

Design by Committee

Seminar Pattern & Antipattern

Robert Sauer
sauer@informatik.hu-berlin.de

15.02.2002

Design by Committee

1

Marschplan

1. Übersicht
2. Problem
 - Symptome/Konsequenzen/Ursachen
3. Lösung
4. Beispiele
5. Resümee

15.02.2002

Design by Committee

2

Übersicht

- ◆ Komplexer Software-Entwurf durch Gruppe von Entwicklern
 - Resultat extrem komplex und vielfältig
 - Scheinbar unmöglich, Design komplett in ein Produkt umzusetzen
 - Unmöglichkeit, komplettes Design zu testen
 - Kein „roter Faden“

Andere Namen

- ◆ Gold Plating
 - Vergolden
- ◆ Standards Disease
- ◆ Make Everybody Happy
- ◆ Political Party
 - Partei, Parteitag (?)

Symptome - Produkt

- ◆ Unnötig komplex, unlesbar und fehlerhaft
- ◆ „dicke Schwarte“
- ◆ Instabilität

Symptome - Prozess

- ◆ Unkonzentrierte Sitzungsatmosphäre
- ◆ Serieller Diskussionsfluss
- ◆ Kein Fortschritt zwischen Sitzungen ...
 - ... und selbst dann nur mühsam
- ◆ Keine Prioritäten
 - Ziele des Entwurfs
 - Initialer Umfang

Konsequenzen

- ◆ Kein Konsens zwischen Architekt und Entwickler
- ◆ Entwurfsprozess verursacht Zeit- und Budgetüberschreitung
- ◆ Spezialisten exklusiv für dieses Design

Ursachen

- ◆ Wer ist Chefarchitekt des Projektes?
- ◆ Ineffizienter Prozess
 - Speziell Besprechungen
 - Jahrmarkt
 - mehr als 5 Entscheider
- ◆ „Vergoldung“
- ◆ Welches sind Prioritäten?

Ausnahmen

- ◆ Kleine Komitees
 - 6 bis 10 Teilnehmer
- ◆ „Tiger Teams“
 - Projektspezifische Gruppe von Spezialisten

Lösung

- ◆ Kern: Sitzungsdurchführung
 - Vorurteil: „Sitzungen ebenso lang wie lahm!“
 - Produktivitätssteigerung: Faktor 100
 - Faktor 100.000 - na ja (?)
 - Zeitverlauf präsent machen
 - Wanduhr
 - Kurze Beiträge
 - Vertiefung bei Nachfrage

Lösung: Rollenmodell

- ◆ Projektverantwortlicher
- ◆ Vermittler
- ◆ Architekt
- ◆ Entwickler
- ◆ Tester
- ◆ Fachlicher Berater

Lösung: Durchführung – „Spitwads“

1. Frage stellen
2. Teilnehmer notieren IHRE Antwort
3. Antworten einsammeln
4. Antworten verteilen, vorlesen, notieren
5. Antworten klären
6. Konsolidieren
7. Diskutieren

Negativ-Beispiel

◆ CORBA C++ Sprachanbindung

- RFP 1992
- Mitte 1993
 - Vorschläge von HP/SUN, IONA vs. HyperDesk
- Herbst 1993:
 - Allianz IONA und SUN
 - Überlauf HP zu HyperDesk → halbherzig aus politischen Gründen

Negativ-Beispiel (cont.)

- Dezember 1993
 - ORB Task Force votiert für HyperDesk
- bis März 1994
 - IBM unterstützt SUN
 - HyperDesk in finanziellen Problemen
 - Vorschlag zurückgezogen

→ Bang!!

Negativ-Beispiel (cont.)

◆ Zweiter Anlauf:

- Mitte 1994
 - drei Personen arbeiten an Kompromiss beider Vorschläge
- September 1994
 - Neuer Vorschlag, zwei Schichten
- Dezember 1994
 - Vorschlag angenommen

→ Dennoch: C++ Anbindung als kompliziert angesehen

Positiv-Beispiel

◆ IBM System/360

- ca. zehn Beitragende
- exakt zwei Autoren
 - herausragende Konsistenz
 - erfolgreiches System

Varianten

- ◆ „Railroad“ (aka „Rubber Stamp“)
 - Koalition dominiert Prozess
 - Erzwingen einer technisch blödsinnigen/fehlerhaften Lösung
 - Fehlerhafte Referenzimplementierung
 - Alle folgen ...

Verwandte Muster und Anti-Muster

- ◆ Keine Angabe!
- ◆ Denkbar:
 - „Corncob“ als Beteiligter (?)
 - „Reinvent the Wheel“ steigert Komplexität (?)



Literatur



1. Anti-Pattern Buch
2. Douglas C. Schmidt, Steve Vinoski: „Object Interconnections“, CUJ November 2000
3. Frederick P. Brooks: „The Mythical Man-Month“