

Spaghetticode

Christina Bell

SE: OO Entwurf & Analyse



Gliederung

- Spaghetticode

1. Steckbrief
2. Symptome und Konsequenzen
3. Ursachen
4. Ausnahmen
5. Problemlösung

- Input Kludge

1. Problem
2. Hintergrund
3. Lösung

- Walking through a Mine Field

1. Steckbrief
2. Hintergrund
3. Lösung



Steckbrief

| | |
|--------------------|---|
| Name: | Spaghetticode |
| Tritt auf in: | Anwendungen |
| Beseitigung durch: | Software Refactoring, Codesäuberung |
| Hauptursache: | Ignoranz, Faulheit |
| Vermeidung durch: | Komplexitätsmanagement, Änderungsmanagement |
| Nachweis: | „Es ist einfacher den Code neu zuschreiben als ihn zu ändern.“; „Hast du realisiert, dass die Sprache mehr als eine Funktion unterstützt.“ |
| Verwandte: | Analyse Paralyse, Lava Fluss |



Symptome

- geringe oder keine Softwarestruktur
- geringe Anzahl an Objekten
- Objekte enthalten eine lange Methode
- Methoden sind prozessorientiert
- Methoden besitzen keine Parameter
- wenige Beziehungen zwischen Objekten
- geringer Grad an dynamischen Interaktionen
- Vererbung und Polymorphismus nicht genutzt



Konsequenzen

- System ist schwer zu pflegen
- Code ist meist nicht wiederverwendbar



Ursachen

- Unerfahrenheit mit OO-Design
- keine Designpriorität für die Programmierung
- ineffektive Codeüberprüfung und Beratung
- häufig Resultat von allein arbeitenden Entwicklern
- Leistungsoptimierung



Ausnahmen

- Interface ist kohärent und nur der Code hat Spaghettiform
- Lebenszeit der Komponente ist kurz und isoliert
- Erfolg der Software ist abhängig von unverzüglicher Fertigstellung
- Softwarearchitekten sind nicht mit der Domain vertraut



Problemlösung

- Möglichkeit der Prävention
- Refactoring und Codesäuberung



Refactoring & Codesäuberung

- Zugriff auf Membervariablen einer Klasse durch Zugriffsfunktionen
- Codestücke konvertieren in Funktionen, die man wiederverwenden kann
- Funktionsargumente umordnen, um bessere Konsistenz zu erhalten
- Teile, auf die nicht zugegriffen wird, löschen
- Umbenennung von Klassen, Funktionen und Datentypen entsprechend des Firmen-/Industriestandards



Prävention

1. Entwicklung eines Domainmodells
2. ausgehend vom Domainmodell das Designmodell entwickeln
3. Designmodell sollte Objekte soweit zerlegen, dass der Entwickler sie versteht
4. Implementierung basierend auf dem Designmodell

Fragen





Gliederung

- Spaghetticode

1. Steckbrief
2. Symptome und Konsequenzen
3. Ursachen
4. Ausnahmen
5. Problemlösung



Input Kludge

1. Problem
2. Hintergrund
3. Lösung

- Walking through a Mine Field

1. Steckbrief
2. Hintergrund
3. Lösung



Problem

- Software die einfache Verhaltenstests nicht besteht
- ad hoc Algorithmen erledigen die Programmeingaben

Hinweis:

„Endnutzer können das neue Programm mit Berührung der Tastatur zum Absturz bringen.“



Hintergrund

1. Tag: herausfinden welche Eingabekombinationen zum Absturz führen
2. Tag: abfangen von dem was man ersten Tag herausgefunden hat
3. Tag: überprüfen, ob jetzt alles klappt
- ...
- ...
- Ende der Woche:
Verinnerlichung aller Eingabesequenzen, die zum Absturz führen



Lösung

- Verwendung von Software zur lexikalischen und semantischen Analyse (z.B. lex und yacc)



Gliederung

- Spaghetticode

1. Steckbrief
2. Symptome und Konsequenzen
3. Ursachen
4. Ausnahmen
5. Problemlösung

- Input Kludge

1. Problem
2. Hintergrund
3. Lösung

- ➔ Walking through a Mine Field

1. Steckbrief
2. Hintergrund
3. Lösung



Steckbrief

| | |
|-------------------|--|
| Name: | Walking through a Mine Field |
| Problem: | Verwendung neuer Softwaretechnologien |
| Auch bekannt als: | Nothing Works, Do you believe in Magic? |
| Ursache: | viele Fehler in veröffentlichten Softwareprodukten |
| Anekdote: | „Es gibt kein richtiges System, nicht einmal unseres“ |



Hintergrund

- Gaskin Story
- Es ist einfacher zu denken, dass man reife Technologien auf mächtigen, robusten Plattformen benutzt, anstatt der Tatsache ins Auge zu schauen das überall Fehler vorhanden sind.



Lösung

- höhere Investitionen beim Softwaretest
- Testsoftware
- formale Verifikation
- Softwareinspektion



Verwendete Materialien

Buch:

AntiPattern: Refactoring Software, Architectures
and Project in Crisis

(Brown, Malvean, Hays, McCormick, Mowbray)

Internetseite zu Antipattern:

www.antipattern.com

Übersetzung:

Oxford Dictionary, 1992

www.dict.tu-chemnitz.de

www.dict.leo.org

Fragen

