

Belegarbeit GIT

Fabian Goerge
Sami Ede
Richard Peters
Jens Meise

24. November 2012

Inhaltsverzeichnis

1	<u>Einführung und Ziel der Arbeit</u>	3
1.1	Einführung	3
1.2	Ziel	3
2	<u>Textuelle Beschreibung</u>	3
3	<u>Modellierung</u>	3
3.1	Use-Case-Modell	3
3.1.1	Use-Case-Szenarien (Storys)	3
3.2	Domänenmodell	4
3.3	Ausgewählte Zustandsdiagramme	4
3.4	Glossar	4
4	<u>Erfahrung aus der Teamarbeit</u>	4
5	<u>Schlussbetrachtung</u>	4

1 Einführung und Ziel der Arbeit

1.1 Einführung

FERTIGSTELLUNG BIS 09.12.!!!!!!! und wehe nicht!!!!!!!
- PDF vollenden (gemeinsame Punkte bearbeiten)
- Presentation fertigstellen (16.12. / 22.12.)

Sami hier gehts los!!!

GIT ist ein Software zur Versionsverwaltung...

1.2 Ziel

Sami... Git modellieren, mit den Modellierungsarten auseinandersetzen

2 Textuelle Beschreibung

Git ist eine freie Software zur verteilten Versionsverwaltung von Dateien. Git hilft, wenn mehrere Programmierer an einem Projekt arbeiten, indem es verschiedene Features vereint: - Git ermöglicht eine nicht-lineare Entwicklung; das heißt, es ist möglich, an verschiedenen „Positionen“ des Quelltextes zu arbeiten, und diese im Nachhinein automatisch zu vereinen („mergen“). - des Weiteren bietet Git die Möglichkeit eines lokalen Arbeitens. Jeder User besitzt eine lokale Kopie des kompletten Repositories (inklusive der Versionsgeschichte), so dass auch ohne Anbindung in ein Netzwerk gearbeitet werden kann. Dies ermöglicht sehr schnelles Arbeiten! Dies bedeutet auch gleichzeitig Multi-Backups! Jeder User speichert die komplette Versionsgeschichte! - Mehrere Möglichkeiten, Repositories abzugleichen; Git besitzt ein eigenes Protokoll, und unterstützt unter anderem http, https, ftp und rsync Übertragungen. - Versionsgeschichte wird kryptografisch gesichert, so dass es nicht möglich ist, Änderungen an der Versionsgeschichte vorzunehmen, ohne dass sich der Name der Revision ändert.

Des Weiteren bietet Git ein Webinterface.

3 Modellierung

3.1 Use-Case-Modell

Fabian...Use case Zeichnung einfügen!

3.1.1 Use-Case-Szenarien (Storys)

- Happy day Szenarien und Optionale Verzweigungen (Abweichungen vom Happy Day) (Richard)

3.2 Domänenmodell

- Klassendiagramme, Objekte, Verknüpfungen, Abhängigkeiten (Jens, Richard)

3.3 Ausgewählte Zustandsdiagramme

- Verschiedene Prozesse beschreiben (Quelltext bearbeiten, Branches mergen) (Sami, Fabian)

3.4 Glossar

- Fachwörter werden erklärt (Repository, Branch, bla...) (Gemeinsam im Laufe der Latexerstellung)

4 Erfahrung aus der Teamarbeit

gemeinsam gewählter Prozess... (Gemeinsam)

5 Schlussbetrachtung

Zusammenfassend kann man sagen das hier noch nicht so viel passiert ist ;) (Gemeinsam)