



Modularitätsbetrachtung von Webanwendungen im Rahmen des Plat_Forms Wettbewerbs

Abschlussvortrag Masterarbeit

Andreas Franz

26.04.2012

Agenda

1. Motivation
2. Grundlagen
3. Aufgabenstellung
4. Vorgehen
5. Entwicklung einer Metrik
6. Betrachtung anhand der Architektur
7. Zusammenfassung und Ausblick

Agenda

1. Motivation
2. Grundlagen
3. Aufgabenstellung
4. Vorgehen
5. Entwicklung einer Metrik
6. Betrachtung anhand der Architektur
7. Zusammenfassung und Ausblick

Motivation

- Plat_Forms Wettbewerb 2011
- 4 Java, 4 PHP, 4 Ruby, 3 Perl, (1 JavaScript)
 - Teams zu je 3 Personen
 - „*Conferences and Participants*“
- Untersuchungen bezüglich der Plattformcharakteristiken (Benutzbarkeit, Funktionalität, Größe, Struktur)
- In dieser Masterarbeit: Betrachtung der Modularität
- Kenntnis der Eigenschaften unterstützt bei Auswahl

Agenda

1. Motivation
2. Grundlagen
3. Aufgabenstellung
4. Vorgehen
5. Entwicklung einer Metrik
6. Betrachtung anhand der Architektur
7. Zusammenfassung und Ausblick

Grundlagen

• Modularität

- Zerlegung eines komplexen Produkts
- Kriterium: Information hiding (Parnas [Par72])
- Organisation wird einfacher
 - Bessere Verständlichkeit → Wartbarkeit
 - Parallele Entwicklung
 - Wiederverwendung von Modulen
- Module interagieren
- Verhalten nach außen genau wie ohne Modularisierung

Grundlagen

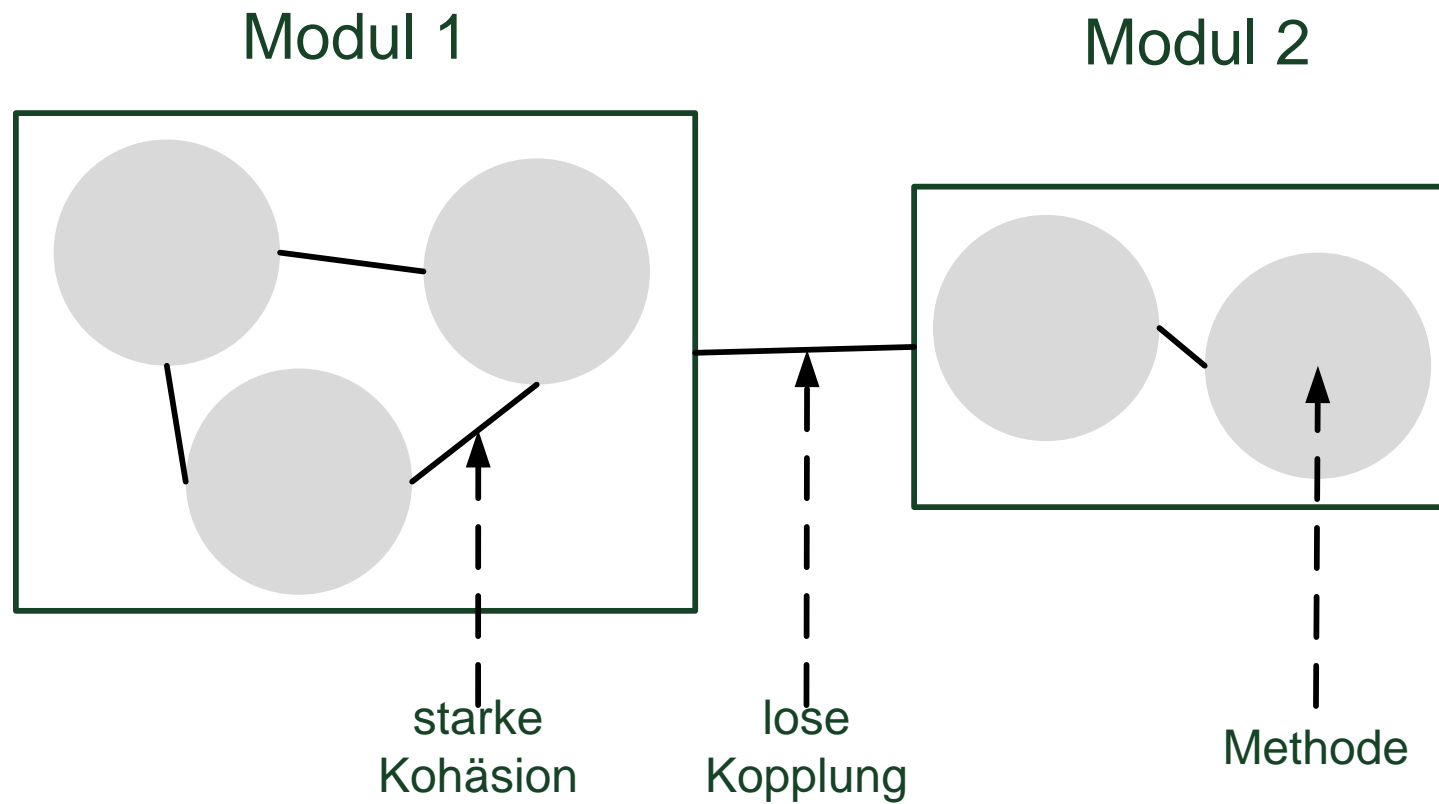
- **Kohäsion**

- 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

Grundlagen



Agenda

1. Motivation
2. Grundlagen
- 3. Aufgabenstellung**
4. Vorgehen
5. Entwicklung einer Metrik
6. Betrachtung anhand der Architektur
7. Zusammenfassung und Ausblick

Aufgabenstellung

- Modularität der Lösungen des Plat_Forms Wettbewerbs *charakterisieren, analysieren* und zu *vergleichen*
- Gewährleistung eines fairen Vergleichs
- Allgemeine Anwendbarkeit
- Erweiterbarkeit

Agenda

1. Motivation
2. Grundlagen
3. Aufgabenstellung
- 4. Vorgehen**
5. Entwicklung einer Metrik
6. Betrachtung anhand der Architektur
7. Zusammenfassung und Ausblick

Vorgehen

- Evaluierung etablierter Konzepte auf ihre Anwendbarkeit und Aussagekraft
- Betrachtung der Struktur von Webanwendungen
 - Frameworks
 - Architekturmuster

Agenda

1. Motivation
2. Grundlagen
3. Aufgabenstellung
4. Vorgehen
- 5. Entwicklung einer Metrik**
6. Betrachtung anhand der Architektur
7. Zusammenfassung und Ausblick

Entwicklung einer Metrik

- Vorgehen zur Entwicklung einer Pseudometrik (nach Ludewig und Lichter [LL07])
 1. Identifikation des darzustellenden Aspekts
 2. Modellierung des darzustellenden Aspekts
 3. Festlegung einer Skala für die Metrik
 4. Entwicklung einer Berechnungsvorschrift
 5. Entwicklung der Messvorschriften für alle Größen, die in der Berechnungsvorschrift verwendet werden
 6. Anwendung und Verbesserung der Metrik

Schritte 1 bis 3

1. Identifikation des darzustellenden Aspekts
 - Modularität anhand der Kopplung (zwischen MVC-Komponenten aus Sicht von Templates)
2. Modellierung des darzustellenden Aspekts

`category.parent.name`

`conference.creator.name`

Zulässiger Variablenzugriff nach dem Gesetz von Demeter

3. Festlegung einer Skala für die Metrik
 - Verhältnisskala (zwischen 0 und 1)

Schritte 4 bis 5

4. Entwicklung einer Berechnungsvorschrift

- Verstoß gegen das Gesetz von Demeter gegenüber der Gesamtzahl von Variablenzugriffen.
- 0 → kein Verstoß gegen die Gesetzmäßigkeit
- 1 → alle Variablenzugriffe verstoßen dagegen

5. Entwicklung der Messvorschriften für alle

Größen, die in der Berechnungsvorschrift

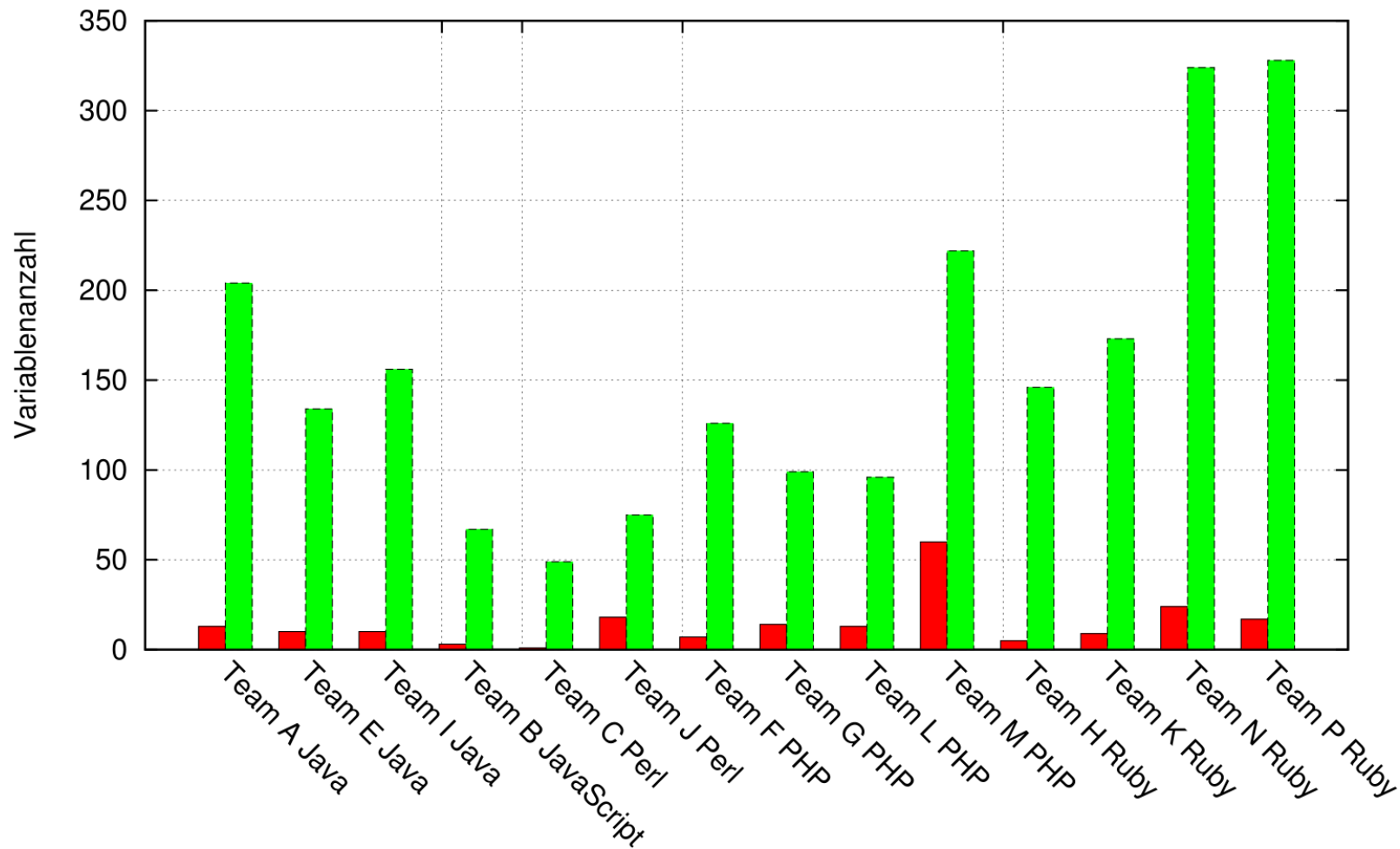
verwendet werden: Aufschlüsselung der Betrachtung

Schritt 6

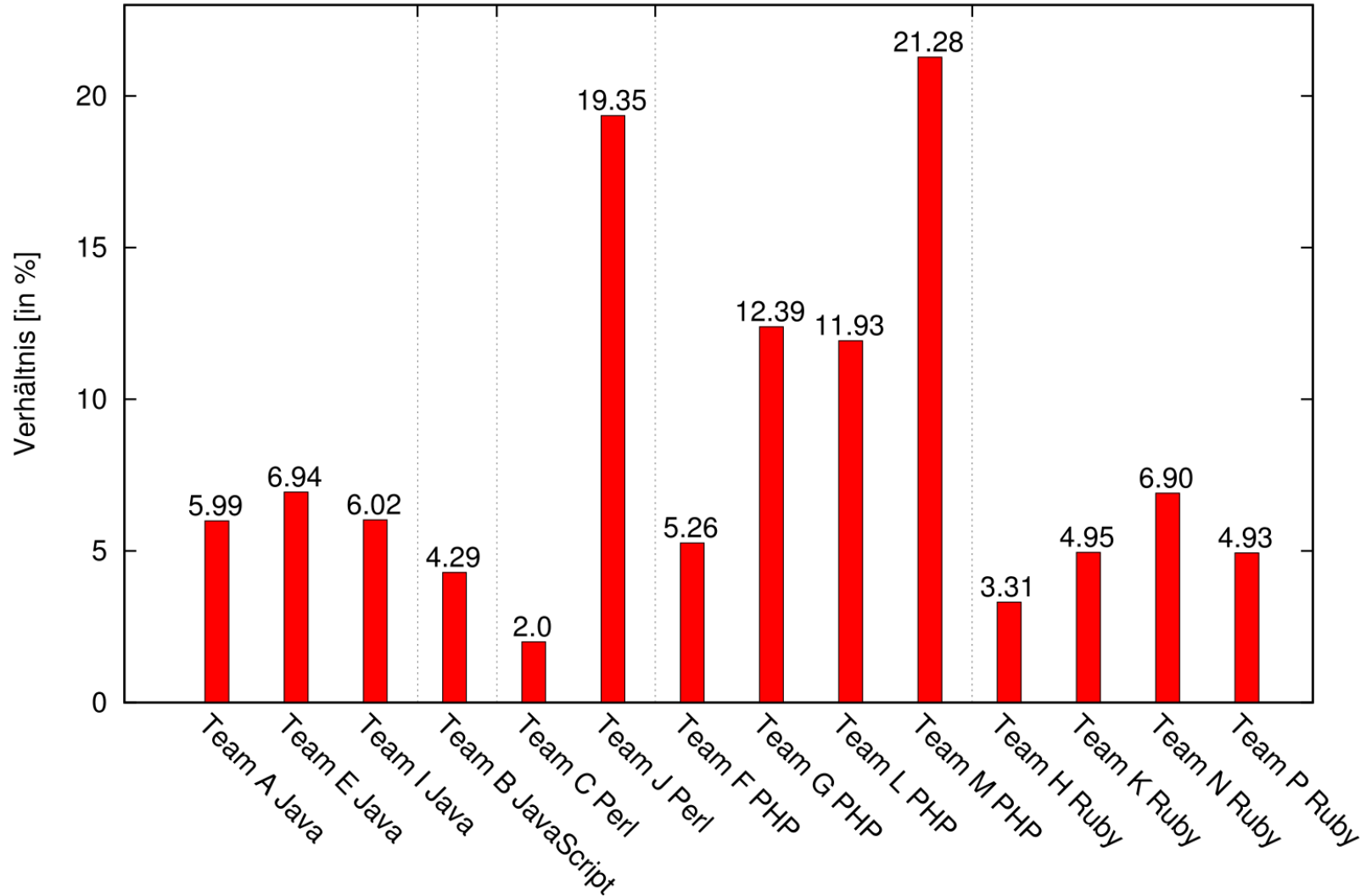
6. Anwendung und Verbesserung der Metrik

- Beschreibung des Vorgehens
- Analyse zentraler Eigenschaften (differenziert, vergleichbar, reproduzierbar, verfügbar, relevant, rentabel, plausibel)

Ergebnisse



Ergebnisse



Bewertung der Ergebnisse

- Eine gesicherte Modularität findet sich vor allem bei den Plattformen Ruby und Java
- Die PHP-Lösungen weisen eine deutlich stärkere Kopplung und damit verbundene schlechtere Modularität auf, als die der Java- und Ruby-Teams
- Unglücklicherweise kann für Perl keine getroffen werden, weil sich die zwei ausgewerteten Teams stark unterscheiden

Agenda

1. Motivation
2. Grundlagen
3. Aufgabenstellung
4. Vorgehen
5. Entwicklung einer Metrik
- 6. Betrachtung anhand der Architektur**
7. Zusammenfassung und Ausblick

Betrachtung anhand der Architektur

- Auffälligkeit: Verwendung von MVC-Frameworks, aber keine korrekte Einhaltung bei der Verwendung
- Aufschlüsselung und Untersuchung der Verstöße

Ergebnisse

- diverse Verstöße gegen das Muster
- Eine plattformweit konsistente Eigenschaft
 - Perl-Teams verwenden HTML::FormHandler
 - Vermischung von View- und Controller-Belangen

Bewertung der Ergebnisse

- Verstoß gegen MVC-Muster kann negativen Einfluss auf Modularität haben
- Vermischung von Zuständigkeiten machen Wartung und Lesbarkeit schwieriger

Agenda

1. Motivation
2. Grundlagen
3. Aufgabenstellung
4. Vorgehen
5. Entwicklung einer Metrik
6. Betrachtung anhand der Architektur
7. Zusammenfassung und Ausblick

Zusammenfassung und Ausblick

- Entwicklung einer Metrik und Untersuchung der Qualität dieser
- Anwendung auf die Lösung und Auswertung
- Außerdem Betrachtung eines Architekturmusters
- Anwendbarkeit konnte gezeigt werden
- statistische Überprüfung der Gültigkeit der Ergebnisse, speziell wegen der Unvollständigkeit der Lösungen
- Problem: hoher Anteil manueller Arbeit (Parser?)
- Problem: Unvollständigkeit der Lösungen

Quellen

- [Par72] PARNAS, D. L.: On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053–1058, 1972.
- [LL07] LUDEWIG, J. und H. LICHTER: *Software Engineering: Grundlagen, Menschen, Prozesse, Techniken*. Dpunkt-Verl., 2007.