

Anforderungsanalyse

Softwaretechnik

Andreas Zeller

Lehrstuhl für Softwaretechnik
Universität des Saarlandes, Saarbrücken

2005-10-24

Flughafen Denver



Grundsteinlegung 1989 · 53 Quadratmeilen groß
Geplant: 1,7 Milliarden US\$ Kosten, Eröffnung 1993

Auftrag an BAE

Ursprüngliche Annahme: Jede Fluggesellschaft baut ihr eigenes
Gepäcktransportsystem

United (70% Anteil am Flugverkehr in Denver) begann als einzige mit
Planung; Auftrag von United an BAE 1991

Denver merkte im Anschluss, dass keine andere Fluggesellschaft ein System plante ⇒ Ausweitung des BAE-Auftrags auf gesamtes Flughafengelände

Geplante Kosten 193 Mio US\$

Herkömmliches System ---

- Barcode-Label auf Gepäckstücken
- Gepäckstücke werden mit Bändern transportiert
- An Weichen sitzen Personen, die die Gepäckstücke in die richtige Richtung leiten
- Grundlage: Barcode-Scanner und Airport-Codes

Hauptziel ist *Geschwindigkeit* — Fluggäste sollen schnell umsteigen können

Das System in Denver ---

- 3550 Wagen transportieren Gepäckstücke
- Kanonen „schießen“ Gepäckstücke in (abgebremste) Wagen
- Sortier- und Führungscomputer kommunizieren per Funk mit Wagen und stellen 92 Weichen
- Führungscomputer verfolgen Flugsteig-Zuordnungen, Ausfälle, Inspektionen

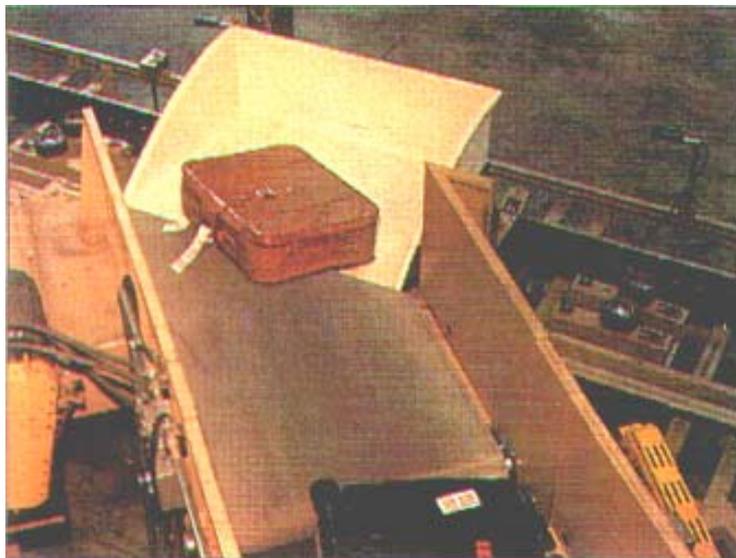
Ziel: Geschwindigkeit, Ausfallsicherheit, Kostensenkung

Last: 1000 Gepäckstücke pro Minute

Gepäcktransport am Flughafen Denver



Verladen eines Gepäckstücks



Entladen eines Gepäckstücks



Planungsprobleme

- Gepäcktransportsystem nachträglich in Planung einbezogen
- Nur zwei Jahre Zeit für Entwicklung
- Betreiber war nicht der Entwickler, sondern Drittfirma
- Nur wenige Wochen Testphase (Vergleich: Vor der Eröffnung des Flughafens München wurde das System 2 Jahre getestet)
- Andere Fluglinien wurden nicht einbezogen

Einsatzprobleme

- Kurven in Tunnels zu eng; Wagen blieben stecken
- Fahrtwind blies leichtes Gepäck von Wagen
- Nur 5% der Labels konnten korrekt gelesen werden
- Netzwerk war schon im Normalbetrieb zu 95% ausgelastet
- Alle Komponenten wurden an ihren Leistungsgrenzen betrieben ⇒ Kettenreaktionen
- Anschluss an Systeme anderer Fluglinien war problematisch

Komplexität

Beispiel: Leerer Wagen muss dorthin, wo er gebraucht wird

- Nachfrage nach leeren Wägen ändert sich ständig
- Wagen muss sich in Verkehr korrekt einordnen...
- ...und zum richtigen Zeitpunkt an seiner „Kanone“ sein
- Schienen zum Einordnen haben begrenzte Länge (Stau)
- Alles von einem zentralen System gesteuert

Folgen

Testlauf 1994 endete in Chaos.

- Fehler im ganzen System zerstörten Taschen und warfen Koffer aus den Wagen.
- Wagen entgleisten und fuhren ineinander
- Koffer zerbrachen und verteilten Inhalt in alle Richtungen.
- Schienen wurden bei Zusammenstößen verbogen.
- Wagen blieben einfach stehen oder kamen nicht, wenn sie gerufen wurden.
- Die meisten Wagen mit unlesbaren Barcodes wurden zu Haltestationen geleitet, die bald überfordert waren.
- Manche Wagen wurden von der eigenen Fracht blockiert oder von herumliegenden Kleidungsstücken, die von anderen aus zerrissenen Taschen verstreut worden waren, und wurden so zu Auslösern weiterer Crashes.
- Besonders an Kreuzungen gab es Massenkarambolagen.

Eröffnung muss von Oktober 1993 verlegt werden:

- erst auf Dezember 1993 (zum Debuggen),
- dann auf März 1994 (um Änderungen der Fluggesellschaften zu realisieren)
- dann auf Mai 1994
- im Mai 1994 schließlich auf „unbestimmte Zeit“, um zunächst einmal einen Plan zur Qualitätssicherung einzurichten.

Stand der Dinge

Der Flughafen wird im Februar 1995 eröffnet: 16 Monate Verspätung;
500 Millionen US\$ Schaden

Manuelles Ersatzsystem eingerichtet für weitere 71 Millionen US\$

Das vollautomatische System wurde vereinfacht, nur von United benutzt,
und auch nur an einem Terminal.

Die laufenden Kosten des Systems betragen 1 Mio US\$/Monat.

Im August 2005 wird es durch ein manuelles System ersetzt.

Die Firma BAE, die das System gebaut hat, ist pleite.

Analyse

Gründe für den Misserfolg waren:

- Fehlende Planung
- Mangelnde Technik, von der trotzdem erwartet wurde, dass sie perfekt funktioniert
- Fehlende Anpassung der Software an sich ändernde Anforderungen
- Die sich ständig ändernden Anforderungen
- Zu starrer und enger Zeitplan
- Zu langsames Netzwerk
- Die gewaltige Komplexität

Lektion 1: Machbarkeitsstudie

Zu Beginn jedes Projekts die *Durchführbarkeit* prüfen:

- Technische Machbarkeit
- Organisatorische Machbarkeit
- Wirtschaftliche Machbarkeit
- Verfügbarkeit von Einsatzmitteln
- Projektumfeld (Marktsituation, politisch, juristisch...)

Mögliche Grundlagen:

- Technisch-wissenschaftliche Analyse
- Expertenbefragung
- Simulationen (insbesondere *vertikaler Prototyp*)

Lektion 2: Anforderungen analysieren!

Fehler in der Anforderungsanalyse sind die Hauptquelle
für Projekt-Fehlschläge.

— Glass (1998)

Weitere Beispiele

Sozialhilfe-System in Florida. Entwickler sollten dezentrales System aufbauen, aber Millionen Codezeilen eines zentralen Systems wiederverwenden

FAA Flugkontrolle. Neue 20''-Displays waren langsamer als bisheriges Papierverfahren

Mautsystem an deutschen Autobahnen. On-Board-Units sollten neben Mautabrechnung zahlreiche Funktionen auf einmal erfüllen

Hintergrund

- Definition der Anforderungen ist schwer
- Unterschiedliche Interessen aller Beteiligten
- Prozess umfasst Lernen und Verhandlungen
- Jeder Beteiligte hat nur begrenztes Wissen
- Technik kann diese Probleme nicht lösen

Lektion 3: Planen!

Während der Anforderungsanalyse und Entwurf treten die meisten Fehler auf; je später sie korrigiert werden, desto teurer werden sie.

— Boehm (1975)

Hintergrund

- Menschen haben gewöhnlich Schwierigkeiten, komplexe Situationen als Gesamtheit zu erfassen
- Die meisten Fehler entstehen durch Weglassen (insbesondere von Ausnahmen und anderen außergewöhnlichen Situationen)
- Die Kosten einer Änderung sind um so höher, je mehr Entscheidungen bereits getroffen wurden – da diese auf den früheren Annahmen basierten
- Je später eine Änderung, desto mehr Menschen sind davon betroffen

Wie Anforderungen analysieren?

Ein gängiges Verfahren, systematisch Anforderungen zu sammeln, sind sogenannte *Use Cases*.¹

Ein Use Case beschreibt, wie ein *Akteur* („Handelnder“) zu seinem *Ziel* kommt.

Welche Akteure gibt es, und welche Ziele haben sie?

¹Nach Larman, *Applying UML and Patterns*, 2002

Definitionen

Ein *Akteur* ist etwas, das sich verhalten kann – etwa eine Person, ein System, oder eine Organisation.

Ein *Szenario* ist eine spezifische Folge von Aktionen und Interaktionen zwischen Akteuren (von denen einer ein Computersystem ist).

Ein *Use Case* ist eine Sammlung von verwandten Szenarien – typischerweise solchen, die erfolgreich sind, und solchen, die fehlschlagen.

Use Case: Warenrückgabe

Erfolgsszenario: Ein Kunde kommt zur Kasse mit Waren, die zurückgegeben werden sollen. Der Kassierer benutzt das Kassensystem, um jede Ware einzugeben...

Alternative Szenarien:

- Hat der Kunde mit Kreditkarte gezahlt, und kann der Betrag nicht der Kreditkarte gutgeschrieben werden, wird der Betrag bar ausgezahlt.
- Ist der Warencode nicht (mehr) im System verzeichnet, muss die Aufsicht gerufen werden, die den Code manuell eingibt.
- Kann das Kassensystem sich nicht mit der Zentrale verbinden...

Formaler Use Case

Akteur: Kassierer

Stakeholder und deren Interessen:

Kassierer will präzise, schnelle Eingabe, und keine Fehler beim Zahlen (fehlendes Geld wird vom Lohn abgezogen)

Kunde will Einkauf und Bezahlung mit minimalem Aufwand. Will Quittung für Reklamationen und Rückgaben.

Firma will korrekte Abrechnung, Buchführung, und korrekte Inventarisierung.

Staat will korrekte Abrechnung und Buchführung für Steuerzwecke.

Formaler Use Case – Bedingungen

Vorbedingungen: Kassierer ist angemeldet.

Nachbedingungen bei Erfolg:

- Verkauf ist gespeichert.
- Steuer ist korrekt berechnet.
- Inventar ist auf neuestem Stand.
- Quittung ist ausgehändigt.

Formaler Use Case – Erfolgsszenario _____

1. Kunde kommt zu Kasse mit zu bezahlenden Artikeln
2. Kassierer beginnt neuen Verkaufsvorgang
3. Kassierer gibt Bezeichner für Gut ein
4. Das System merkt sich die verkauften Artikel und zeigt Artikelbeschreibung, Preis, und Gesamtsumme.
Schritte 3-4 wiederholen, bis alle Artikel bearbeitet
5. Das System zeigt die Gesamtsumme
6. Der Kassierer nennt die Summe dem Kunden und bittet um Bezahlung.
7. Der Kunde zahlt; das System verarbeitet die Zahlung.
8. Das System verzeichnet die Zahlung und sendet die Information ans Zentralsystem weiter.
9. Das System druckt die Quittung.
10. Der Kunde geht mit Quittung und Waren (sofern vorhanden)

Formaler Use Case – Alternative Szenarien _____

*a. (jederzeit) Das System schlägt fehl.

1. Alle Transaktionen müssen jederzeit rekonstruierbar sein.
2. Der Kassierer startet das System neu und meldet sich an.
3. Das System rekonstruiert die bisherige Transaktion.
 - (a) Schlägt auch dies fehl, meldet das System dies, und geht in den Initialzustand über.
 - (b) Der Kassierer beginnt den Verkauf von vorne.

Formaler Use Case – Alternative Szenarien (2) _____

3a. (Artikeleingabe) Ungültiger Bezeichner.

1. Das System meldet einen Fehler und weist die Eingabe zurück.

3b. (Artikeleingabe) Es gibt mehrere Artikel gleicher Art, bei denen individuelle Kennzeichnungen nicht wichtig sind (z.B. fünf gemischte Tafeln Milka-Schokolade)

1. Der Kassierer gibt Kategorie und Artikelanzahl ein.

...

7a. (Zahlung) Zahlung mit Debitkarte...

7b. (Zahlung) Zahlung mit Kreditkarte...

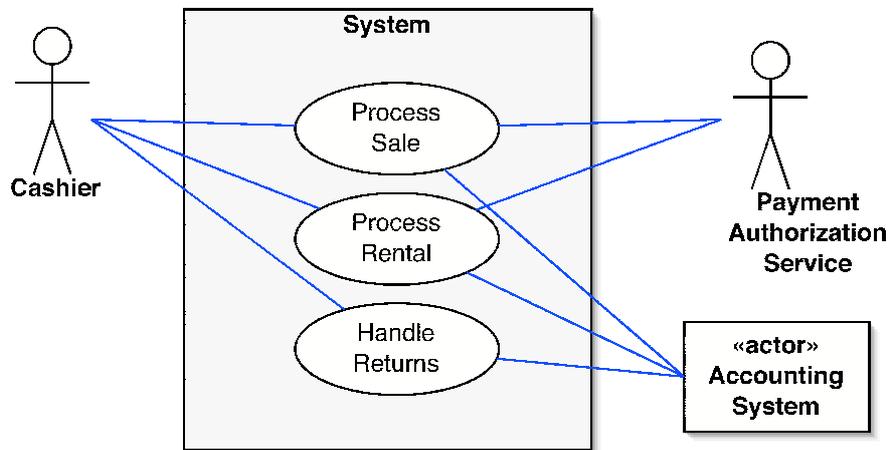
7c. (Zahlung) Zahlung mit Geldkarte...

Akteure und Ziele finden _____

Standard-Vorgehen:

1. Wo ist die Grenze des Systems? Ist es nur die Software, die Hardware und Software, oder zusätzlich die benutzende Person, oder eine ganze Organisation?
2. Wer sind die primären Akteure – die, deren Ziele das System erfüllt?
3. Was sind die Ziele dieser Akteure?
4. Beschreibe, wie das System diese Ziele erfüllt – einschließlich aller Ausnahmen.

Graphische Darstellung



Use-Case-Diagramme können Texte allenfalls ergänzen!

Zusammenfassung

- ✓ Fehler in der Anforderungsanalyse sind die Hauptquelle für Projekt-Fehlschläge.
- ✓ Während der Anforderungsanalyse und Entwurf treten die meisten Fehler auf; je später sie korrigiert werden, desto teurer werden sie.
- ✓ *Use Cases* sind ein erprobtes Mittel, um systematisch Anforderungen zu sammeln.
- ✓ Ein Use Case beschreibt, wie ein *Akteur* („Handelnder“) zu seinem *Ziel* kommt.
- ✓ Ein Use Case besteht aus einem *Erfolgszenario* und *alternativen Szenarien* (textuell gegliedert).