



Verbesserung der Stabilität von Saros durch Urgrundanalyse

Konzeptvorstellung Masterarbeit

Sebastian Starroske

Datum: 23.08.2012

Gliederung

- 1) Einleitung
- 2) Zeitplan
- 3) Geplantes Vorgehen
- 4) Probleme / Risiken

Fokus der Arbeit: Konsistenz

- Bei allen Teilnehmern einer Saros Sitzung werden die gleichen Informationen / Daten angezeigt
- Bei allen Teilnehmern liegen die gleichen Daten vor

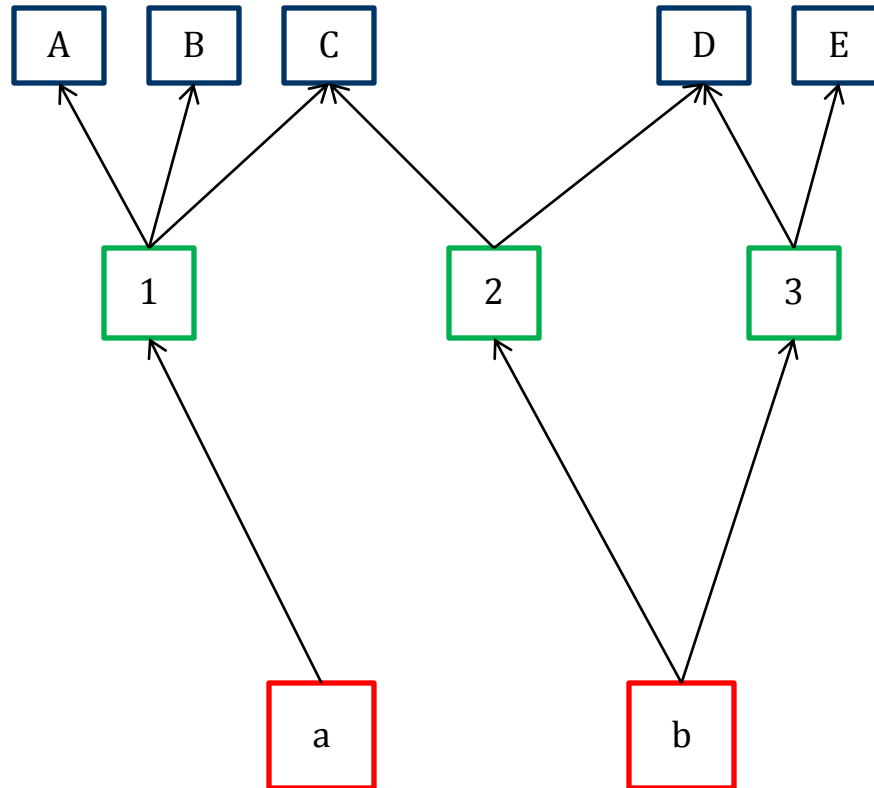
Warum?

- Inkonsistenzen stören die Grundfunktionen von Saros und führen oft zum Abbruch einer Sitzung

- Jedem Fehler / Ereignis geht eine Kette von Ursachen voraus
- Analyse von bestehenden Fehlern bis hin zur Wurzel des Problems

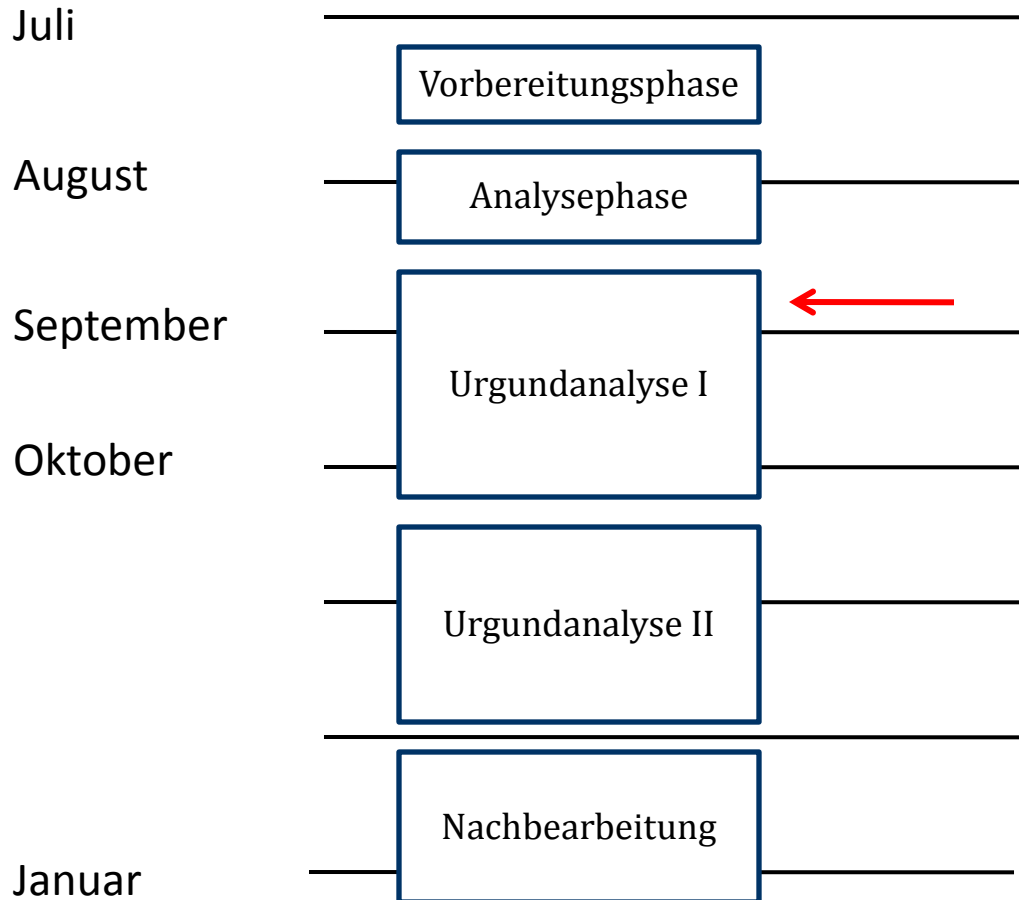
Wurzel = Urgrund

Zweck: Fehler nicht nur behandeln sondern das Auftreten weiterer Fehler in der Zukunft vorbeugen



- Konsistenz von Saros verbessern
- Alle aktuellen Inkonsistenztypen nennen und beschreiben
- Eintrittswahrscheinlichkeit und Auswirkung bestimmen
- Urgrundanalyse für die schwerwiegensten Typen
- Lösungen für die gefundenen Urgründe finden
- Nice-To-Have: Anleitung zur Fehleranalyse in Saros

- 1) Einleitung
- 2) Zeitplan**
- 3) Geplantes Vorgehen
- 4) Probleme / Risiken



- 1) Einleitung
- 2) Zeitplan
- 3) Geplantes Vorgehen**
- 4) Probleme / Risiken

Ausgangspunkt: SourceForge Bugtracker und eigene Beobachtungen

Clustering

Pro Cluster:

- Beheben der Bugs
 - Finden von tiefer liegenden Defekten
-

- Beheben der Urgünde
- Analyse / Rückschluss

- Testen (falls möglich automatisiert), wann ein Fehler das erste Mal auftritt
- Analysieren, ob ein Fehler zeitweilig „untergetaucht“ ist
- Vergleichen von offenen und geschlossenen Bug Einträgen mit ähnlichen Eigenschaften
- Quellcodeanalysen (Suche nach Mustern)
- Einträge im Reviewboard / Gerrit
- Analyse Patchgröße
- Gibt es wiederkehrende Probleme?

- 1) Einleitung
- 2) Zeitplan
- 3) Geplantes Vorgehen
- 4) **Probleme / Risiken**

Umgang mit nicht nachvollziehbaren / rekonstruierbaren Bugs:

- Bei dem Clustern berücksichtigen
- Gibt es verwandte Bugs, die nachvollziehbar sind?

Wie kann ich nachweisen, dass ich einen solchen Bug behoben habe?

Risikobewertung

- Wie sollten Tragweite und Häufigkeit eingeteilt werden?
- Wie ermittle ich daraus den Risikowert?

Fehlende Refferenzen

- Wie wird Urgundanalyse in anderen Softwareprojekten durchgeführt?
- Gibt es nützliche Werkzeuge / Methoden?

Komplexe Struktur von Saros:

- Hohes Maß an Nebenläufigkeit
- Testen / Reproduzieren von Fehlern sehr schwierig / aufwändig

Es gibt zu wenig tatsächliche Fehler im Bereich „Konsistenz“

Maßnahme:

- Anderen Bereich wählen

Es werden keine Urgründe gefunden:

Maßnahmen:

- Clustering überarbeiten

Fragen?

A Case Study in Root Cause Defect Analysis.pdf

<http://users.ece.utexas.edu/~perry/work/papers/DP-00-icse-rca.pdf>

Root Cause Analysis for Beginners

https://servicelink.pinnacol.com/pinnacol_docs/lp/cdrom_web/safety/management/accident_investigation/Root_Cause.pdf

CMMi - Causal Analysis and Resolution (CAR)

<http://www.software-quality-assurance.org/cmmi-causal-analysis-and-resolution.html>