

Software Engineering I
Prof. Dr. Martin Glinz

Fallstudie “CONFIRM”



Universität Zürich
Institut für Informatik

Wie verliert man mit Software 285 Millionen Dollar?

- Durch **Nichtbeherrschen** des Software-**Entwicklungsprozesses**.
- ...unter anderem
 - Wahl des **falschen Prozesses**
 - **Fehler** in der **Anwendung** des gewählten Prozesses
- Durch **Fehler** im **Software-Projektmanagement**
- ...unter anderem
 - Fahrlässige **Terminführung**
 - Nichterkennung kritischer **Risiken**

Der Fall – 1

- **Projekt:** CONFIRM, ein integriertes Reise- und Buchungssystem
- **Lieferant:** AMRIS, ein US-Softwarehaus (Tochter von American Airlines)
- **Pilotkunden:** Konsortium aus Marriott, Hilton und Budget Rent-a-Car
- **Beginn:** Oktober 1987
- **Vertragsabschluss:** September 1988
- **Geplanter Liefertermin:** Juni 1992
- **Einstellung des Projekts:** Juli 1992
- **Gründe:**
 - Massive technische Probleme
 - Massive Verfehlung der Leistungsanforderungen
 - Zeitverzug von über einem Jahr

Der Fall – 2

- **Erstes Budget:** Gesamtkosten von 55.7 Millionen \$
- **Aufgelaufene Kosten bei Projektabbruch:** 125 Millionen \$
- **Anschliessend:** Die Partner verklagen sich gegenseitig
- **AMRIS entschädigt Marriott, Hilton und Budget** in einem außergerichtlichen Vergleich mit Zahlungen in unbekannter Höhe, wahrscheinlich 160 Millionen \$

Quelle: Oz, E. (1994). When Professional Standards are Lax. The CONFIRM Failure and its Lessons. *Communications of the ACM* 37, 10 (Oct 1994). 29-36.

Stationen einer Software-Katastrophe – 1

Wann	Was	Kosten [M US\$]	Verzug	End- termin
Mar 87	Projektidee			
Okt 87	Beginn Anforderungsspezifikation			
Mai 88	Vertragsabschluss	55.7		Jun 92
Sep 88	“Base Design” vom Kunden nicht akzeptiert			
Mar 89	Funktionale und Technische Spezifikation vom Kunden nicht akzeptiert			
Sept 89	Nachbesserung abgeschlossen Beginn Entwicklung Teilsysteme	72.6	keiner	Jul 92
Feb 90	BAA (Business Area Analysis)- Meilenstein verfehlt		≥13 Wo	Jul 92
Mai 90	“Project is on time”			Jul 92
Okt 90			~1 Jahr	Jul 92

Software Engineering I

Kapitel <nr>

© 2003 by Martin Glinz

5

Stationen einer Software-Katastrophe – 2

Wann	Was	Kosten [M US\$]	Verzug	End- termin
Feb 91	Planrevision	92		Jul 92/ Mar 93
Sommer 91	Kündigungswelle beim Lieferanten; Projektelevaluation durch Berater, kritischer Bericht wird schubladiert, Berater entlassen			
Anfang Apr 92	System geht in Versuchsbetrieb ("Beta-Test") bei erstem Kunden		2-6 Monate	offen
Ende Apr 92	Lieferant feuert 8 Spitzenmanager und 15 weitere Leute, gibt Ver- heimlichung massiver Probleme zu		15-24 Monate	offen
Jul 92	Projekt wird eingestellt • unlösbare technische Probleme • nicht wirtschaftlich betreibbar	125		† R.I.P.

Software Engineering I

Kapitel <nr>

© 2003 by Martin Glinz

6

CONFIRM – Ein Fall für Software Engineering – 1

Beispiel 1: Projektführung mit Meilensteinen

- Meilenstein: Business Area Analysis (BAA) abgeschlossen
- geplant: Februar 1990
- erreicht: August 1990
- Aber: AMRIS (der Lieferant), weigert sich, Marriott (einem der drei Kunden) Einsicht in die Ergebnisdokumente der Phase BAA zu geben

Was schließen Sie daraus?

CONFIRM – Ein Fall für Software Engineering – 2

Beispiel 2: Terminführung

- Mai 1990:
 - Meilenstein Spezifikation um 6 Monate verfehlt
 - Meilenstein BAA noch nicht erreicht; Verzug bisher 3 Monate
 - Der Lieferant versichert: "Project is on time"

Was ist hier faul?

CONFIRM – Ein Fall für Software Engineering – 3

Beispiel 3: Risikoführung

- Zentrale Projektrisiken nicht erkannt und nicht gelenkt
 - Kosten pro Transaktion
 - Integration heterogener Teilsysteme
 - Kosten bei Scheitern des Vorhabens

Was hätte der Lieferant tun können?