

# Beseitigung von Stolpersteinen im Saros- Entwicklungsprozess

Holger Hans Peter Freyther

# Ziel der Arbeit

- Entfernung von Stolpersteinen.
- Um den Einstieg für neue Entwickler zu erleichtern.
- Um das Arbeiten an Saros angenehmer machen.
- Um Änderungen effektiver vornehmen zu können.
- Um mehr Vertrauen bei Änderungen zu haben.

# Was ist ein Stolperstein?

# Stolpersteine



# Stolpersteine



# Stolpersteine

Ein Stolperstein ist ein Hindernis, das ein effektives Arbeiten an Saros erschwert. Dies kann durch Mangel an Informationen, falschen Informationen oder unnötigen manuellen Aufgaben entstehen.

# Ausgangssituation

# Ausgangssituation

- Verantwortlichkeit ist an Beschäftigung geknüpft.
- Wechsel von Mitarbeitern nicht fliegend.
- Studenten verlassen das Projekt nach der Arbeit.
- Kein Kernteam mit erfahrenen Entwicklern.



# Entscheidungen – Stolperstein und Lösung

„Wichtige Entscheidungen... werden anderswo nicht festgehalten.“

Entscheidungen werden ad-hoc getroffen und stehen folgenden Entwicklergenerationen nicht direkt zur Verfügung.

Vorschlag Soll-Verhalten zu dokumentieren.



# Webseite – Stolperstein und Lösung

„Ich weiß nicht, wie ich die Webseite editiere.“

Technik erschwert das Editieren von Inhalt. Der Inhalt ist veraltet und wichtige Informationen fehlen.

Eine andere Bachelorarbeit soll dieses Thema bearbeiten.



# Kommunikation – Stolperstein und Lösung

„Ich lese die Mailingliste nicht.“

Zu viele E-Mails von Systemen an die Entwicklerliste, es bleibt kein Raum für Diskussionen.

Erstellung der ddp-robot Liste. Mehr Raum für Diskussionen auf der Mailingliste.



# Verwaister Code - Stolperstein und Lösung

Bei Änderungen entstand verwaister Code. Auf den ersten Blick ist nicht zu erkennen welche Pfade benutzt werden oder nicht.

Moose half mir Kandidaten zu finden, es lieferte jedoch zu viele falsche Positive für einen automatischen Betrieb.



# Einstieg – Stolperstein und Lösung

Die Webseite beschrieb den Durchsichtsprozess nicht vollständig. Das Erstellen des nötigen Kontos für das Review-Board ist nicht dokumentiert.

Erstellen einer ausführlichen Anleitung, die Anmeldung beim Werkzeug für Durchsichten geschieht durch OpenID. Aus Sicherheitsgründen muss der neue Entwickler in eine neue Gruppe aufgenommen werden um eigene Änderungen machen zu dürfen



# Durchsicht – Stolperstein und Lösung

„I wish SVN had stashing.“

Eine Änderung zur Durchsicht zu stellen ist ein manueller Prozess. Das Eingehen auf Kritik ist schwer möglich. Die Änderung in SVN kann anders als die Änderung bei der Durchsicht sein. Änderungen können wurden auch vorbei an der Durchsicht ins SVN gestellt.

Umstellung von SVN auf Git, Review-Board auf Gerrit. Git Umstellung mit Werkzeug des KDE Projektes. Änderungen werden durch Gerrit zum Zeitpunkt der Durchsicht getestet.



# Architektur – Stolperstein und Lösung

Java Ausnahmen durch Abbau einer Sitzung, z.B. in einem Stresstest, aber auch bei den normalen Ende-zu-Ende Tests.

Umstellung von einer Liste von Singletons zu Exemplaren, die an eine Sitzung gebunden sind. Den Aufbau einer Sitzung und die Erzeugung der Objekte wird mit EasyMock und PowerMock getestet.



# Entwickler Dokumentation – Stolperstein...

Es existiert wenig Architektur Dokumentation. Die Dokumentation auf der Webseite ist in der Regel veraltet.

Benutzung von JTourBus um Architektur im Quelltext zu beschreiben. Wiederbelebung von JTourBus, Schreiben von neuen Touren basierend auf meiner Erfahrung. Anfänge von Anti-Mustern.





# Fazit

- Mailinglisten bieten mehr Raum zur Diskussion.
- Umstieg auf Git und Gerrit automatisiert den Durchsichtsprozess.
- JUnit Tests werden bereits zum Zeitpunkt der Durchsicht ausgeführt.
- Dokumentation hilft neuen Entwicklern beim Einstieg.

# Kritik

Wenig Empirie, viel Technik

Ist E-Git mit dem SVN Team Modus vergleichbar?

Wie viel einfacher ist der Einstieg?

Wie viel mehr Diskussionen existieren auf der Mailingliste?

Wie viel robuster ist Saros?

# Ausblick

Ziel sollte es sein externe Entwickler langfristig zu binden..

Verantwortlichkeiten definieren und entweder im Team oder externe Entwickler verteilen.

STF-Test Stabil bekommen.

**Fragen? Diskussion?**