

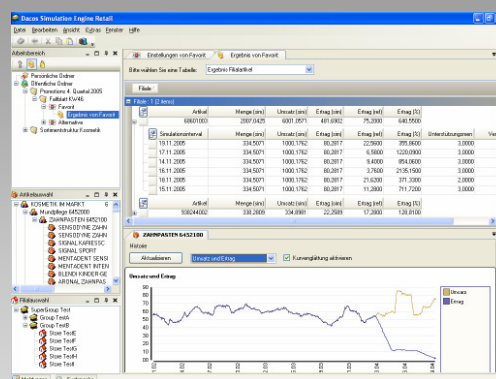
# Organisation einer modernen mittelständischen Software-Entwicklung

am Beispiel der Dacos Software GmbH, Saarbrücken



## Dacos & die (Retail) Simulation Engine

- ▶ KI-Modelle (Kunden-Agenten) werden aus homogenen Verhaltensdaten (Kassenboni-Cubes) gelernt.
- ▶ Die (Abverkaufs-)Reaktion der Agenten auf Umgebungsreize (Preise, Bewerbungen, Wetter, Konkurrenzpreise) wird simuliert.
- ▶ Einbettung der Simulation in konkrete Entscheidungshilfen
  - ▶ Optimierung von Sortimentspreisen
  - ▶ Ein- und Auslistungsentscheidungen
  - ▶ Bewerbungsplanung
  - ▶ Kunden-Segmentierung und -prognose
  - ▶ Vergangenheitsanalyse



Software Management

**DACOS**  
SOFTWARE

## Software- & Organisations-Struktur muss dem Geschäftsziel angepasst sein

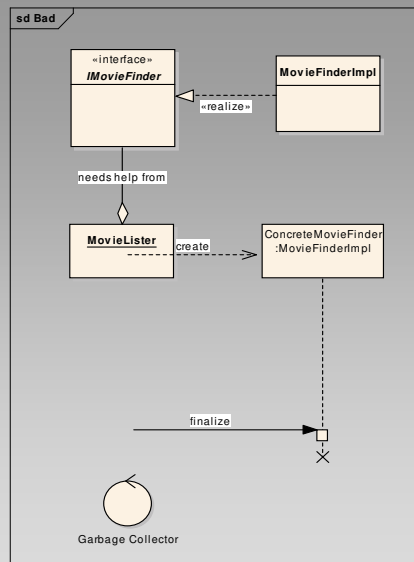
- ▶ Vor einem Jahr
  - ▶ Technische Unterstützung für individuelle Forschungsprototypen
  - ▶ Analysedienstleistungen
  - ▶ 4 Festangestellte
- ▶ Heute
  - ▶ Anpassbare Standard-Software für deutschen Markt
  - ▶ Simulation als Ergänzung zu bestehenden operativen Systemen
  - ▶ Kein Werkzeug, sondern Anwendung
  - ▶ 25 Mitarbeiter
- ▶ Vision
  - ▶ Überschaubares, Hoch-Qualifiziertes Team
  - ▶ OEM- und Vertriebs-Partnerschaften
  - ▶ Internationale Verbreitung.
- ▶ Organisations-Struktur ist oftmals schneller änderbar als Software-Struktur!

Software Management

**DACOS**  
SOFTWARE

## OO falsch verstanden

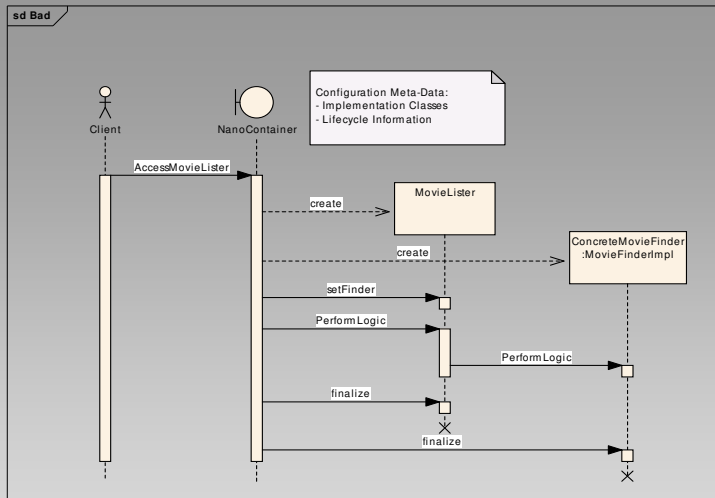
- ▶ Implementierungs-Beziehungen „hart verdrahtet“.
- ▶ Lebenszyklen unvorhersehbar.



Software Management

**DACOS**  
SOFTWARE

## Mikro-Architektur: Nano-Container

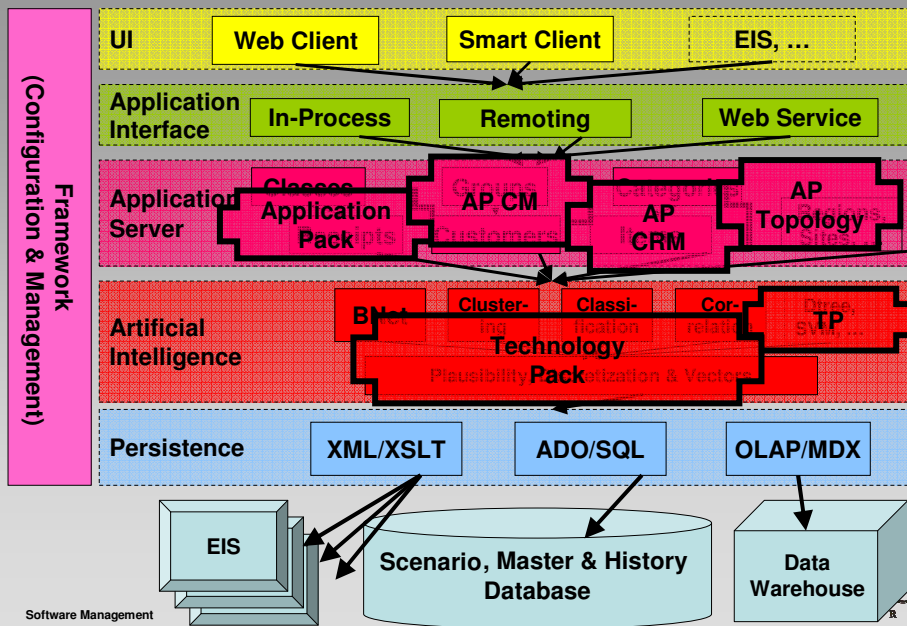


- ▶ Martin Fowler, *Inversion of Control Containers and the Dependency Injection Pattern*, Jan 2004
- ▶ Spring Framework, <http://www.springframework.net>

Software Management

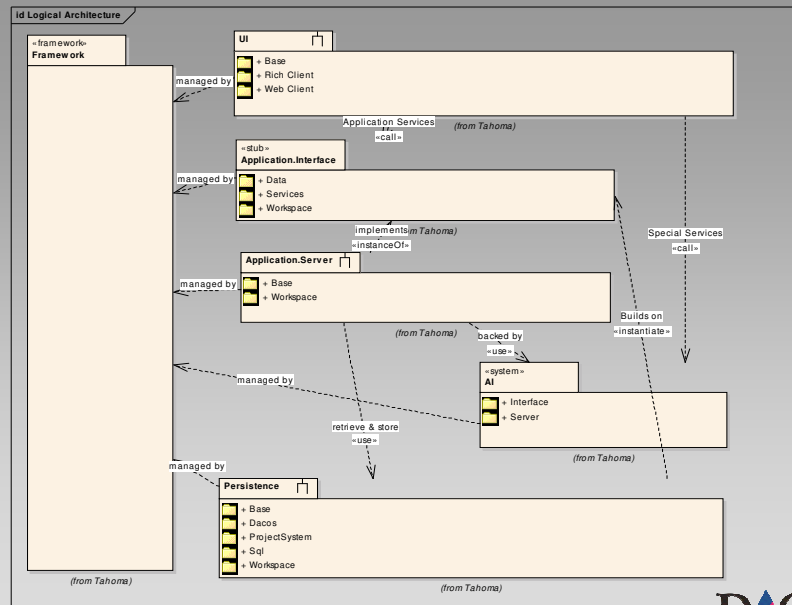
DACOS  
SOFTWARE

## Makro-Architektur: Services, Komponenten, Schichten



Software Management

## Alles Echt!



Software Management

**DACOS**  
SOFTWARE

## Service vs. Schnittstelle?

- ▶ Service ist Spezialfall einer Schnittstelle
- ▶ Trennung zwischen (Daten-)Objekten von (Manipulations-)Diensten
- ▶ Für Remote-Zugriff geeignet
- ▶ Nachrichten-Orientierte Methoden = Individuelle End-Punkte

```
public interface IService {
    IResponse ServiceCall(IRequest theRequestMessage);
}

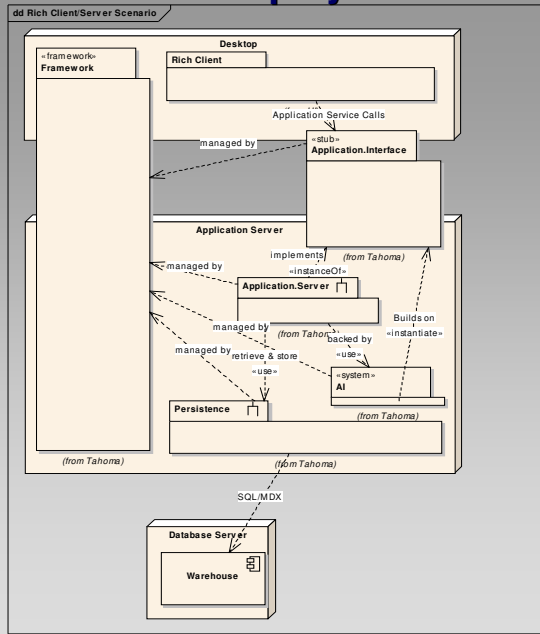
public interface IRequest {
    ISerializable Input {
        get; set;
    }
}

public interface IResponse {
    ISerializable Output {
        get; set;
    }
    IEvent[] Hints {
        get;
    }
}
```

Software Management

**DACOS**  
SOFTWARE

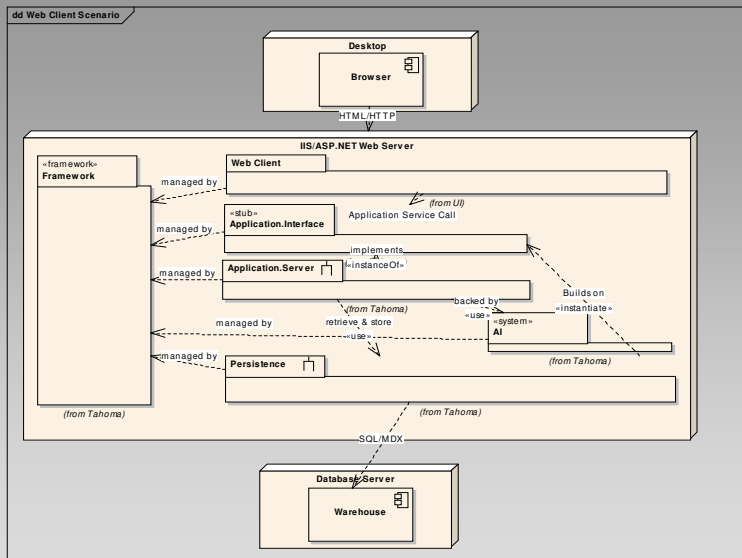
## Rich-Client 3 Knoten Deployment



Software Management

**DACOS**  
SOFTWARE

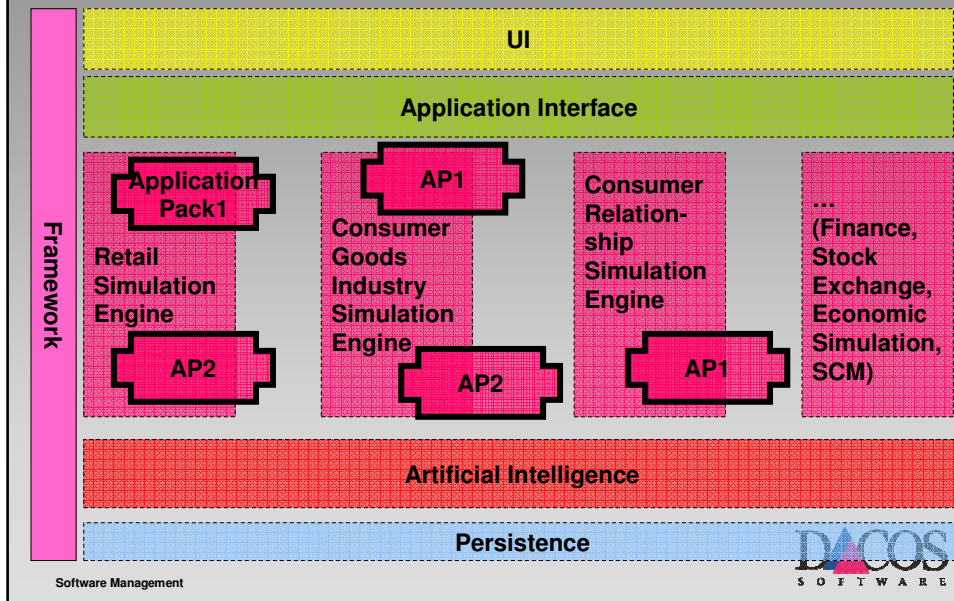
## 2-Knoten Web-Client



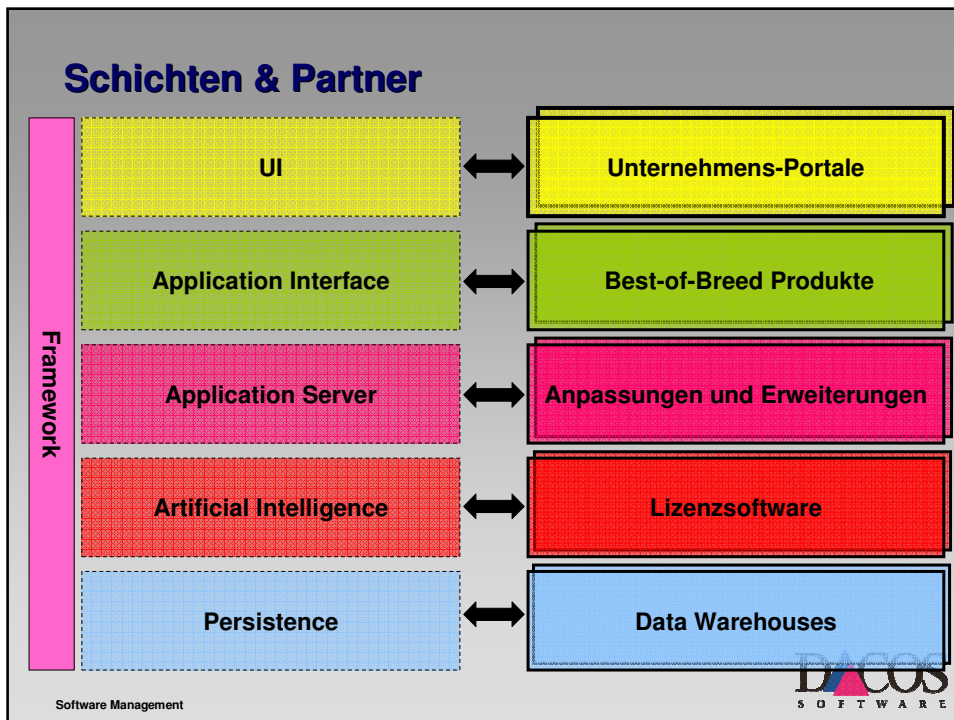
Software Management

**DACOS**  
SOFTWARE

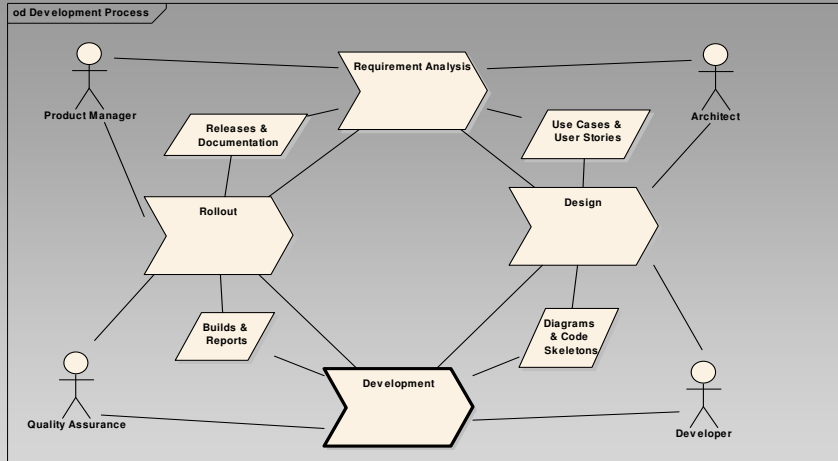
## Branchenpakete als Logik-Komponenten



## Schichten & Partner



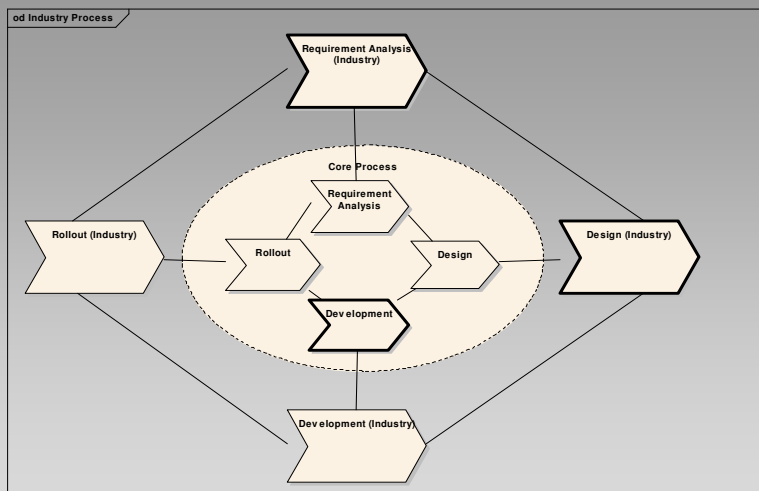
# Der Dacos Entwicklungsprozess



Software Management



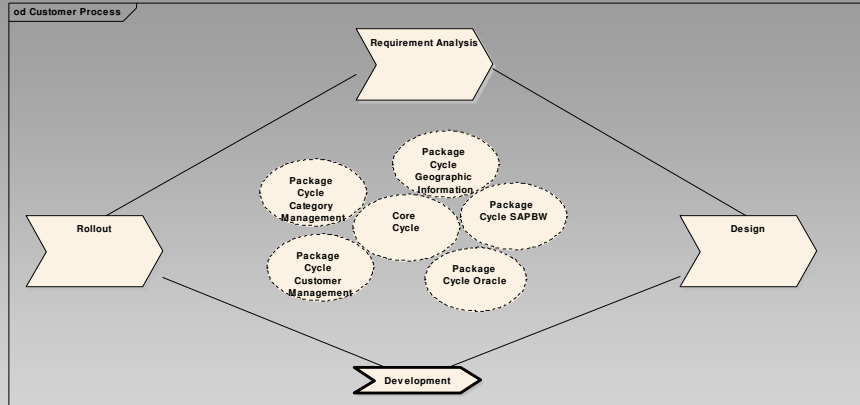
# Branchen- & Kernprozesse



Software Management



## Ziel: Standardisierte Individualentwicklung



Software Management

**DACOS**  
SOFTWARE

## Personen und Rollen

- ▶ 3 Produkt-Manager
- ▶ 2 Forscher
- ▶ 3 System-Architekten
- ▶ 2 Datenbank/Warehouse-Spezialisten
- ▶ 5 Modul-Architekten
- ▶ 1 Qualitätsmanager
- ▶ 12 Entwickler
  
- ▶ pro Projekt eine Rolle (einen „Hut“)
- ▶ 1 Entwicklungsleiter

Software Management

**DACOS**  
SOFTWARE

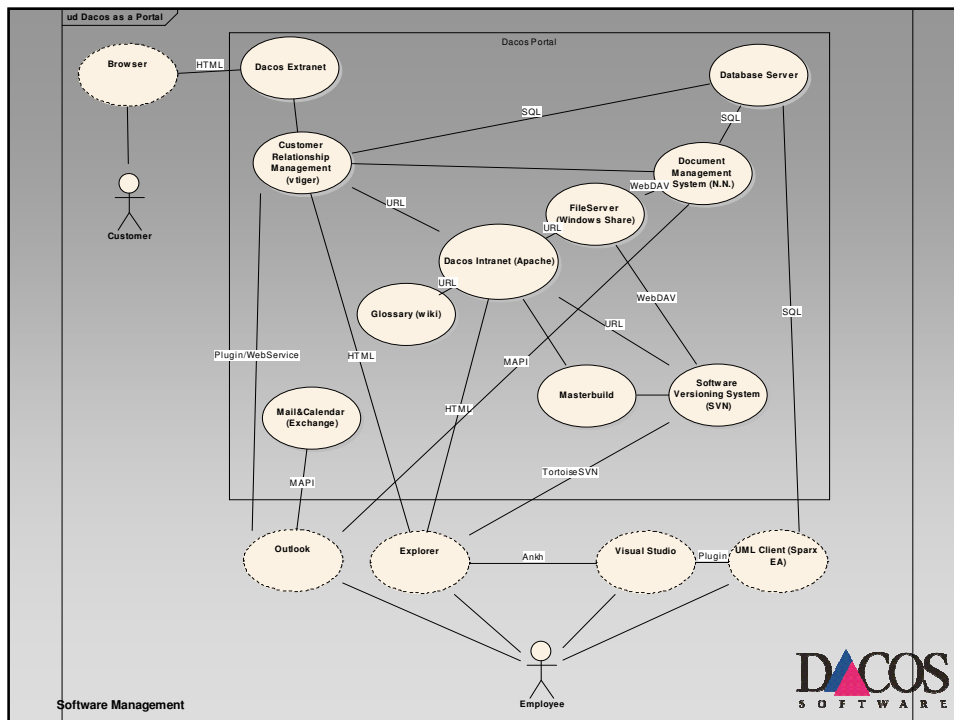


## Grundlage: Das Software-Entwicklungs-Portal

- ▶ Eine Sammlung von geeigneten Werkzeugen, deren Zusammenschluss den gesamten Entwicklungsprozess überhaupt erst möglich macht.
  - ▶ Versionierung und sichere Speicherung aller Artefakte
  - ▶ Einfacher und effizienter Zugriff
  - ▶ Unterstützt Phasenübergänge
  - ▶ Selbstbeschreibend
  - ▶ Standardisierung der Arbeitsumgebung
  - ▶ Mit Kunden & Vertriebsanbindung.

Software Management

**DACOS**  
SOFTWARE



Software Management

**DACOS**  
SOFTWARE

## Issue-Tracking mit CRM (http://www.vtiger.com)

**Kampagne=Projekt oder Release**

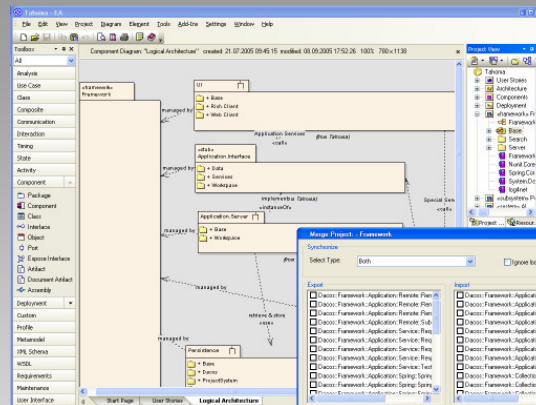
**Ticket=Teilprojekt oder Bug**

**Aktivität=Entwicklungsschritt oder Behebung**

**Software Management**

**DADOS SOFTWARE**

## UML als Spezifikations- und Modellierungswerkzeug



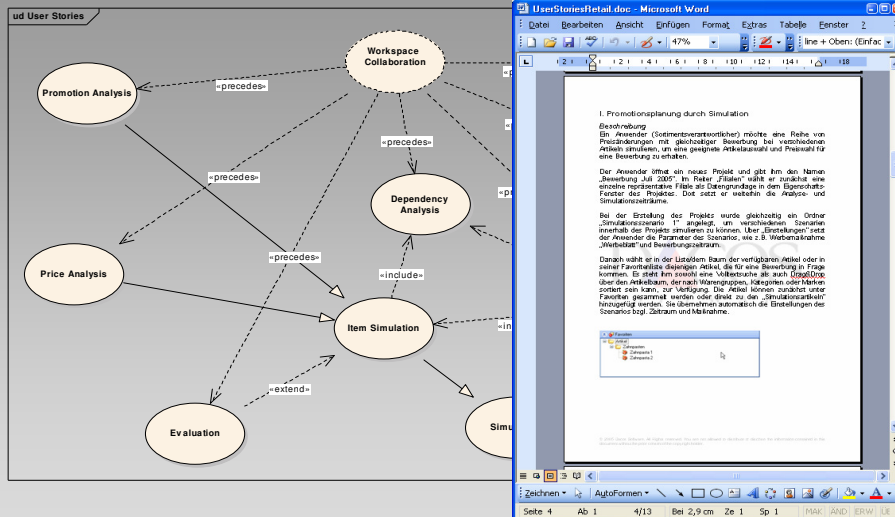
Sparx Systems Enterprise Architect 6.0 gekoppelt an Visual Studio

- ▶ „Single-Source“
  - ▶ Modell-Änderungen erzeugen Skeletons oder Code-Änderungen
  - ▶ Code-Änderungen wirken auf das Modell

Software Management

**DADOS SOFTWARE**

## Grob-Spezifikation=Use Cases + User Stories



Software Management

**DACOS**  
SOFTWARE

## Feinspezifikation durch Unit-Tests

- ▶ Gegen die aus UML generierten Schnittstellen-Skeletons werden Tests geschrieben
  - ▶ Whitebox, Low-Level
  - ▶ Blackbox, High-Level
    - ▶ realisieren beispielhafte Durchläufe der Aktivitäts-/Sequenzdiagramme
- ▶ Nano-Container + NUnit = Test-Umgebung
  - ▶ Spezielle Mock-Implementierungen ersetzen aufwändige Deployments
  - ▶ Geht bis zur Modell-Schicht am Client (gefüttert mit Test-Ereignissen)

Software Management

**DACOS**  
SOFTWARE

## Konfigurationsmanagement besteht aus

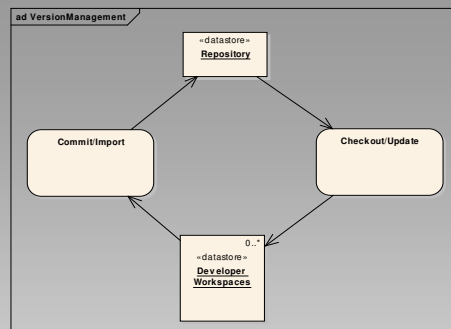
- ▶ Versionsmanagement
  - ▶ Persistierung und gezieltes Retrieval/Verteilung von allen freigegebenen Source-Code-Änderungen an Dateien und Modulen.
- ▶ Buildmanagement
  - ▶ Bau und Ablage definierter und konsistenter Binär-Versionen inkl. zugehöriger Dokumentation
- ▶ Qualitätsmanagement
  - ▶ Erstellen von Qualitätsberichten auf der Basis definierter Source-Code und Binärstände
    - ▶ Automatische Test-Reports
    - ▶ Code-Analyse
    - ▶ Manuell durchgeführte Testpläne.

Software Management



## Versionsmanagement

- ▶ Verschiedene Repositories
  - ▶ File-basiert (RCS)
  - ▶ Client/Server (CVS)
  - ▶ Datenbank-basiert (Perforce)
- ▶ Verschiedene Ablage-Modelle
  - ▶ Delta-Forward (RCS)
  - ▶ Delta-Backward (SVN)
- ▶ Verschiedene Versionierungsprinzipien
  - ▶ Version pro File (CVS)
  - ▶ Changelist/Version pro Pfad (SVN)

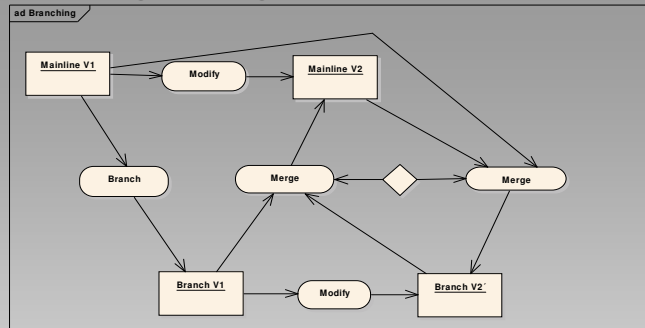


Software Management



## Branching und 3-Wege Merge

- ▶ Branch:
  - ▶ Release
  - ▶ Service Pack
  - ▶ Projekt
  - ▶ Branche
  - ▶ User-Branch



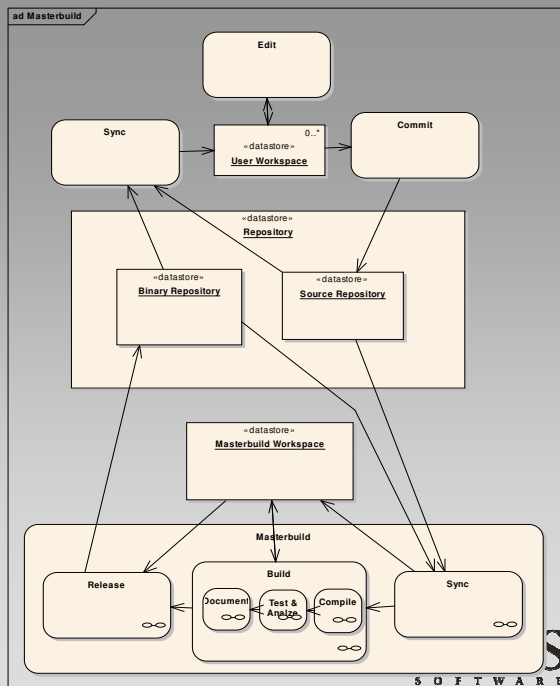
- ▶ 3 Wege Merge
  - ▶ Merge(Zieldatei,Diff(Alte Startdatei, Neue Startdatei))
  - ▶ Meist zeilenbasierter Merge
  - ▶ Manuelle Konfliktauflösung

Software Management

**DACOS**  
SOFTWARE

## Akademikergehänge, nutzlos!

- ▶ Martin Fowler, Matt Foemmel: *Continuous Integration*
- ▶ Jede Änderung am Source-Code stösst den Build-Prozess an.
- ▶ Dessen Ergebnisse werden ASAP den Entwicklern bereitgestellt.
- ▶ Pro Branch ein eigener „Prozeß“.

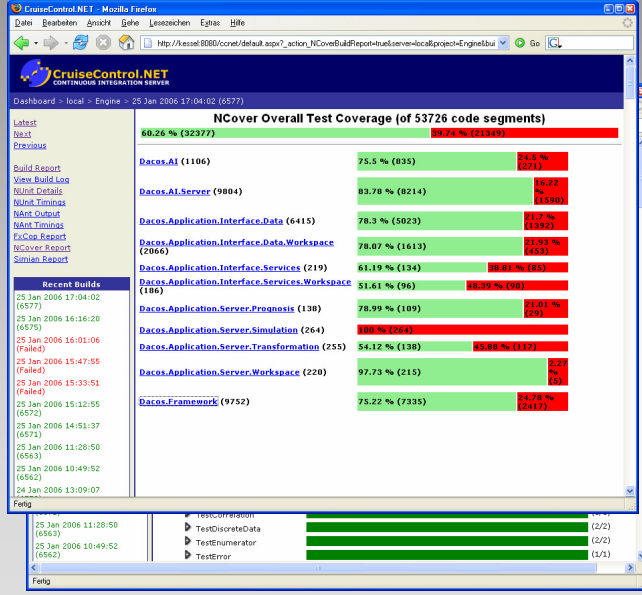


Software Management

SOFTWARE

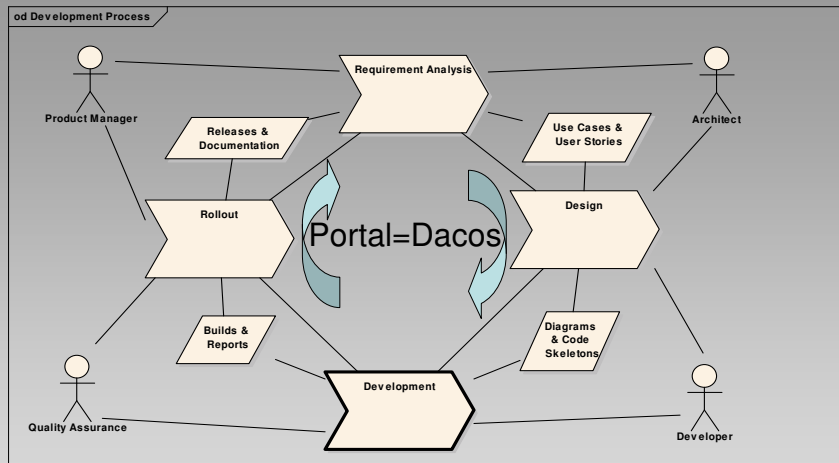
# Cruise Control .NET

- ▶ Compiler-Warnings
- ▶ Nunit-Details
- ▶ Performanz Kennzahlen
- ▶ (Byte-)Code Analyse
- ▶ (Test) Coverage Analyse
- ▶ Code Redundanz



Software Management

# Der Dacos Entwicklungsprozess



Software Management



**Wir simulieren die Welt!**

