



Integration von Versionsverwaltungssystemen in Saros

Andreas Haferburg

Abschlusspräsentation Bachelorarbeit



Saros Demo

Saros Demo

Replikation von Zuständen

- Verzeichnisbaum
- Dateiinhalte
- Eclipse Dokumente
- Anzeige

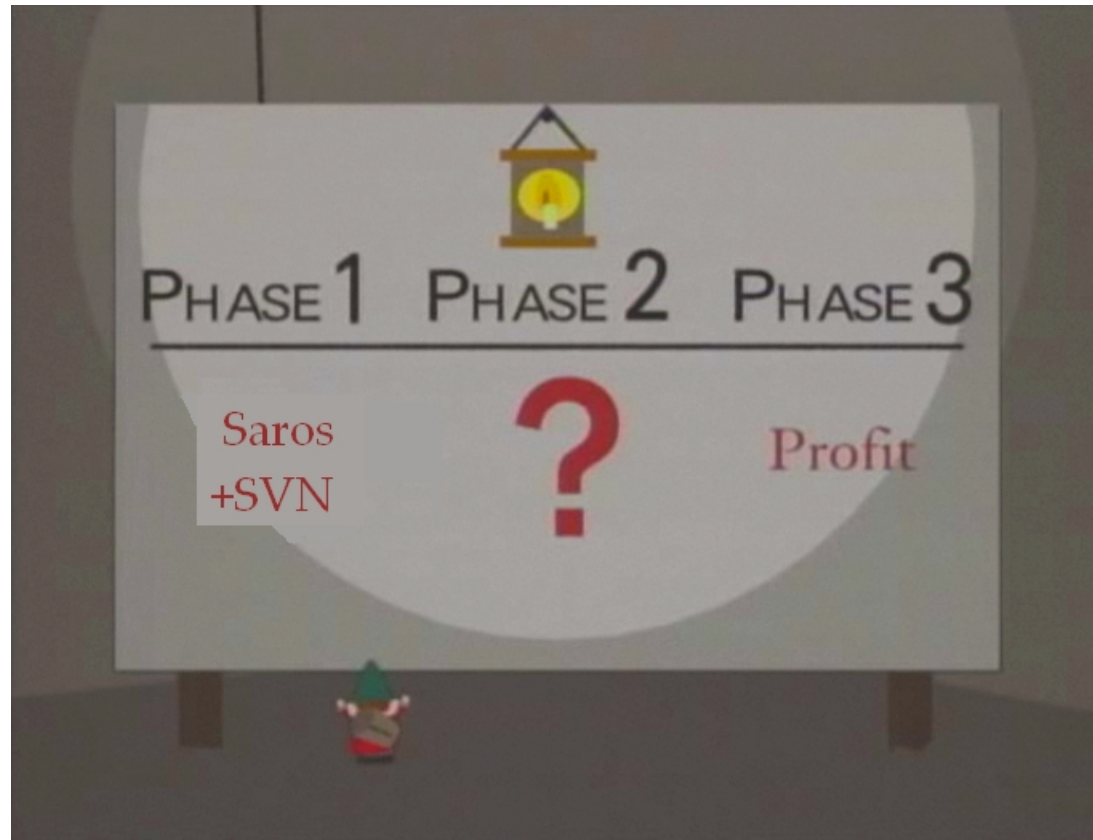
Subversion Demo

Subversion Demo

SVN URL + Revision = VC state (SyncInfo)



VCS Integration in Saros - Motivation



VCS Integration in Saros - Motivation

Checkout während der Einladung

Häufig verwendete Operationen wie Update und Switch replizieren

→ Base version replizieren

→ Voraussetzung für korrekte Funktionsweise von Operationen wie Merge oder Compare



Saros bisher

Replikation von Zuständen

- Verzeichnisbaum
- Dateiinhalte
- Eclipse Dokumente
- Anzeige

VCS Integration in Saros

Replikation von Zuständen

- Verzeichnisbaum
- Dateiinhalte
- Eclipse Dokumente
- Anzeige
- VC state von Dateien und Verzeichnissen



Replikation von Zuständen im Allgemeinen

Initialisierung

Aktualisierung (update)

Replikation von Zuständen in Saros

Initialisierung

→ Einladungsprozess erweitern

Aktualisierung (update)

→ Activities erzeugen, versenden, anwenden

Initialisierung des VC state

Während der Einladung VC state bei eingeladenen Teilnehmern mit dem Host synchronisieren

Initialisierung des VC state

Während der Einladung VC state bei eingeladenen Teilnehmern mit dem Host synchronisieren:

- Wenn das Projekt von VCS verwaltet wird, übermittelt der Host mit der Liste der Projektdateien auch den VC state.

Initialisierung des VC state

Während der Einladung VC state bei eingeladenen Teilnehmern mit dem Host synchronisieren:

- Wenn das Projekt von VCS verwaltet wird, übermittelt der Host mit der Liste der Projektdateien auch den VC state.
- Der eingeladene Teilnehmer wendet entsprechende Operationen des VCS an um den VC state zu replizieren.

Aktualisierung (updates) des VC state

Wenn sich der VC state bei Drivern ändert, dann muss er bei den anderen Teilnehmern ebenfalls geändert werden



Aktualisierung (updates) des VC state

Wenn sich der VC state bei Drivern ändert, dann muss er bei den anderen Teilnehmern ebenfalls geändert werden:

- Änderungen erkennen
- Änderungen analysieren
- Anweisungen an andere Teilnehmer senden
- Änderungen replizieren



Aktualisierung (updates) des VC state

Wenn sich der VC state bei Drivern ändert, dann muss er bei den anderen Teilnehmern ebenfalls geändert werden:

- Änderungen erkennen
- Änderungen analysieren
- Anweisungen an andere Teilnehmer senden („Activity“)
- Änderungen replizieren

Neue Klasse VCSActivity

```
ISarosSession sarosSession = ...;  
IResource file = workspaceRoot.getFile(new Path("../Math.java"));  
IActivity update = VCSActivity.update(sarosSession, file, "8395");
```



Weitere Features

-Erweiterbare Architektur

Offene Probleme

-Verteilte Versionsverwaltungssysteme wie git, Mercurial, Bazaar