitzung gestartet

- ▶ Implementierung eines **Saros-Super-Servers** als übergeordnete Instanz
- ▶ Verwaltung der Projekte aus den Saros-Sitzungen
- ▶ Implementierung einer Rechteverwaltung

- ▶ Wie werden die XMPP-Adressen der dynamisch gestarteten Sitzungs-Server gebildet ?
- ▶ Eine Möglichkeit:
  - ▶ Vom Super-Server genutzten XMPP-Account (JID) wiederzuverwenden und lediglich einen anderen Ressourcennamen zu nutzen z.b JID/01
- ▶ Wenn Adressen gebildet worden sind wie werden sie den Teilnehmern bekannt gemacht?
  - ▶ Teilnehmer melden sich selbst auf für eine Saros-Sitzung an

# Fragen?

Saros Entwicklung Offene Probleme Ziele der Arbeit **Offene Fragen** Literaturverzeichnis



- [1] Kent Beck 1999  
Extreme Programming Explained: Embrace Change (Aw Professional)
- [2] Riad Djemili 2006  
Entwicklung einer Eclipse-Erweiterung zur Realisierung und Protokollierung verteilter Paarprogrammierung  
Diplomarbeit  
*Freie Universität Berlin, Inst. für Informatik*
- [3] Nils Hauke Bussas 2014  
Entwicklung eines Server-Prototypen für Saros  
Bachelorarbeit  
*Freie Universität Berlin, Inst. für Informatik*

- [4] Arndt Lasarzik 2015  
Refaktorisierung des Eclipse-Plugins Saros für die Portierung  
auf andere IDEs  
Bachelorarbeit  
*Freie Universität Berlin, Inst. für Informatik*
  
- [5] Denis Washington 2016  
Entwicklung und Evaluation eines unabhängigen  
Sitzungsservers für das Saros-Projekt  
Masterarbeit  
*Freie Universität Berlin, Inst. für Informatik*

- [6] Michael Krummrei 2017  
Überführung des Saros-Server in die Codebasis  
Bachelorarbeit  
*Freie Universität Berlin, Inst. für Informatik*
  
- [7] Victor Brekenfeld 2019  
Saros-Server: Realisierung Nutzer-unabhängiger Sitzungen für  
den Produktiveinsatz  
Masterarbeit  
*Freie Universität Berlin, Inst. für Informatik*



Vielen Dank für ihre  
Aufmerksamkeit !

