



Plat_
FORMS The web
development
platform
comparison

Konzeptvortrag Masterarbeit: Betrachtung der Modularität von Plat_Forms-Lösungen

Andreas Franz

14.04.2011

Gliederung

- Motivation
- Modularität
- Metriken
- Methoden
- Erfassung der Messdaten
- Zusammenfassung

Motivation

- Plat_Forms Wettbewerb 2011
 - Java, Perl, PHP, Ruby, (JavaScript)
 - Teams zu je 3 Personen
 - gleiche Anforderungen
 - Webanwendung für Konferenzverwaltung
 - Ermittlung von Vor- und Nachteilen der Plattformen

Motivation

- Ziel dieser Masterarbeit
 - Betrachtung der Modularität
 - Evaluation und automatisierte Erfassung von Metriken
 - Auswertung (möglicher) Plattformunterschiede

Modularität

- Zerlegung eines komplexen Produkts
- Kriterium: Information hiding
- Organisation wird einfacher
 - Bessere Verständlichkeit → Wartbarkeit
 - Parallele Entwicklung
 - Wiederverwendung von Modulen
- Module interagieren
- Verhalten nach außen genau wie ohne Modularisierung

Metriken

- Zwei häufig gebrauchte Begriffe in Zusammenhang mit Modularität
- Kohäsion
 - „Zusammenhang“ innerhalb eines Moduls
 - Modul soll abgegrenzte Zuständigkeit haben
- Kopplung
 - Verknüpfung zwischen Modulen
 - Wunsch nach:
 - starker Verbindung innerhalb eines Moduls
 - loser Kopplung zu anderen Modulen

Methoden(Kohäsion)

- Lack of cohesion in methods (Hitz/Montazeri)
 - graphentheoretischer Ansatz
 - Komponentenzugehörigkeit
 - Aufruf einer Methode durch die andere
oder
 - gemeinsam genutzte Variable
 - mehrere Komponenten weisen auf Möglichkeit zur Aufteilung hin
 - Teile ohne jeglichen funktionellen Zusammenhang

Methoden(Kopplung)

- Kopplung zwischen Objekten
 - Methoden- und Variablenaufruf
 - Art der Parameter hat Einfluss auf Stärke der Kopplung
- Kopplung durch Vererbung
 - Zunahme der Kopplung mit Tiefe der Vererbungshierarchie
 - Depth in inheritance tree (DIT)
 - Methoden und Klassenvariablen der Superklasse werden vererbt
 - Abhängigkeiten werden „weitergereicht“

Erfassung der Messdaten

- statische Codeanalyse mit vorhandenen Parsern
- mögliches Problem:
 - Parser für bilden Metriken unterschiedlich ab, oder sind nicht vorhanden
- Parsererzeugung mittels ANTLR
 - abstrahieren des Quellcodes
 - Grammatik bildet Sprache ab
 - Quellcode als abstrakten Syntaxbaum abbilden

Zusammenfassung

- Erfassung vorgestellter Metriken
 - Indikatoren für Kopplung und Kohäsion
 - automatisierte, statische Codeanalyse
- Auswertung des Einflusses
- gegebenenfalls Plattformunterschiede aufzeigen