technische universität
dortmund

Software-Engineering für langlebige Systeme

Thomas Ruhroth | Dortmund
SoSe 2015

PA

- HA1
- Übung zur Prüfung
- Code-Guidelines

HA - Bemerkungen

- Bearbeitungspunkt Aufgabe
  - Wer nur die Hälfte der Aufgabe bearbeitet kann nur die Hälfte der Punkte erwarten
- Heimübungslösungen sollen strukturiert sein

Verwechseln von Patterns

- Im Pattern liegende Gründe
  - eine ähnliche Struktur
  - ein ähnliches Verhalten
  - ein ähnliches Problem lösen
- Im Entwickler liegende Gründe
  - Kennt nicht alle Pattern
  - Pattern nicht gekennzeichnet
  - Falsche Nutzung

Cluster

- Abstract Factory
- Builder
- Factory Method
- Singelton

- Command
- Visitor
- Strategie

- Adapter
- Bridge
- Decorator
- Wrapper
- Proxy

Ein ernstes Wort: Programmierstil

- Bei einer Mehrheit der abgegebenen Taschenrechner ist der Programmierstil nicht in Ordnung!

- Copy und Past herrscht vor
- Kommentare, weil man Kommentare machen muss
- Unnötige Komplexität
- „Verzeigungsmethoden“
- Methoden haben mehr als eine Aufgabe
- Lange Methoden
- Kaum Struktur
  - Keine Trennung der Aufgaben
    - View-Document?

Analyse

- Copy und Paste-Programmierung
  - Duplizierter Code
  - Gleiche Anweisungen mit verschiedenen Parametern
- Lange Methode
  - Methode ist sehr lang (hier mehr als 25 Zeilen)
- Große Klasse
  - Zuviele Klassenelemente
- Neid (engl. Feature Envy)
  - Ständiger Zugriff auf Elemente einer anderen Klasse
  - Deutet auf flasche Verteilung der Features hin
- Case-Anweisungen in objektori. Code
  - Auch if-Rutschen
  - Objektorientierter Code sollte möglichst ohne Switch/cAse auskommen
  - Verletzung der OO-Eigenschaften

- Faule Klasse
  - Eine Klasse leistet zu wenig, um ihre Existenz zu rechtfertigen.
- Unangebrachte Intimität
  - Zwei Klassen haben zu enge Verflechtungen miteinander.
- Datenklasse
  - Klassen mit Feldern und Zugriffsmethoden ohne Funktionalität. Oft zu finden in einem sogenannten anämischen Fachmodell.
- Ausgeschlagenes Erbe
  - Unterklassen brauchen die Methoden und Daten gar nicht, die sie von den Oberklassen erben

- Kommentare
  - Angemessenheit
  - Ort
  - Aussagekraft
- Name Wahl (Aussage, lang, kurz)
  - Gute Namen .-)
- Komplexe/Tiefe Verzweigungen
  - Starke Schachtelung von Scheiben, Verzweigungen, Fehlerbehandlungen

- Code-Elemente Sprache
- Kommentar-Sprache

Übung zur Mündl. Prüfungen

- Zweier Teams
- Person A:
  - Stellt Fragen
  - Achtet auf Korrektheit
  - Macht sich zur Korrektheit Notizen
  - Bei Unsicherheiten in der Antwort nachfragen!
- Person B:
  - Beantwortet die Fragen

- Keine Notizen auf den Fragezettel!

Rückmelderunde

- B stellt kurz dar, wie es ihm ergangen ist.
- A gibt Rückmeldungen, ob
  - Die Antworten richtig waren
  - Die Antworten gut verständlich und nachvollziehbar waren
  - Ob B Notizen erstellt hat
  - Wie er die Leistung von B benoten würde

Rollentausch

Alles nochmal mit anderen Fragen und getauschten Rollen

Runde 1

- Was ist Softwareerosion?
- Welche Themenfelder kann man bei der Softwareerosion betrachten (Übersicht)?
- Welche wahrnehmungspsychologischen Prinzipien werden durch Coding Guidelines angesprochen?
- << Bitten Sie ihren Partner eines der genannten Prinzipien genauer zu erklären>>
- Was sind Design Pattern?
- Wie soll Code aus Sicht der langlebigen Systeme dokumentiert werden? Warum?

Runde 2

- Was sind langlebige Systeme?
- Wie heißt das Hauptproblem bei langlebigen System? Bitte beschreiben Sie dieses genau.
- Geben sie bitte eine Übersicht über die für die Vorlesung relevanten Bereiche der Wahrnehmungspsychologie.
- Nennen Sie einige Design-Pattern.
- << Bitten Sie ihren Partner eines der genannten Pattern genauer zu erklären>>
- Was ist Atomsemiotik ?
- Was hat die Atomsemiotik mit langlebigen Systemen zu tun?

Code-Guidelines und Dokumentation

- Überlegen Sie sich was an Code-Regeln ihnen wichtig ist.
- Stellen Sie drei Regeln zusammen, die sich als die wichtigsten bezeichnen würden.

  Verteidigen Sie ihre Regeln