

NuvoControl V1.0

B. Limacher

Ch. Imfeld

25-August-2009

Agenda / Inhalt

- Projektübersicht
- Demo
- Architektur
- Technologien / Methoden
- Entwicklungsumgebung / Softwareprozess
- Stundenübersicht
- Fragen

Projekt

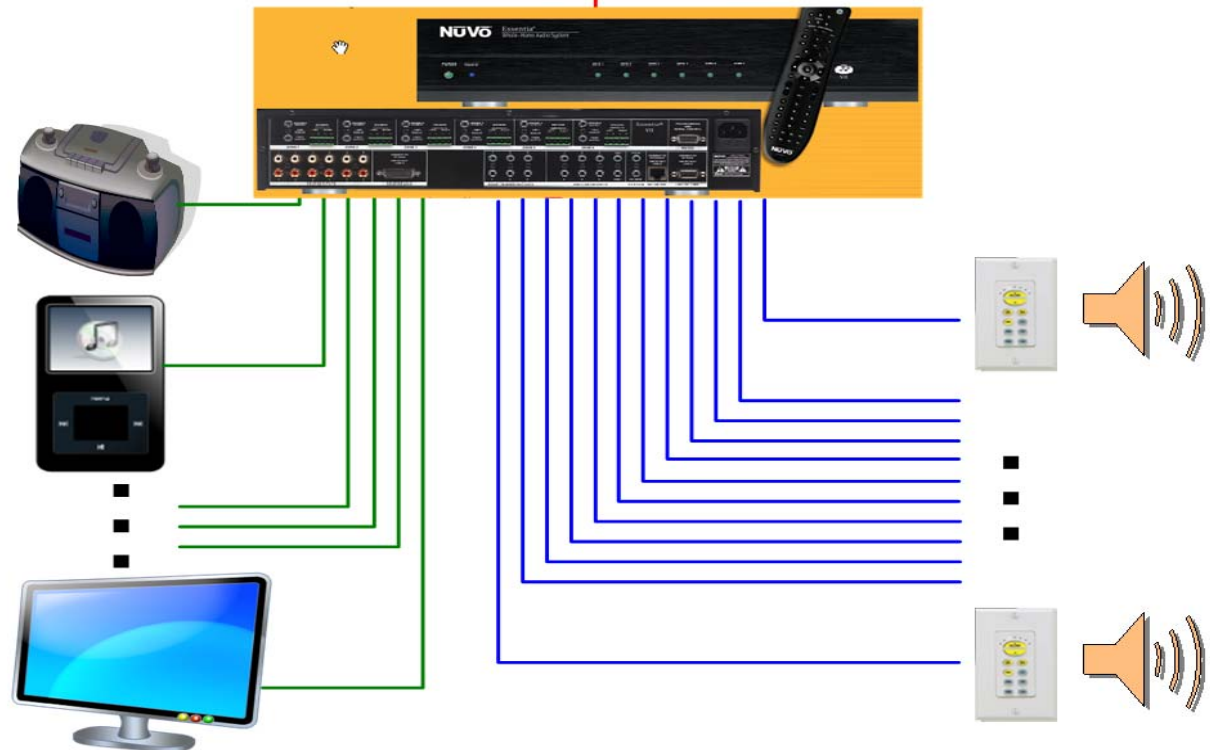
NuvoControl
Client



...

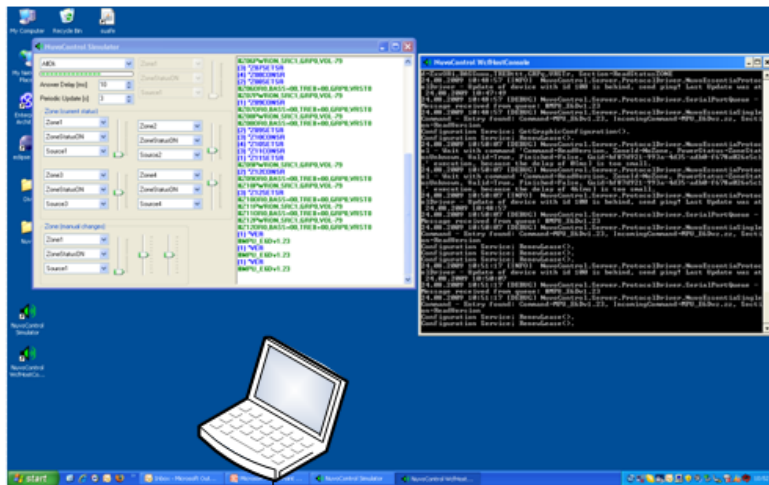


NuvoControl
Server



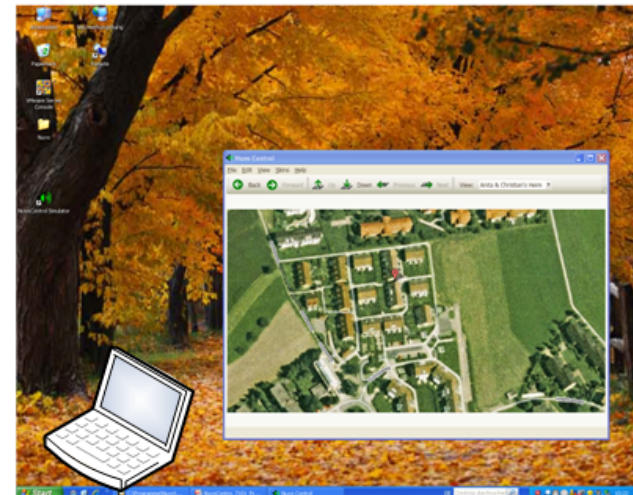
Demo Aufbau

NuvoControl
Simulator

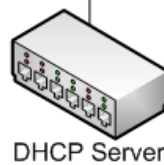


Chzugw9713
(192.168.1.100)

NuvoControl
Server



Imfi-laptopdell
(192.168.1.101)

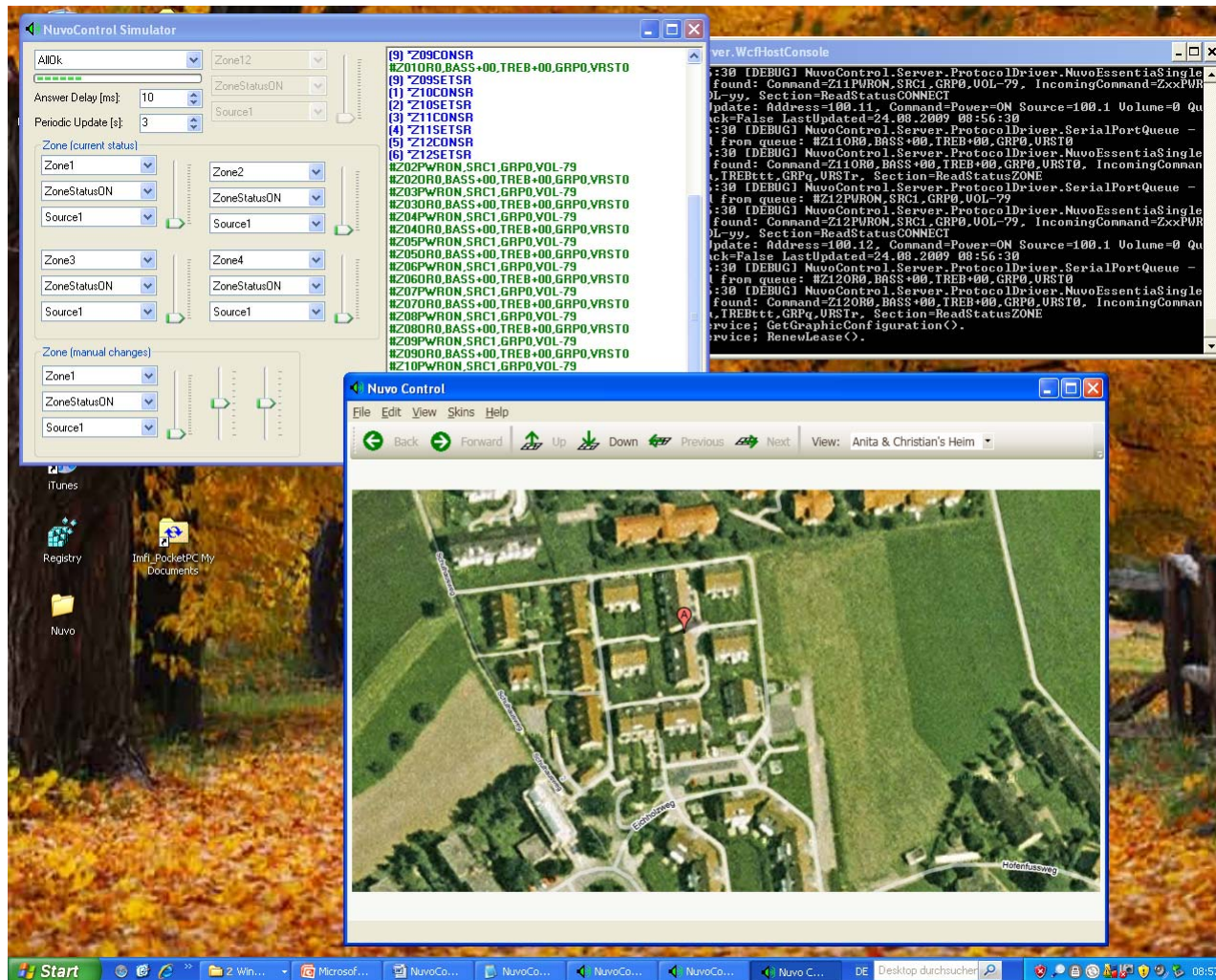


DHCP Server / Router
(192.168.1.21)

DHCP Server



Demo



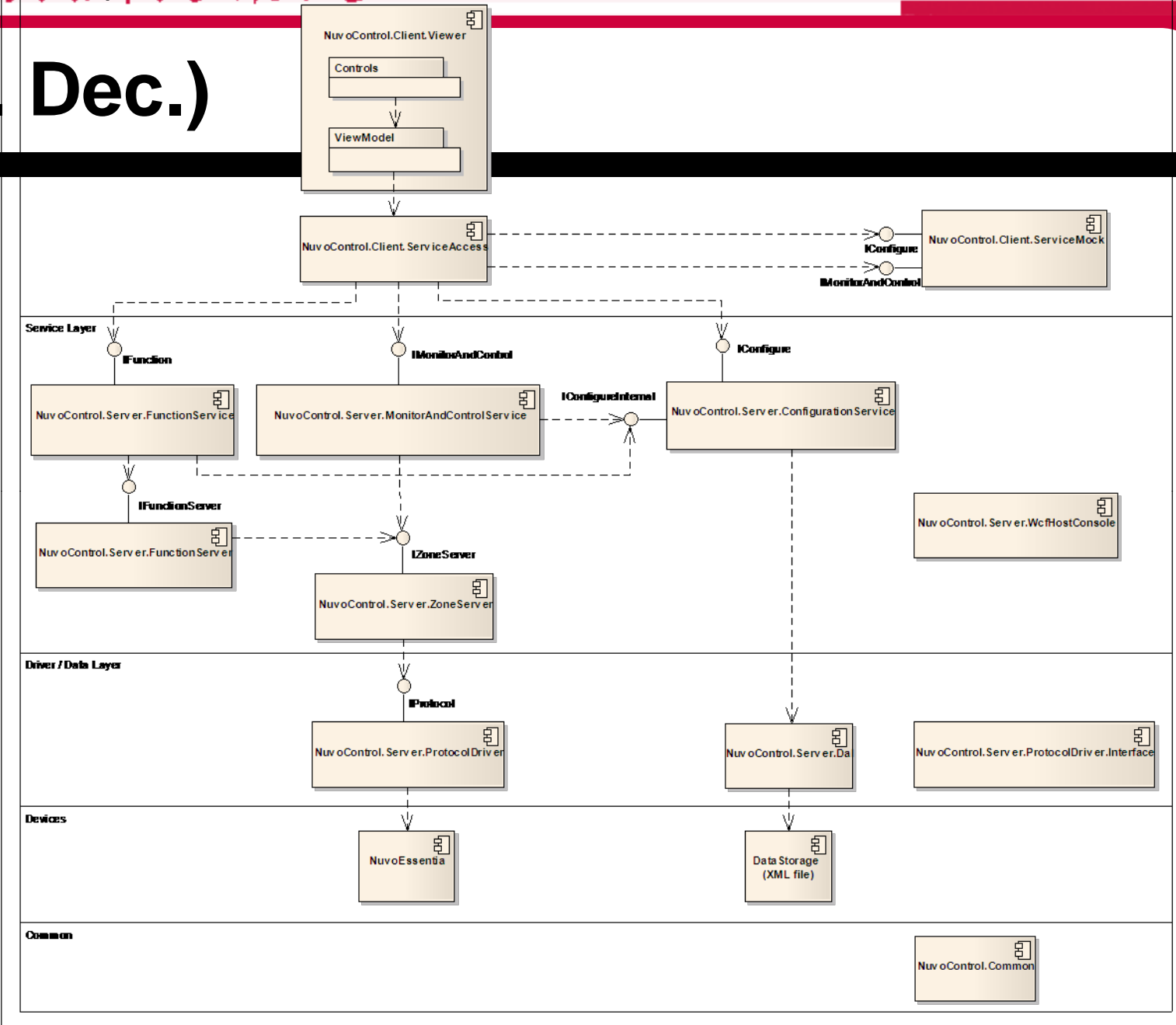
Architektur (Generell)

- Client-Server, multithreaded, stateful
RPC-Style -> synchrone Calls sind alle kurz.
- Subscription Pattern
um Clients über Statusänderungen der
Hardware asynchron zu notifizieren
- Multi-Client fähig
- System Konfiguration (Hardware, Graphische
Aspekte) ist in XML definiert.
=> Client wie auch Server werden dynamisch
gesteuert.

Architektur (Service)

- Ausgeprägtes Layering
Rückrufe von tieferen Layers nur mittels
Dependency Inversion,
Alle Layers sind über Interfaces gekapselt.
- Dependency Injection (light)
Jeder „Service Provider“ wird mittels einem
Konfigurator dem „Service User“ injiziert.
=> Voraussetzung für umfangreiche Unittests
- Unittests überall.

(A. Dec.)



Architektur (Service/Comm.)

- Service Hosting -> Console-App
(IIS oder WAS wäre auch möglich)
- Client-Service-Kommunikation mittels
WebServices
(single-channel und dual-channel)
=> HTTP und Text-Encoding
=> Reliable Message Delivery
=> Ordered Message Delivery
- Driver-Device-Kommunikation:
Seriell, MSMQ (Simulator)

Architektur (Client)

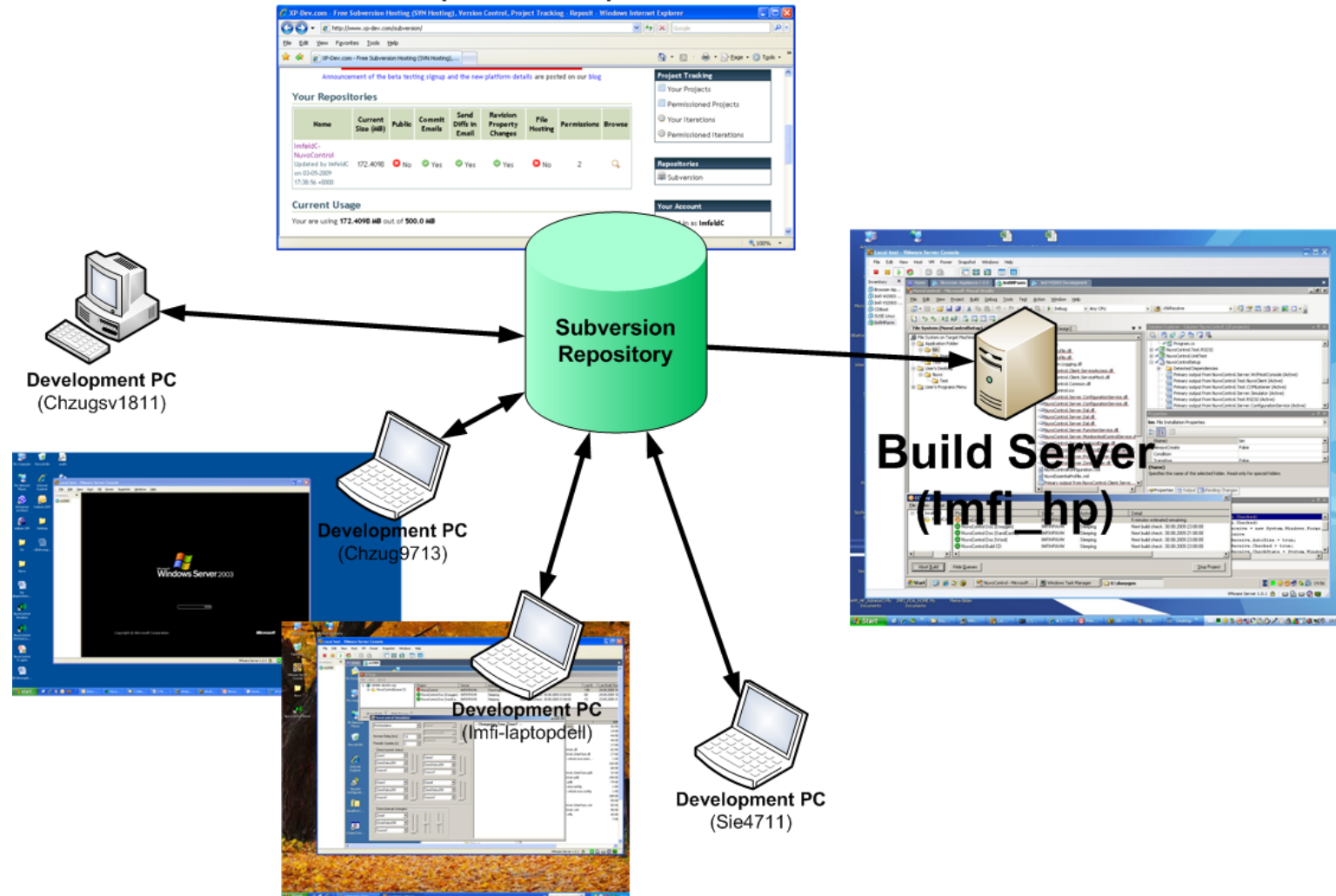
- Starke Trennung von:
 - UI-Layout
 - Controller-Logik
 - Styling
 - Service-Access
- => Layout und Styles leicht änderbar.
- => Unittests für Controller-Logik möglich

Technologien / Methoden

- Implementation mit C#
- Client-Server Kommunikation mit Web Services, implementiert mit WCF
- Server zu Multiraum-Simulator Kommunikation mit Queues (MSMQ)
- GUI Implementation mit Windows Presentation Foundation (WPF)
- Visual Studio 2008 mit dem Unit Test Framework von Microsoft (VSTS)
- Continuous-Build Integration mit CruiseControl.NET
- Source Code Verwaltung mit Subversion und Tortoise
- Dokumentation erstellt mit Hilfe von Microsoft Sandcastle (NDoc) und Doxygen
- UML Modellierung mittels Enterprise Architect (EA)
- Use-Case basiertes Requirements Engineering (nach Cockburn / Bianchi)
- Requirements Driven Development (RDD) (gemäss Wirfs-Brock / Tobler)
- Agiler Entwicklungsprozess gemäss SCRUM

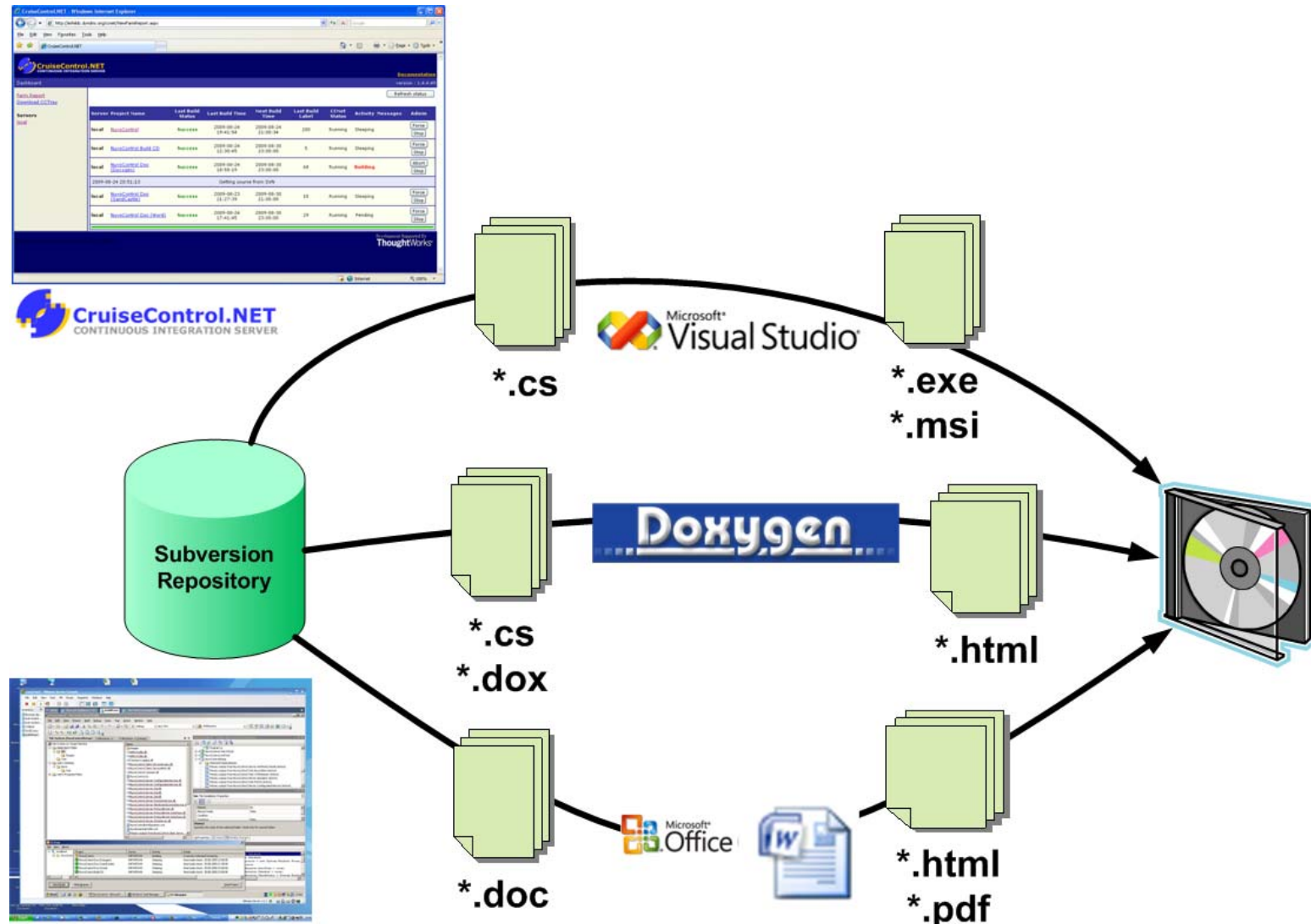
Entwicklungsumgebung

<http://www.xp-dev.com/>





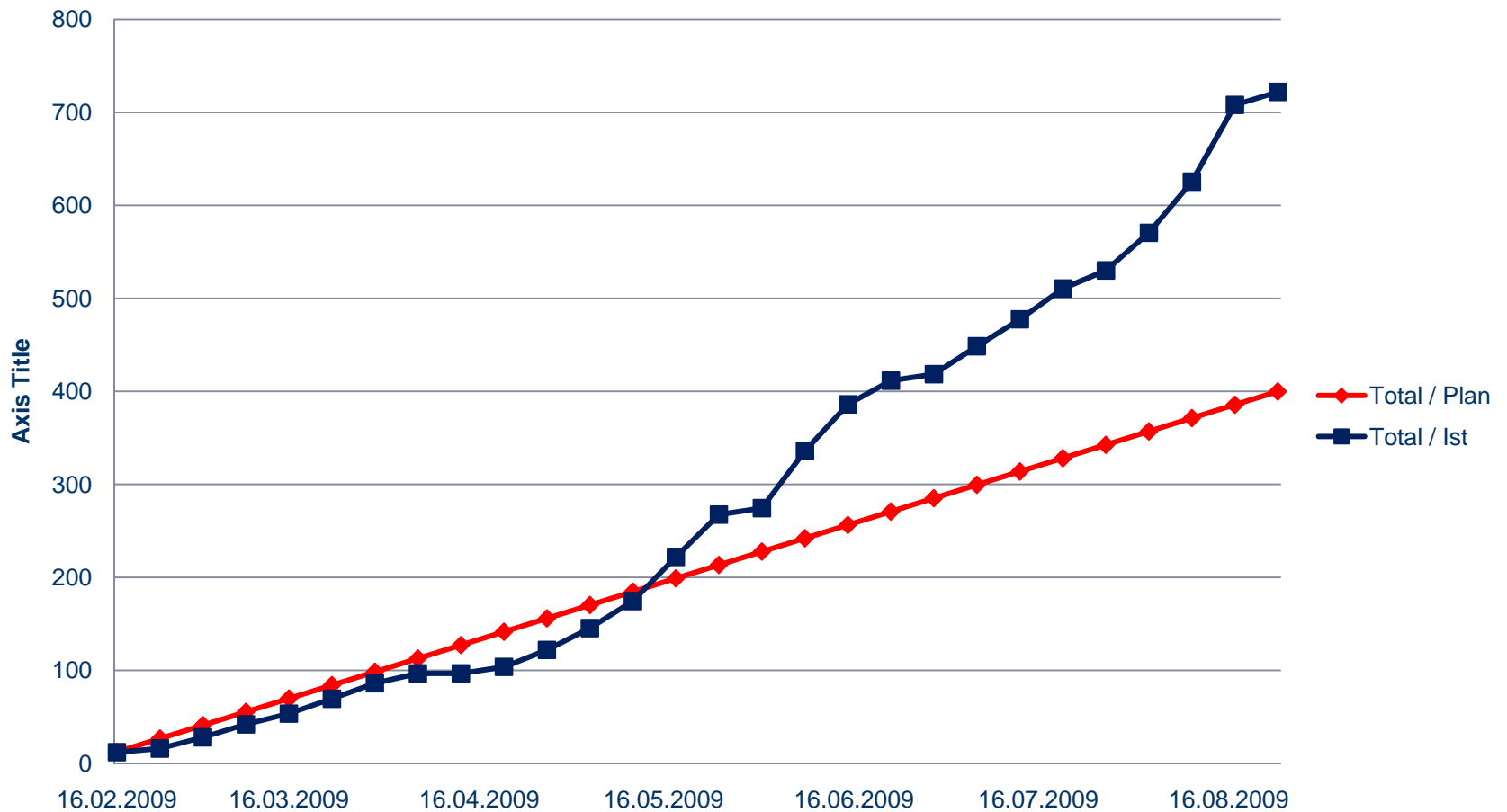
Softwareprozess





Stundenübersicht

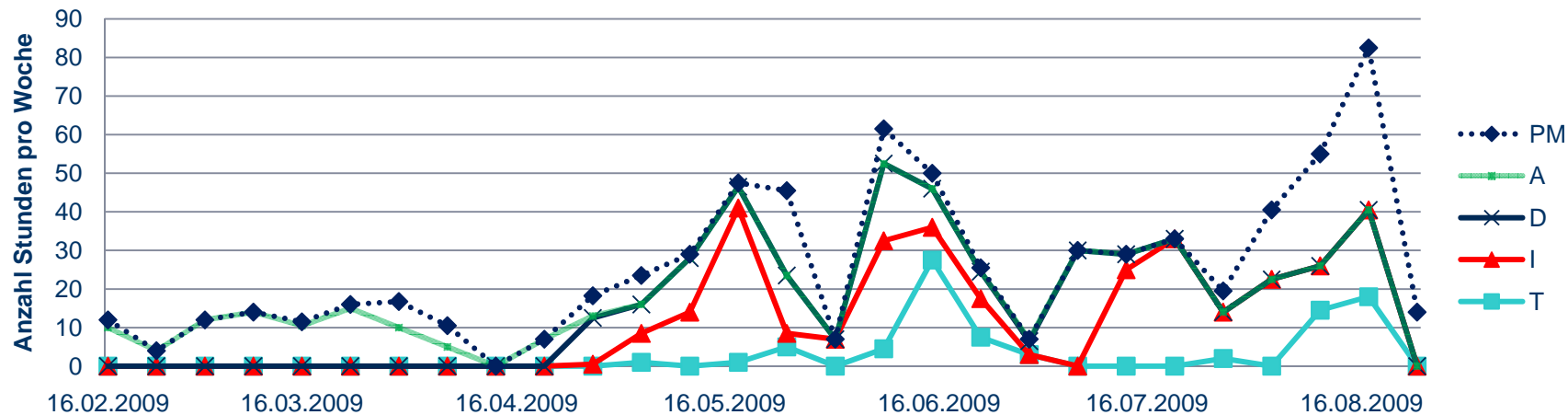
Plan / Ist Vergleich (Total)



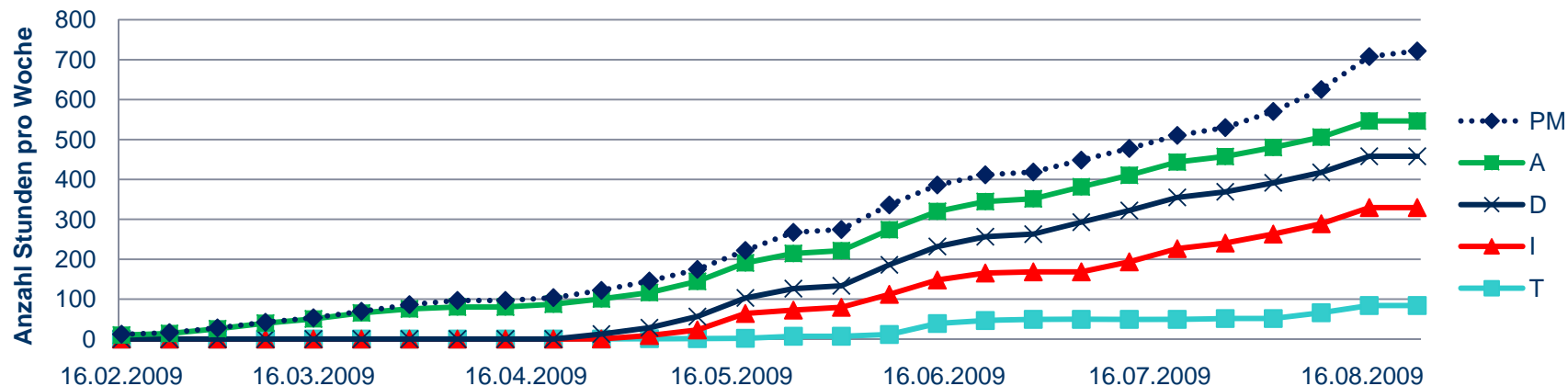


Stundenübersicht (Disziplinen)

Kummulierte Stunden pro Woche



Kummulierte Stunden pro Woche



Fragen

