

Erleichterung des Einstiegs für Entwickler in Saros mittels Analyse und Beseitigung von Hürden

Konzeptvorstellung Diplomarbeit

Gliederung

- Motivation
- Ziel
- Vorgehen
- Developer Documentation & Probleme
- Website & Probleme
- sonstige Artefakte & Probleme
- Angestrebte Änderungen

Motivation


*„Projects that fail to attract and, more importantly, to retain new contributors rarely get beyond a name and a few lines of code stored on Sourceforge.“
(Ghosh and Prakash, 2000; Capiluppi et al., 2003)*

*„socializing new members an essential ingredient in the long-term survival of OSS projects.“
(Ducheneaut, 2005)*

Ziel

*Neuen Entwicklern den Einstieg in Saros
so einfach wie möglich zu gestalten
(mittels guter Einstiegsdokumentation)*

Vorgehen

- Probleme & Verbesserungsvorschläge aus dem Projekt
- Ist-Zustand erfassen → neue Probleme
- Interviews mit Beteiligten führen → noch mehr Probleme
- 
 - Lösungen für Probleme suchen und abwägen
 - Änderungen durchführen
 - formative heuristische Evaluation → neue Probleme
- Evaluation ?

Developer Documentation

Saros Developer Documentation

Saros Team

Copyright © 2012 Holger Hans Peter Freyther

This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Table of Contents

1. Introduction

- Overview

- About this manual

2. Rules

- Code Rules

- Review Rules

3. Installing and Configuring Plugins

- Installing the SVN Plugin

 - Installation of Subclipse (textual)

 - Installation of Subclipse (graphical)

- Installing the Git Plugin

 - Installation of EGit (textual)

 - Installation of EGit (graphical)

 - Known issues

- Installing the JTourbus Plugin

 - Installation of JTourBus (textual)

 - Installation of JTourBus (graphical)

4. Sourcecode and Review

- Version Control with Git

 - Introduction to Git

 - Git clone with EGit (textual)

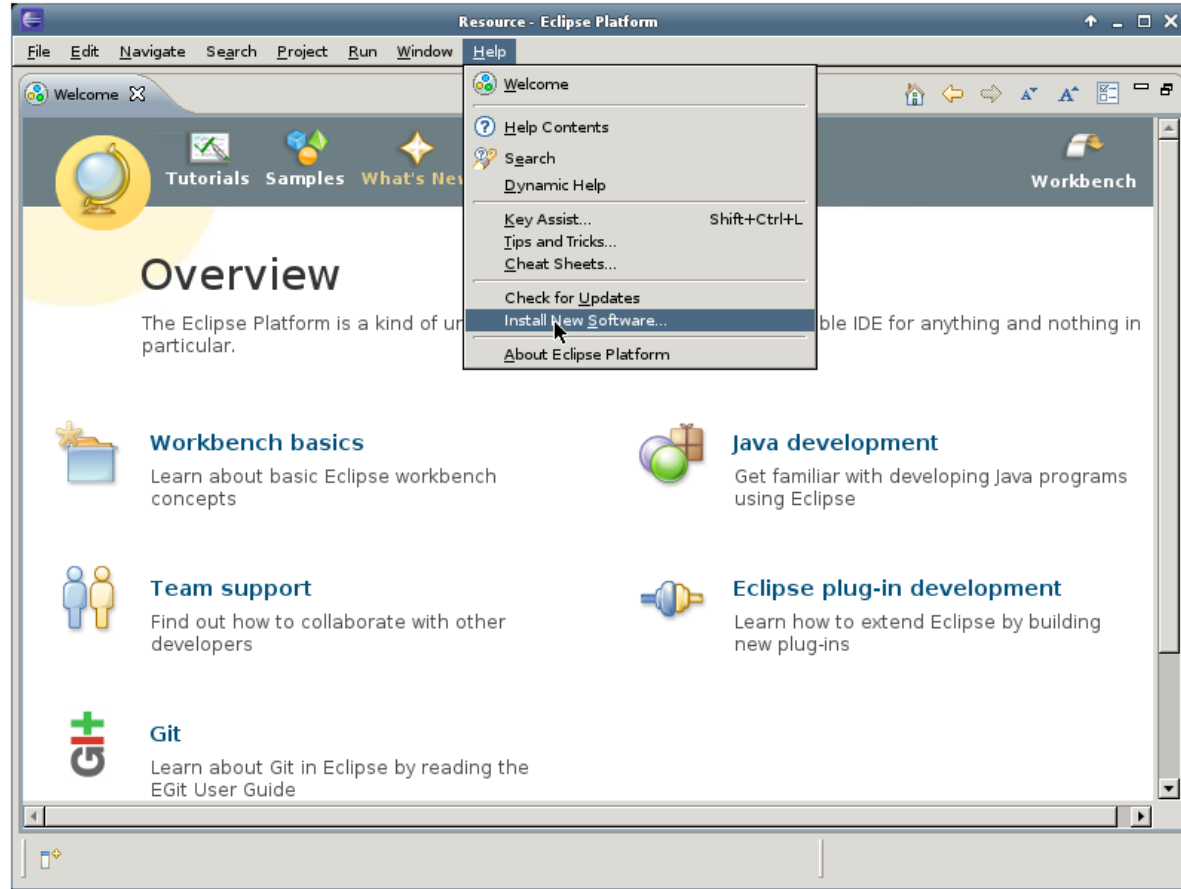
 - Git clone with EGit (graphical)

- Configuring Git for Gerrit

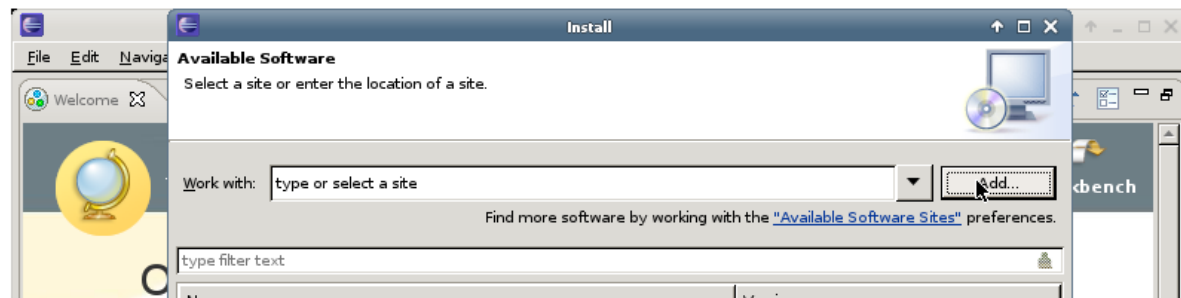
Installation of Subclipse (graphical)

Procedure 3.2. Installation of Subclipse

1. Open the New Software Dialog



2. Press the Add Button to add a site



Probleme: Developer Documentation

- bekannte Probleme
 - Thema **Testing** fehlt
 - Schritte fehlen
 - Schritte gruppiert
 - Zusatzinformationen
 - Unnötige Schritte
 - Tipps ohne Anwendung
 - Screenshots & Grafiken
- neue Probleme
 - Inhalte fehlen
 - Reihenfolge „unnatürlich“
 - Überschriften nicht klar genug

Website

Navigation

→ Add content

→ Administration

User menu

→ My account

→ Log out

Get Started

[View](#) [Edit](#)

In this section we want to offer you an easy introduction to **start with the development** on Saros. There are different ways to start depending on how much experience you have.

- [Developer Documentation](#)

If you are a **new developer**, you **ought to read this document**. You will find information about how to check-out the project and **how to start to develop**.

- [Learn about the code](#)

The **Eclipse Plug-in JTourBus** helps you to **discover the Saros code**. It is an easy way to learn about how Saros works technically.

Probleme: Website

- bekannte Probleme
 - Inhalte inkonsistent
 - Inhalte veraltet
 - Usability
- neue Probleme
 - Redundanz
 - Überschriften nicht klar genug
 - Inhalte nicht gefunden
 - Grafiken

sonstige Artefakte

- STF User Manual
- STF Quickstart
- Mailingliste
- 3 Tracker in SourceForce
- JTourBus Touren
- Quelltext
- X-Arbeiten

Angestrebte Änderungen

- Developer Documentation
 - Texte aktualisieren
 - Zusatzinformationen geben
 - neue Themen aufnehmen
 - Grafiken überarbeiten
 - erklärende Grafiken hinzufügen
 - Reihenfolge ändern
 - in Website integrieren
- Quelltext
 - Exemplarischer Test !!!
 - plugins.xml kommentieren
- Website
 - Übersicht über alle Artefakte
 - Texte aktualisieren
 - Texte konsister gestalten
 - Grafiken überarbeiten
 - Usability
- sonstige Artefakte
 - Kern in Website integrieren
 - Kern in Devel.Doc. integrieren
 - Tracker aufräumen

Allgemein:

- Informationsstreuung eindämmen
- wichtige Informationen zum richtigen Zeitpunkt vermitteln

Ich habe Fragen!

- Evaluation?
- Titel? (*„Erleichterung des Einstiegs für Entwickler in Saros mittels Analyse und Beseitigung von Hürden“*)

Haben Sie Fragen?

- ... ?

- ... ?

- ... ?

Danke für Ihre Aufmerksamkeit

Referenzen

- Capiluppi, A., P. Lago and M. Morisio (2003): Evidences in the Evolution of OS Projects through Change Log Analyses. In Proceedings of the 3rd Workshop on Open Source Software Engineering, Portland OR, pp. 19–24.
- Ducheneaut, N. (2005): Socialization in an Open Source Software Community: A Socio-Technical Analysis. Computer Supported Cooperative Work (CSCW), 14(4): pp. 323–368
- Ghosh, R. and V.V. Prakash (2000): The Orbiten Free Software Survey. First Monday 5(7).
- OpenClipart.org (2013): http://openclipart.org/detail/170151/loop_arrow
Zugriff: 11.03.2013

Erleichterung des Einstiegs für Entwickler in Saros mittels Analyse und Beseitigung von Hürden

Konzeptvorstellung Diplomarbeit

hallo, mein name ist lev stejngardt, ich bin im 13. semester und schreibe meine diplomarbeit in der arbeitsgruppe software engeneering. heute möchte ich ihnen vorstellen, woran ich arbeiten werde und was der inhalt meiner arbeit sein wird

Gliederung

- Motivation
- Ziel
- Vorgehen
- Developer Documentation & Probleme
- Website & Probleme
- sonstige Artefakte & Probleme
- Angestrebte Änderungen

zu anfang werde ich versuchen meine motivation und das ziel meiner arbeit erklären
dann mein vorgehen beschreiben und die wichtigsten artefakte vorstellen, die einen entwickler beim einstieg in saros begleiten. und die schwächen dieser artefakte
zum schluß zeige ich ihnen noch meine angestrebten änderungen
der punkt „evaluation“ ist hier ganz bewusst ausgelassen

Motivation

*„Projects that fail to attract and, more importantly, to retain new contributors rarely get beyond a name and a few lines of code stored on Sourceforge.“
(Ghosh and Prakash, 2000; Capiluppi et al., 2003)*

*„socializing new members an essential ingredient in the long-term survival of OSS projects.“
(Ducheneaut, 2005)*

ich fange mal mit einem frei übersetzten zitat an:

*Projekte die es nicht schaffen **neuen Entwickler** zu aquirieren und zu halten, werden selten mehr sein, als ein Namen und ein paar Zeilen Quelltext.*

DAS FÜHRT UNS UNWEIGERLICH ZUM 2. ZITAT:

*Das **Eingliedern** von **neue Mitgliedern** ist ein wesentlicher Bestandteil des Langzeiterfolgs von **OSP***


und genau dieses „eingliedern“, das funktioniert bei OSP größtenteils durch interaktion mit anderen mitgliedern und durch eine gute einsteigerdokumentation. darum ist das ziel meiner arbeit....

Ziel

*Neuen Entwicklern den Einstieg in Saros
so einfach wie möglich zu gestalten
(mittels guter Einstiegsdokumentation)*

neuen entwicklern den einstieg in saros so einfach
wie möglich zu gestalten
einfach = effektiv (sie sollen alles notwendige
erhalten) und effizient (so schnell wies geht)

Vorgehen

- Probleme & Verbesserungsvorschläge aus dem Projekt
- Ist-Zustand erfassen → neue Probleme
- Interviews mit Beteiligten führen → noch mehr Probleme
- 
 - Lösungen für Probleme suchen und abwägen
 - Änderungen durchführen
 - formative heuristische Evaluation → neue Probleme
- Evaluation ?

- zu anfang habe ich eine sammlung von möglichen probleme aus projekt erhalten
- dann istzustand: wie? zuerst fachliteratur zu info-struktur, info-design, textdesign, web usability gelesen und aus prinzipien → guidelines. dann habe ich mich in die perspektive eines entwicklers begeben und habe mich selber eingearbeitet (saros-quellen gelesen) dabei habe ich den ist-zustand erfasst (= positives und negatives notiert). positives & negatives anhand der guidelines identifiziert. **(Heur.Eval)** → Dokument mit Verbesserungsvorschlägen. warum H.E.? Beobachtung unmöglich keine Objekte, Fragebogen unzureichend (**+** für quantifizierung bekannter aspekte, **–** für entdeckung von problemen)
- Gespräche mit Team geführt; Interview mit SWT-Projekler (die sollten kleinen bug fixen); mögliche Probleme von Projekt teilweise verifizieren
- Festgestellte Probleme zusammentragen
- Lösungen für Probleme erarbeiten und gegeneinander abgewegt
- Änderungen durchführen (iterativ, formative Heu.Eval. nach jeder iteration nach o.g. prinzipien)

Click to add an outline

Developer Documentation

developer documentation ist im rahmen einer bachelorarbeit (beseitigung von stolpersteinen im saros entwicklungsprozess) von holger freyther entstanden.

es handelt sich dabei um eine schritt-für-schritt anleitung für angehende entwickler.

sie beschreibt alle notwendigen schritte vom einrichten der entwicklungs Umgebung in eclipse bis zum einchecken von überprüfem quellcode

Saros Developer Documentation

Saros Team

Copyright © 2012 Holger Hans Peter Freyther

This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit <http://creativecommons.org/licenses> to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Table of Contents

1. Introduction

- Overview
- About this manual

2. Rules

- Code Rules
- Review Rules

3. Installing and Configuring Plugins

Installing the SVN Plugin

- Installation of Subclipse (textual)
- Installation of Subclipse (graphical)

Installing the Git Plugin

- Installation of EGit (textual)
- Installation of EGit (graphical)
- Known issues

Installing the JTourbus Plugin

- Installation of JTourBus (textual)
- Installation of JTourBus (graphical)

4. Sourcecode and Review

Version Control with Git

- Introduction to Git
- Git clone with EGit (textual)
- Git clone with EGit (graphical)

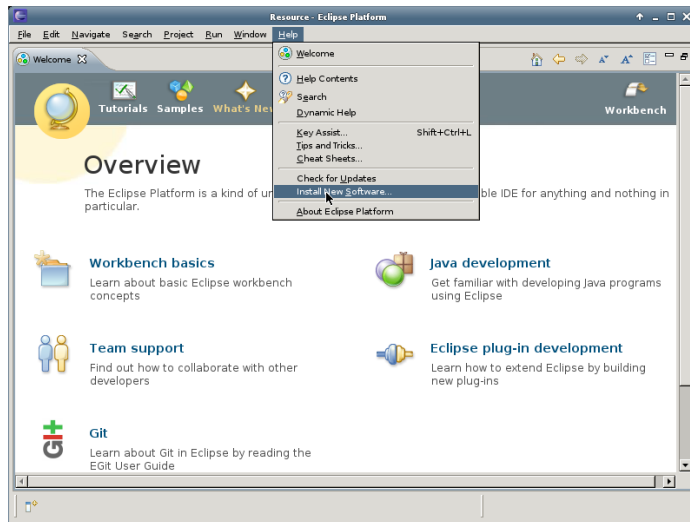
Configuring Git for Gerrit

teil des inhaltsverzeichnis; beschreibt notwendige schritte; linear; auffällig: (textual) und (graphical); jeder Schritt in 2 versionen erklärt. textual für erfahrene benutzer; graphical für anfänger

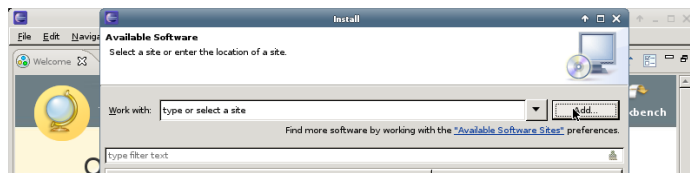
Installation of Subclipse (graphical)

Procedure 3.2. Installation of Subclipse

1. Open the New Software Dialog



2. Press the Add Button to add a site



sehr genau erklärt; mit richtigem „clickguide“. sehr fehlervermeidend.

Probleme: Developer Documentation

- bekannte Probleme
 - Thema **Testing** fehlt
 - Schritte fehlen
 - Schritte gruppiert
 - Zusatzinformationen
 - Unnötige Schritte
 - Tipps ohne Anwendung
 - Screenshots & Grafiken
- neue Probleme
 - Inhalte fehlen
 - Reihenfolge „unnatürlich“
 - Überschriften nicht klar genug

bekannt = aus projekt & heuristische evaluation, neu = interviews

bekannt:

- mit keinem Wort wird Testen erwähnt, obwohl TDD vorgeschlagen; swpler: „wir wissen nicht wann & wie testen)
- manche Schritte fehlen → Fehlerquelle!; manche sind gruppiert → können nicht gefunden werden!
- Zusatzinformationen fehlen z.B. Warum Subclipse → Unklarheiten im systemverständnis entstehen!
- unnötige Schritte → aufgebläht+Verlust des Ziels+leser wird unnötige strapaziert!
- Tipps ohne Anwendungsmöglichkeit (Cherry Pick, steht im Anhang) → Tipps ignoriert!
- Screenshots zuschneiden → Seite kürzer, nur wichtige Infos; Konzepte mit Grafiken erklären → Steigert Verständnis b. Leser

neue:

- Inhalt fehlen: Requirements, Git, mögliche Fehler
- Abfolge der Schritte entspricht nicht der real Welt
- Überschriften sagen nicht was sich dahinter verbirgt

Click to add an outline

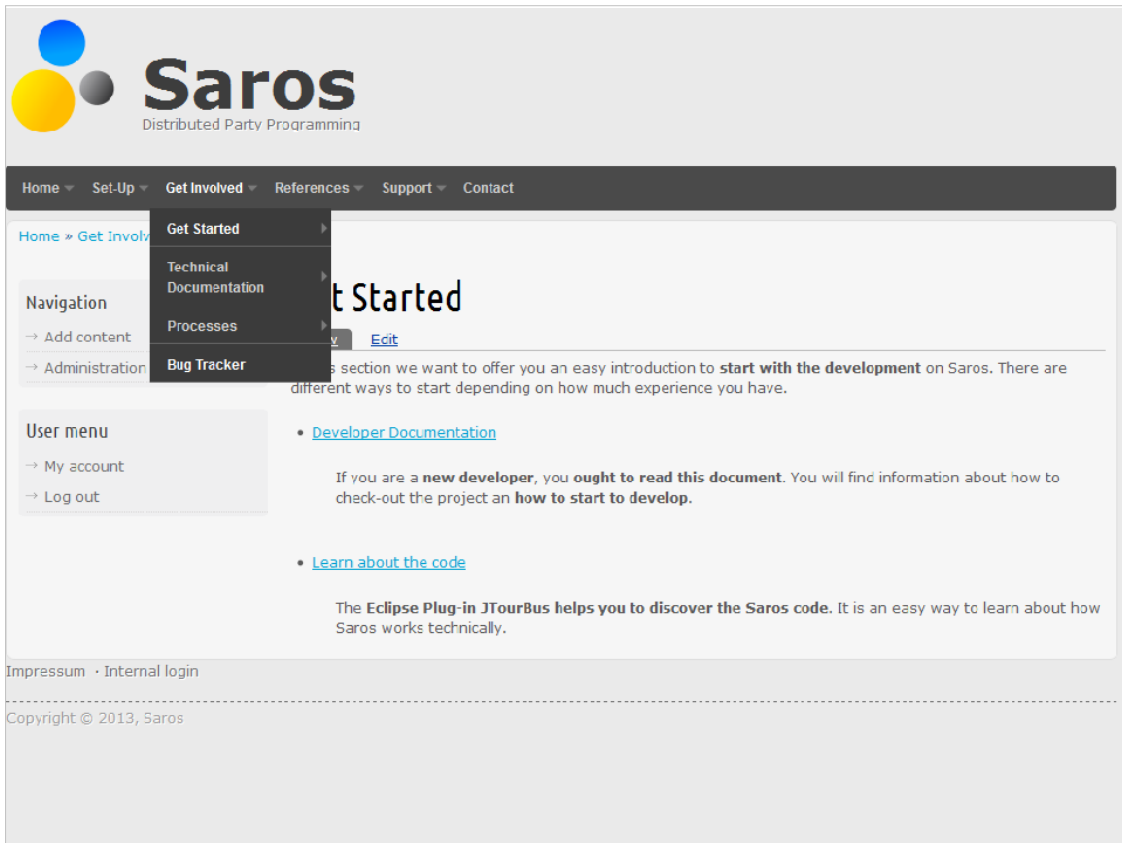
Website

ich rede hier von der drupal website, nicht von der twiki-seite.

neue website im rahmen einer bachelorarbeit von sascha kretzschmann entstanden.

dabei „übertrug“ der autor die inhalte von der alten seite in die neue seite; er berücksichtigte dabei prinzipien info-struktur, info-design, text-design (wie meine), allerdings hat er die inhalte nicht überprüft! er schlägt selber eine überprüfung vor...

ebenfalls schlägt er selber eine heuristische evaluation der usability vor



logo, tagline, hauptnavigation, drop-down-menu, inhalt, breadcrumb-nav... eine website halt, wie sie halt so ist

Probleme: Website

- bekannte Probleme
 - Inhalte inkonsistent
 - Inhalte veraltet
 - Usability

- neue Probleme
 - Redundanz
 - Überschriften nicht klar genug
 - Inhalte nicht gefunden
 - Grafiken

alte Probleme:

- inhalte inkonsistent: manchmal gibt's 1) stichpunkte 2) fließtext 3) formatierung 4) grafiken → sieht aus als ob 30 menschen ohne konzept inhalte eingefügt
- inhalt veraltet: 1) links auf alte seite 2) alte grafiken 3) inaktueller inhalt
- usability: suchfunktion, besuchte links kennzeichnen, news sektion, Mischung pdf und html, texte nicht scanbar, svg statt raster

neue Probleme:

- Redundanz „es gab gefühlt mehrere Orte, an denen ähnliche Sachen beschrieben worden sind“
- Überschriften nicht aussagekräftig
- Inhalte werden nicht gefunden, weil versteckt
- Grafiken: keine Aussage (Riesengrafik) od. nicht verstanden

sonstige Artefakte

- STF User Manual
- STF Quickstart
- Mailingliste
- 3 Tracker in SourceForce
- JTourBus Touren
- Quelltext
- X-Arbeiten

- STF user manual: erklärt das STF zu genau, wie man damit arbeitet zu ungenau
- STF Quickstart: nicht auf seite!!! sollte in DD überführt werden
- Mailingliste → viele hintergrundinformationen und diskussionen, aber vieles davon veraltet
- 3 tracker mit vielen tickets
- jtourbustouren enthalten virtuelle touren durch den quelltext
- quelltext mit kommentaren
- x-arbeiten

Angestrebte Änderungen

- Developer Documentation
 - Texte aktualisieren
 - Zusatzinformationen geben
 - neue Themen aufnehmen
 - Grafiken überarbeiten
 - erklärende Grafiken hinzufügen
 - Reihenfolge ändern
 - in Website integrieren
- Quelltext
 - Exemplarischer Test !!!
 - plugins.xml kommentieren
- Website
 - Übersicht über alle Artefakte
 - Texte aktualisieren
 - Texte konsistent gestalten
 - Grafiken überarbeiten
 - Usability
- sonstige Artefakte
 - Kern in Website integrieren
 - Kern in Devel.Doc. integrieren
 - Tracker aufräumen

Allgemein:

- Informationsstreuung eindämmen
- wichtige Informationen zum richtigen Zeitpunkt vermitteln

diese folie wird einfach vorgelesen. falls mir spontan was einfällt, sag ich mehr (wird wohl eher nicht der fall sein)

1) DD 2) website 3) quelltext 4) sonstige artefakte 5) allgemein

Ich habe Fragen!

- Evaluation?
- Titel? („*Erleichterung des Einstiegs für Entwickler in Saros mittels Analyse und Beseitigung von Hürden*“)

- Evaluation nicht möglich, weil
 - Probleme nicht konkret genug erfasst (Interviews geben nur einen Teil wider) → die kann man jedoch evaluieren
 - es keine Evaluatoren gibt 1) ich nicht 2) Team geht nicht, weil es alles weiß 3) SWPLer gibt's nicht 4) BetreuerIdee: irgendeinen Anfänger zum „durchklicken“ der DD suchen
- Titel? zur Erinnerung: „*Erleichterung des Einstiegs für Entwickler in Saros mittels Analyse und Beseitigung von Hürden*“

Haben Sie Fragen?

- ... ?
- ... ?
- ... ?

haben sie noch fragen oder
verbesserungsvorschläge?

Danke für Ihre Aufmerksamkeit

Referenzen

- Capiluppi, A., P. Lago and M. Morisio (2003): Evidences in the Evolution of OS Projects through Change Log Analyses. In Proceedings of the 3rd Workshop on Open Source Software Engineering, Portland OR, pp. 19–24.
- Ducheneaut, N. (2005): Socialization in an Open Source Software Community: A Socio-Technical Analysis. Computer Supported Cooperative Work (CSCW), 14(4): pp. 323–368
- Ghosh, R. and V.V. Prakash (2000): The Orbiten Free Software Survey. First Monday 5(7).
- OpenClipart.org (2013): http://openclipart.org/detail/170151/loop_arrow
Zugriff: 11.03.2013