



Action-Centered Design

Proseminar

Mensch - Maschine Interaktion



Inhalt

- Einleitung
 - Peter J. Denning
 - Pamela A. Dargan
- Softwaredesign
 - Wichtigkeit des Softwaredesigns
 - Software Engineering
 - Human-Centered Design
- Action-Centered Design in der Praxis
 - Ontologie
 - Pattern Language
 - Business-Process Mapping
- Fazit



Einleitung

- *Bringing Design to Software*, Kapitel 6
⇒ Action-Centered Design
- Peter J. Denning
- Pamela A. Dargan

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Action-Centered Design

- Disziplin der Softwarearchitektur
- Unterkategorie des Human-Centered Designs
- Benutzer stehen im Mittelpunkt der Entwicklung
- Kommunikation zwischen Entwickler und Benutzer sehr wichtig

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Peter J. Denning

Auszug aus dem Lebenslauf:

- Vorsitzender des *Computer Science Departement* an der *Naval Postgraduate School* in Monterey, Kalifornien
- Zuvor Professor an der *George Mason University* in Fairfax, Virginia
- Davor Präsident der *Association for Computing Machinery (ACM)*

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Peter J. Denning

Forschungsinteressen:

- Innovative Prinzipien
- Betriebssysteme
- Zukunft des IT-Business
- Performance Modeling
- Arbeitsablaufoptimierung
- Hochleistungsrechnen
- Sicherheitsproblematik

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Pamela A. Dargan



- Erfahrene Softwareentwicklerin
- Aktuell bei einer Nonprofit-Gesellschaft in Washington, D.C.
- Entwicklung von Architekturen für offene System

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Softwaredesign

- Softwaredesign legt Struktur und Form des Programms fest
- Wichtiges Kriterium für Erfolg der Software
- Es gibt verschiedene Ansätze zum Softwaredesign
 - ⇒ Software Engineering
 - ⇒ Human-Centered Design

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Wichtigkeit

- Benutzer entscheiden über den Erfolg einer Software
- Benutzer wollen effizient arbeiten
- Gutes Design
 - ⇒ Leichte Bedienbarkeit
 - ⇒ Kundenzufriedenheit
 - ⇒ Volle Nutzung des Funktionsumfangs
 - ⇒ Höhere Motivation

Einleitung

Software design

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



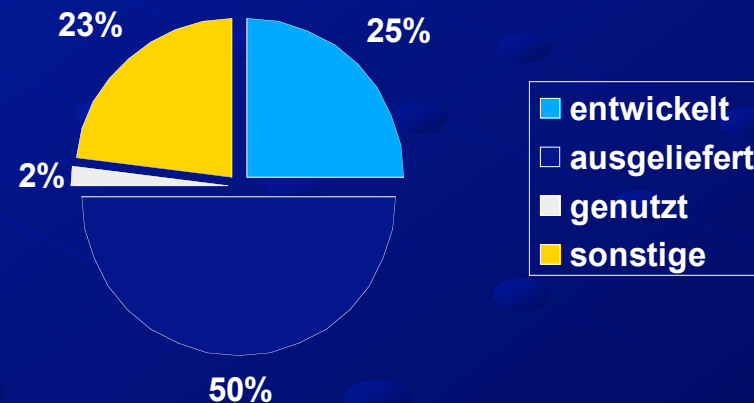
Wichtigkeit

Beispiel:

Statistik des US Government Accounting Office von 1979

Von Mitteln für 9 Software-Projekte wurden:

- 25 % ausgegeben für Software, die entwickelt und nie ausgeliefert wurde
- 50 % ausgegeben für Software, die ausgeliefert und nie genutzt wurde
- 2 % ausgegeben für Software, die ausgeliefert und genutzt wurde



Einleitung

Software design

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Software Engineering

- Entstanden in den 1960er Jahren
- Strukturierter Ansatz zur Softwareentwicklung
- Hervorgegangen aus den Ingenieurwissenschaften
- Spezifikationen (Leistungsbeschreibung) stehen vor Entwicklungsbeginn fest
 - ⇒ Geringer Kommunikationsbedarf zwischen Entwickler und Kunde
- ⇒ Software-lifecycle model

Einleitung

Software design

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

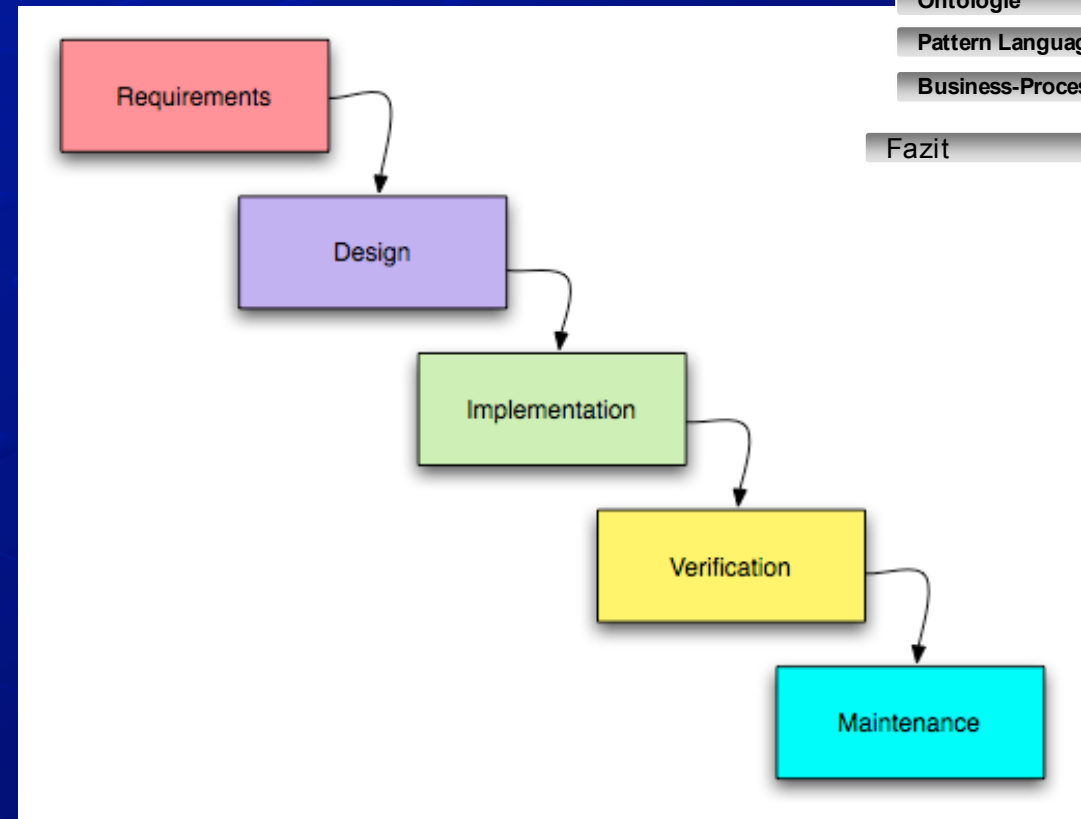
Fazit



Software-lifecycle Model

waterfall model:

- Vorgegebene Reihenfolge
- Schritte bauen aufeinander auf
- Rückschritte nicht vorgesehen
- Mittlerweile überholt



Einleitung

Software design

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Software-lifecycle Model

Einleitung

Software-design

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

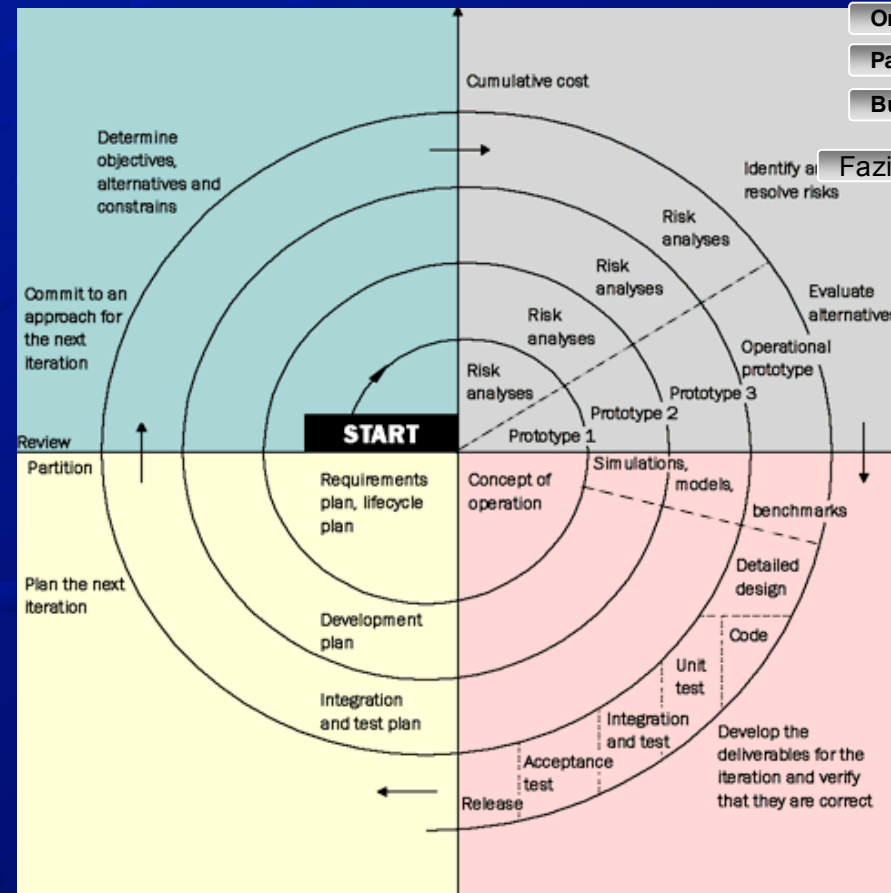
Pattern Language

Business-Process

Fazit

spiral model:

- Weiterentwicklung des waterfall model
 - Mehrere Iterationen vorgesehen
 - Zufriedenstellende Erfüllung der Spezifikation
- Aber:
- Kommunikationsmangel





Software Engineering

Im Allgemeinen gibt es 3
(ingenieurwissenschaftliche)

Annahmen zum Software Engineering:

- Das Ergebnis des Designprozesses ist ein Produkt
- Das Produkt entsteht aus den vom Kunden vorgegebenen Spezifikationen
(Entstehungsprozess theoretisch automatisierbar)
- Geringer Kommunikationsbedarf zwischen Kunde und Entwickler, sobald Spezifikationen festgelegt

Einleitung

Software design

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Human-Centered Design

- Aktuellerer Ansatz aus den 1980er Jahren
- Benutzer steht im Mittelpunkt des Entwicklungsprozesses
 - ⇒ Software wird an den Nutzer angepasst
- Ständige Kommunikation zwischen Entwickler und Kunde
- Sinnvolle Aufteilung der Aufgaben zwischen Mensch und Maschine

Einleitung

Software design

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit

Human-Centered Design

Einleitung

Software design

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

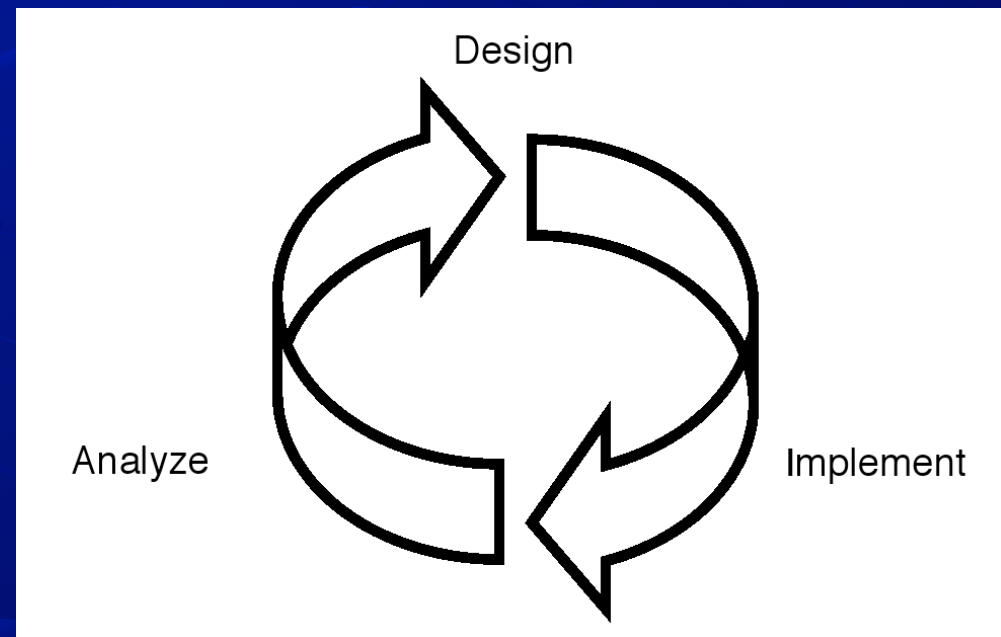
Pattern Language

Business-Process

Fazit

Realisierung durch DIA-Cycle:

- Zyklus wird durchlaufen bis Kunde zufrieden ist
- Benutzerfeedback geht direkt in nächste Iteration ein
- Produktdesign wird mit jedem Durchlauf konkreter





Human-Centered Design

Einleitung

Software design

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit

Auch hier existieren 3 Annahmen:

- Das Ergebnis eines guten Designs ist ein zufriedener Kunde
- Designprozess ist Zusammenarbeit zwischen Designer und Kunden. Design muss sich an veränderte Ansprüche des Kunden anpassen
- Designer und Kunde stehen während des Entwicklungsprozesses in permanentem Kontakt



Action-Centered Design in der Praxis

Anhaltspunkte für erfolgreiches Design:

- Richtigen Arbeitsbereich wählen
- Arbeitsschritte der Nutzer analysieren
- Funktionen entwickeln, die sich Nutzer schon immer gewünscht haben
- Frühzeitiges Verteilen von Prototypen
- Ziel ist es, die Benutzer zufrieden zu stellen
- Kundenzufriedenheit ist nicht dauerhaft

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Action-Centered Design in der Praxis

Beispiel: Finanzsoftware *Quicken*

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

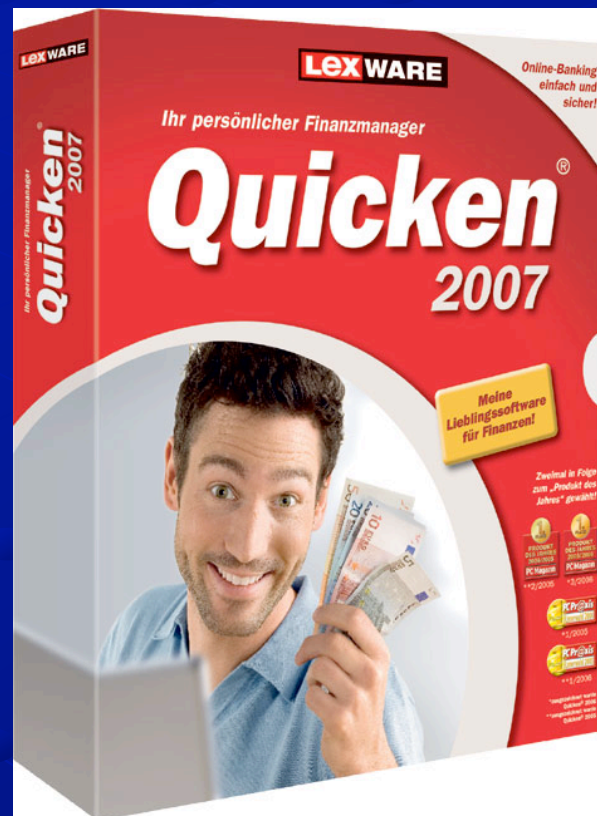
Praxis

Ontologie

Pattern Language

Business-Process

Fazit



- Erste Version 1986
- Heute immer noch erfolgreich
- Mitte Mai dieses Jahres Quicken 2007
- Gute Umsetzung der eben genannten Punkte





Action-Centered Design in der Praxis

Beispiel: Finanzsoftware Quicken

- Einleitung
- Softwaredesign
- Wichtigkeit
- SW Engineering
- Human-Centered
- Praxis
- Ontologie
- Pattern Language
- Business-Process

Fazit

The screenshot shows the Quicken 2007 interface. The main window is titled 'Zahlungsverkehrs-Zentrale'. On the left is a navigation menu with categories like 'Meine Finanzen', 'Meine Bankgeschäfte', 'Mein Business', 'Meine Wertpapiere', 'Mein Vermögen', 'Meine Steuern', and 'Meine Auswertungen'. The main area displays a table of transactions with columns for 'Datum', 'Vorgangart', 'Empfänger/Zahlungspflichtiger', 'BLZ/BIC', 'Bankname', 'Kontonummer/BAI1', 'VWZ1', 'Betrag', and 'Übert...'. A modal window titled 'Überweisungen: Volksbank Freiburg (€)' is open, showing a form to create a transfer. The form includes fields for 'Konto auswählen' (Volksbank Freiburg), 'Formular ausfüllen' (QuickPay, Adressen), 'Überweisung/Zahlschein' (Bankleitzahl, Kontoinhaber, Begünstigter/Adressenkennung, Kontonummer des Begünstigten, BLZ suchen, Kreditinstitut des Begünstigten, Verwendungszweck, Betrag: Euro, Cent), 'Kategorie' (60 Sonst. Betriebsausg.), 'Ausführungsart' (Online an die Bank übertragen, Auf ein Formular drucken, Sammelüberweisung, Per DTA-Diskette, Terminierte Überweisung), and 'Ausführen am' (21.04.05). Buttons for 'Abbrechen', 'Hilfe', 'Sofort ausführen', and 'Speichern' are at the bottom.



Action-Centered Design in der Praxis

Vorgehensweise beim Action-Centered Design:

Erstellen einer Ontologie



Darstellung als Pattern Language



Business-Process Mapping

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Action-Centered Design in der Praxis

Eine Ontologie ist ein System von Konzepten und Relationen zur Wissenspräsentation.

Sie besteht aus:

Der Menge der im Arbeitsumfeld

- Gängigen Fachausdrücke
- Üblichen Vorgänge
- Gebräuchlichen Werkzeugen
- Möglichen Problemsituationen
- Dauerhaften Zielsetzungen

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Action-Centered Design in der Praxis

Eine Pattern Language ist eine formale Sprache zur Beschreibung von Ontologien.

Einige Beispiele hierfür sind:

- RDF-Schema
- DAML+OIL
- F-Logic
- Web Ontology Language (OWL)
- Web Service Modeling Language (WSML)

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Action-Centered Design in der Praxis

Business-Process Mapping

- Aufbauend auf Pattern Language
 - Darstellung der Zusammenhänge und Prozesse in einem Arbeitsumfeld
 - Wiedergabe der zeitlichen und logischen Abfolge
 - Informationsflüsse und sich wiederholende Abläufe sind erkennbar
- ⇒ Grundlage für Designentwurf

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Action-Centered Design in der Praxis

Business-Process Mapping

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

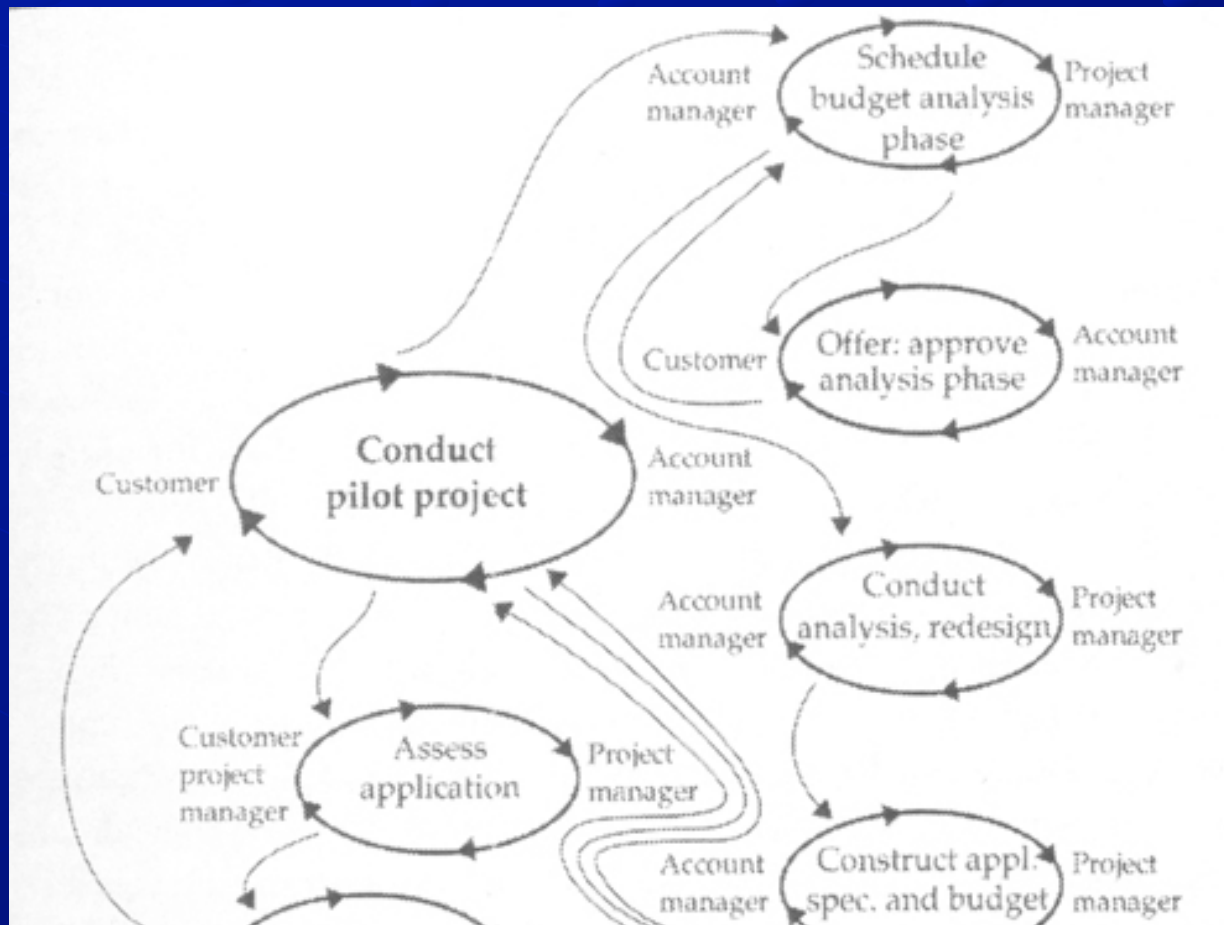
Praxis

Ontologie

Pattern Language

Business-Process

Fazit





Fazit

- Primäres Ziel des Softwaredesign: Kundenzufriedenheit
- Ansätze des Software Engineering nicht ausreichend
 - ⇒ Human-Centered Design
- Benutzer und seine Arbeitsumgebung stehen im Mittelpunkt der Entwicklung
- Ständige Kommunikation zwischen Entwickler und Kunden
- Produkt wird in mehreren Iterationen des DIA-Cycle verfeinert, bis der Kunde zufrieden ist

Einleitung

Softwaredesign

Wichtigkeit

SW Engineering

Human-Centered

Praxis

Ontologie

Pattern Language

Business-Process

Fazit



Vielen Dank für die Aufmerksamkeit