

**Bereitstellung einer flexiblen KM-
Umgebung „eXtensible CM Toolkit“
für die Mobile Phone Software
Entwicklung basierend auf den
Telelogic SYNERGY Produkten**

Barbara Neeb-Bruckner und Asad Raja

CM, BenQ Mobile

Telelogic Anwenderkonferenz 2005, Berlin

BenQ

Abstract

- Der Vortrag beschreibt eine KM-Lösung → XCM Toolkit.
- Die Vortragenden, Barbara Neeb-Bruckner und Asad Raja, stellen zunächst BenQ Mobile und die Mobile Software Entwicklung vor.
- Sie geben dann eine Einführung in die Organisation des XCM Projekts und die Architektur des XCM Toolkit.
- Sie zeigen anhand von „Kernel“ Funktionalitäten auf, wie viel Flexibilität der XCM Toolkit Ansatz benötigt.
- Die Vorstellung der Einführungsstrategie und die bisher gesammelten Erfahrungen schließen den Vortrag ab.

Inhaltsverzeichnis

- 📱 BenQ Mobile
- 📱 Mobile Software Entwicklung
- 📱 XCM Toolkit „on top“ der SYNERGY Produkte
- 📱 Organisation des XCM Toolkit Projekts
- 📱 Architektur des XCM Toolkits
- 📱 Status des XCM Toolkit Projekts
- 📱 Beispiele für „Building Blocks“ im XCM Toolkit
- 📱 Einführung des XCM Toolkits in die SE- Projekte
- 📱 Gesammelte Erfahrungen

Inhaltsverzeichnis

- 📱 BenQ Mobile
- 📱 Mobile Software Entwicklung
- 📱 XCM Toolkit „on top“ der SYNERGY Produkte
- 📱 Organisation des XCM Toolkit Projekts
- 📱 Architektur des XCM Toolkits
- 📱 Status des XCM Toolkit Projekts
- 📱 Beispiele für „Building Blocks“ im XCM Toolkit
- 📱 Einführung des XCM Toolkits in die SE- Projekte
- 📱 Gesammelte Erfahrungen

BenQ Mobile

Change Management überall...

■ Übernahme von Siemens COM MD durch BenQ

■ Datum: 01.10.2005

■ BenQ's Vision: Bringing Enjoyment and Quality to Life

■ Gegründet: 1984

■ Zentrale: Taoyuan (Taiwan)

■ Mitarbeiter: 14.000 (bevor Übernahme)

■ Umsatz: US\$ 5 Milliarden (2004)

■ Produkte: LCD-Bildschirme, Mobile Phones, Projektoren, Speichermedien, Scanner, Digitalkameras, ...



BenQ Mobile

Change Management überall...

Die neue BenQ Mobile Company

Landesgesellschaften:

BenQ Mobile GmbH & Co. OHG,
BenQ Denmark ApS, etc.

 Zentrale: München

 Fertigung: Manaus (Brazil), Suzhou (China),
Kamp-Lintfort (Germany),
Mexicali (Mexiko)

 Mitarbeiter: 7000+ weltweit

 Umsatz: 5,6 Milliarden Euro

 Marktanteil: 2004: 7,3%

Mobile Software Entwicklung

Architektur

■ Produkte/Plattformen

- gemeinsame Komponenten mehrerer Produkte werden als Plattform entwickelt, eine Art „Software Framework“
- aus der Plattform werden Produkte abgeleitet und darin wird gegebenenfalls weitere Funktionalität entwickelt



■ Features

- neue Eigenschaften von Produkten werden in der nächsten Version der Plattform oder in den Produkten selbst bereitgestellt

■ Code Umfang

- je nach Plattform, z.B. bei der 75er Generation: ~5 Millionen Lines of Code (LOC), ~25.000 Dateien, 80 Komponenten

■ Programmiersprachen

- C, C++, Java

Mobile Software Entwicklung

Entwicklung / Test

■ Übersetzung

- UNIX-Systeme
- akzeptable Compile-Dauer: ca. 30 Min. für ein clean build

■ Test/Simulation

- Simulation des Handys auf dem PC oder Laden der Software in Field Test Handys

■ Teststufen

- Komponenten Test, Integrationstest, System Test, Field Test
- KM endet mit der Übergabe an System Test



Mobile Software Entwicklung

Verteilte Entwicklung

- Entwicklungsteams an mehreren Standorten
 - Einbindung von speziellem Know-how
 - Entwicklung von Produktvarianten für lokale Märkte



Inhaltsverzeichnis

- BenQ Mobile
- Mobile Software Entwicklung
- XCM Toolkit „on top“ der SYNERGY Produkte
- Organisation des XCM Toolkit Projekts
- Architektur des XCM Toolkits
- Status des XCM Toolkit Projekts
- Beispiele für „Building Blocks“ im XCM Toolkit
- Einführung des XCM Toolkits in die SE- Projekte
- Gesammelte Erfahrungen

XCM Toolkit „on top”

Wie „lean” können die SYNERGY Produkte eingesetzt werden?

Komplexität aufgrund des Mengengerüsts

-  in der größten Plattform/Produkt Entwicklung: 800 Entwickler und 20 Build Manager in knapp 60 Datenbanken
-  zusätzlich weitere Plattform/Produkt Entwicklungen
-  KM Support weltweit: ~65 Mitarbeiter in unterschiedlichen OEs

Komplexität aufgrund der Verteilung

-  sechs Entwicklungsstandorte
-  Datenbanken vor Ort an den Standorten
-  CM Support weltweit lokal verfügbar (Zeitzone!)

Komplexität aufgrund der Entwicklung

-  Teile der Plattform sind zugeliefert, manchmal sogar die Ganze
-  Plattformen stellen Features für alle darauf basierenden Produkte bereit, manchmal mehrere Varianten desselben Features
-  DCM Verbund von Datenbanken

XCM Toolkit „on top“

Motivation und lässt sich eine KM-Lösung überhaupt zentral aufsetzen?

■ Vorteile einer KM-Lösung

- sie lässt sich zentral „Vordenken“
- neue Funktionen können integriert und dann ausgerollt werden
- der höhere Dokumentationsgrad hilft den Kunden und dem KM First Level Support
- die KM Verfahren im DCM Verbund sind ähnlicher als wenn die KM'ler an den Standorten eine abgestimmte Lösung individuell umsetzen oder „das Rad jeweils neu erfinden“

■ Erfolgreicher Vorgänger CMT = Configuration Management Tool

- kleines Produktentwicklungsteam von 4 – 5 Spezialisten = „KM Think Tank“
- beginnend bei CMT 2.0 ist heute nach vier Jahren CMT 18.4 im Einsatz, mit einer drastisch erweiterten Funktionalität im Vergleich zur Anfangslösung

XCM Toolkit „on top“

Warum eine neue KM-Lösung?

- Zunehmende Komplexität, kurze Produktzyklen, zielgruppen-spezifische Produktvarianten
 - erfordern flexible Entwicklungsprozesse und in der Folge anpassungsfähige Supportprozesse
 - ➔ sich ändernde KM-Lösung
- manchmal ist ein Re-Design / Re-Implementation besser als das Aufsetzen auf dem bisherigem Design
- Schlagworte dieser kundengetriebenen Neuanforderung
 - „wir wollen mit unterschiedlichen Planungsansätzen entwickeln“
 - „gebt uns den Developer Client [statt Classic Client und CLI]“
 - „wir wollen alles überall bearbeiten“

XCM Toolkit „on top“

Der Weg zur neuen KM-Lösung

- aufgrund dieser Kundenwünsche wurden zwei KM Assessments durchgeführt
- dabei entstanden Vorstudien einer neuen KM-Lösung: LCM 0.5, LCM 1.0
- Entscheidung zur Zusammenführung von CMT und diesen Vorstudien
- Bereitstellung einer KM-Lösung maßgeschneidert für die Erfordernisse der Mobile Software Entwicklung
→ e**X**tensible **CM** = XCM Toolkit

Organisation des XCM Toolkit Projekts

Wir leben, was wir unterstützen wollen ...

- joint: von weltweit ca. 65 Mitarbeitern der KM Gruppe sind 11 an der Entwicklung von XCM beteiligt
- multisite: sie sind beheimatet in Deutschland, Dänemark und China
- planned: das Projektmanagement basiert auf dem agilen Prozess SCRUM
- effort: die Architektur des XCM Toolkits wird genutzt, um Verantwortlichkeiten und Arbeitspakete zu schneiden
- “die eigenen Kleider tragen”
 - das entstehende XCM Toolkit wird schrittweise eingesetzt, um die eigene Entwicklung KM-technisch zu unterstützen

Architektur des XCM Toolkits

Wer setzt das XCM Toolkit ein und wie?

■ „Kunden“

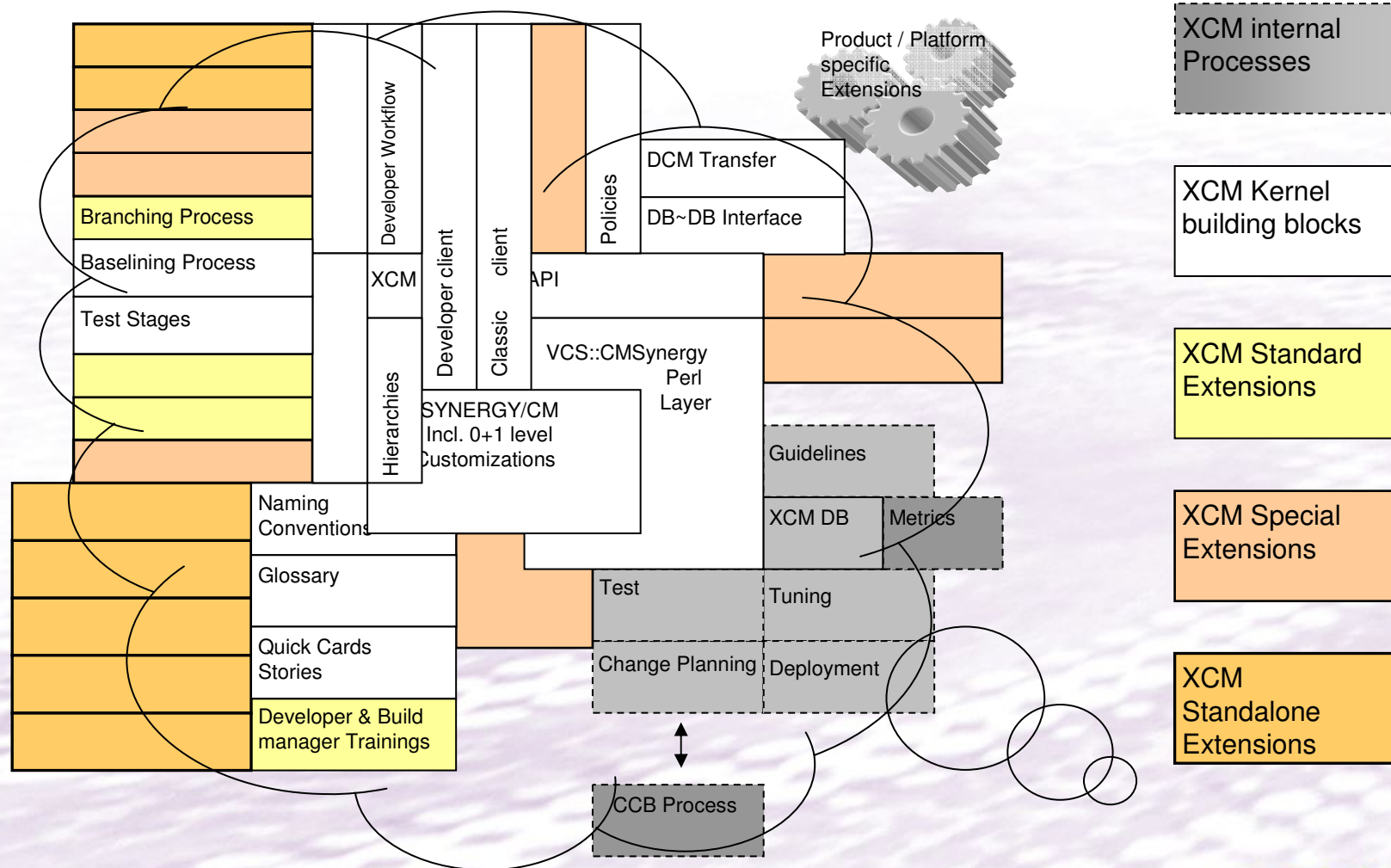
- das sind die Handy Plattformen mit ihren auf der Plattform basierenden Produkten

■ Der XCM Toolkit Gedanke → „Kernel & Extensions“

- Kunden wählen die Teile des XCM Toolkits, die sie für ihren Entwicklungsprozess benötigen
- dabei ist die Nutzung des „Kernel“ obligatorisch, die Nutzung von „Extensions“ fakultativ
- „Extensions“ benötigen den „Kernel“, können aber projekt-spezifisch angepasst werden
- Änderungen und neue „Extensions“ werden vom Kunden im „CCB Prozess“ beauftragt
- modifiziert der Kunde das XCM Toolkit selber, profitiert er nur eingeschränkt von zentralem Support und Weiterentwicklung

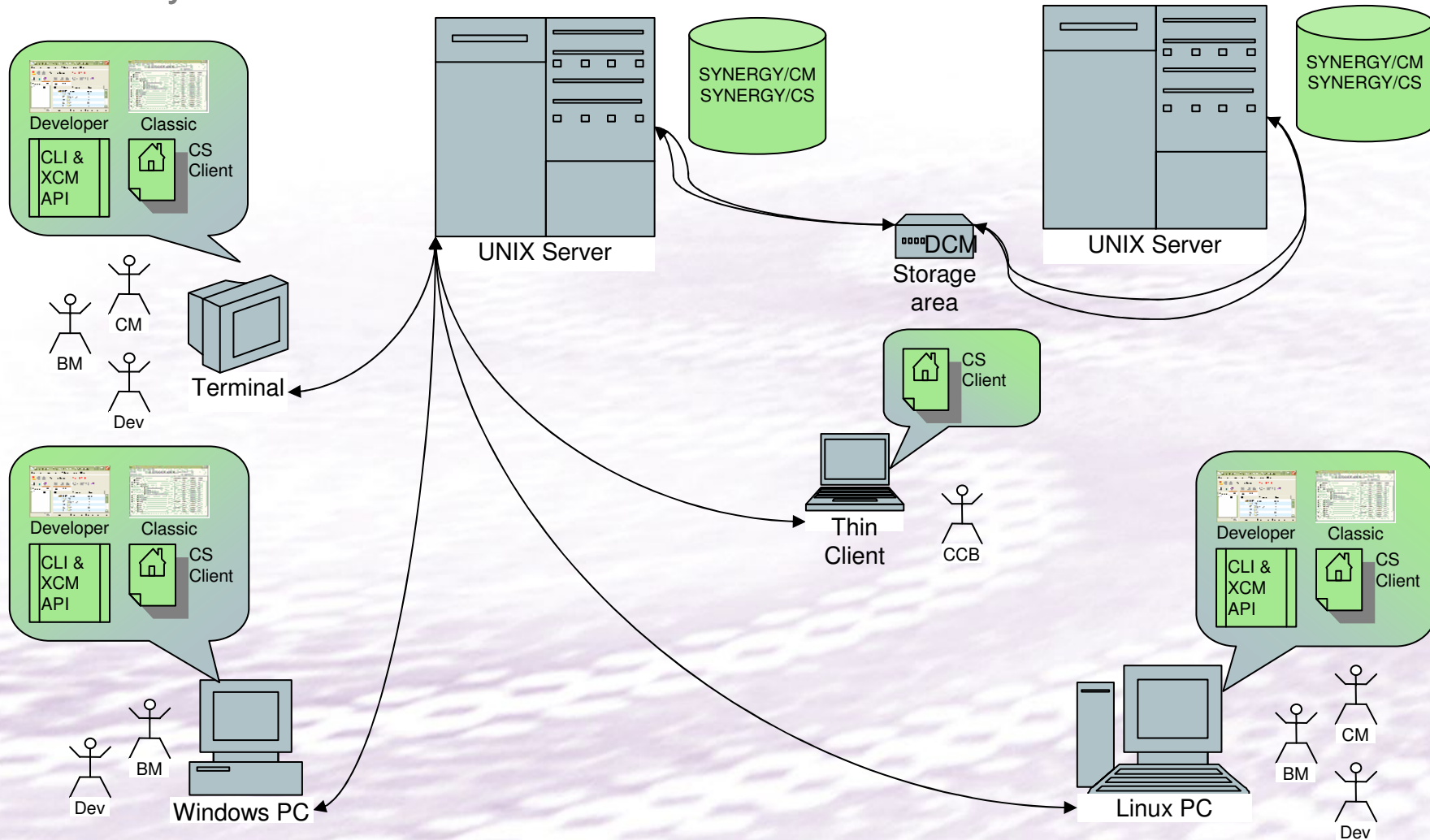
Architektur des XCM Toolkits

XCM Softwarearchitektur



Architektur des XCM Toolkits

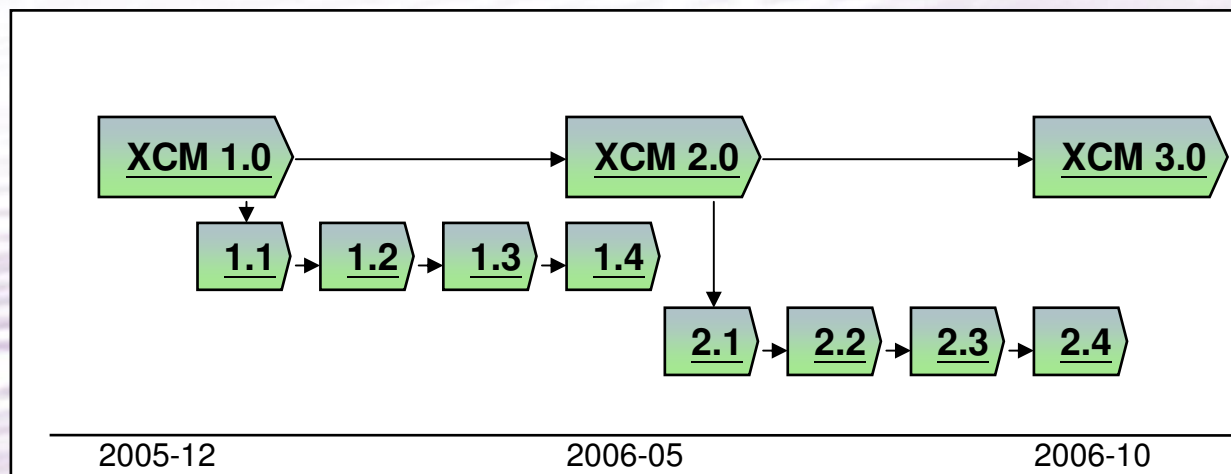
XCM Systemarchitektur



Status des XCM Toolkit Projekts

Wo stehen wir zur Zeit?

- „XCM 1.0 ist kurz vor der Vollendung
 - es werden die XCM internen Prozesse genutzt
 - es enthält die als „Kernel“ markierten „Building Blocks“ aus der XCM Softwarearchitektur
 - es unterstützt die Planungs-, Entwicklungs- und Testprozesse
 - es unterstützt „multi-site“ Entwicklung mit verteilten Entwicklungsteams
 - es automatisiert eine Reihe von KM und Build Manager Aufgaben



Beispiele für „Building Blocks“

Vorstellung einiger „Building Blocks“ aus der XCM Softwarearchitektur

■ „Kernel“

- DCM Transfer
- DB~DB Interface mit Channels und Interfaces
- XCM API mit dem Persistence Layer, Logging etc.
- Test Stages
- Baselineing Process
- QuickCards and Stories
- ...

■ XCM interne Prozesse

- CCB Process
- ..



Beispiele für „Building Blocks“

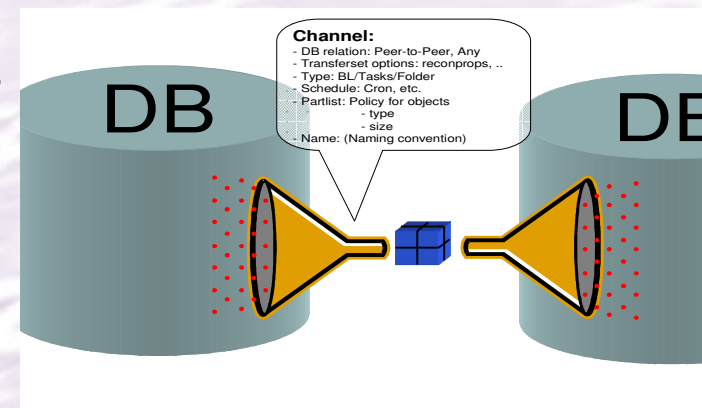
Kernel Building Block – DCM Transfer

- Zuständig für den Transport von Paketen
- DCM Transfer ist
 - ein reiner Transport Mechanismus,
 - Transport von bereits generierten und in der Ursprungs Datenbank abgelegten Paketen
 - über ein oder mehrere „Storage Areas“
 - zur jeweiligen Ziel Datenbank
 - der das „Routing“ und
 - die sichere Zustellung
 - sowie die Unversehrtheit der Pakete gewährleistet.
- Zukünftige Versionen des DCM Transfer könnten
 - ein „re-routing“ initiieren falls eine „Storage Area“ ausfällt
 - das „Tracking“ der Pakete erlauben („wo ist mein Paket gerade“)
 - sowie dem Absender eine Zustellbestätigung zukommen lassen

Beispiele für „Building Blocks“

Kernel Building Block – DB~DB Interface mit Channels und Interfaces

- Zuständig für das Zusammenstellen, Generieren und Empfangen von Paketen
- „Channels“ sind ein genereller Mechanismus, um die Schnittstellen zwischen Datenbanken zu beschreiben
 - Liste der Datenbanken „wer mit wem“ Informationen austauscht
 - Austausch als dediziertes Paket oder als ANY Paket
 - Query zur Bestimmung der „einzupackenden“ Objekte
 - Zeitpunkte / Ereignisse, wann das Paket zusammengestellt werden soll
 - Aktionen, die beim Empfangen eines Pakets ausgeführt werden sollen



Beispiele für „Building Blocks“

Kernel Building Block – DB~DB Interface mit Channels und Interfaces

- Interfaces nutzen „Channels“ zur Implementierung einer Schnittstelle
- Beispiele für Schnittstellen um den Entwicklungsprozess zu unterstützen
 - „completed“ Tasks hochfrequent im DCM Verbund verteilen („logische Datenbank“)
 - bei Verfügbarkeit einer neuen Plattform Baseline diese an alle Produkt-Datenbanken im DCM Verbund ausliefern
- Beispiele für Schnittstellen um die KM Administration zu unterstützen
 - neue Typen oder geänderte Typdefinitionen im DCM Verbund propagieren
 - die eingetragenen Benutzer je Datenbanken zentral verfügbar machen z.B. für SYNERGY/Change

Beispiele für „Building Blocks“

Kernel Building Block – XCM API mit Persistence Layer, Logging etc.

- Implementierung des XCM Toolkits in Perl
 - SYNERGY/CM CLI API ← VCS::CMSynergy Perl Layer
← Bibliothek XCM API mit generell verwendbaren Routinen
- „Persistence Layer“ (ein besonderer Wunsch)
 - Ablage von Metadaten zum XCM Toolkit in der Datenbank
 - Operationen zum Anlegen von Objekten ohne Bindung zu einem Project und zur Erzeugung von Versionen dieser Objekte
- Tracing / Logging
 - Unterstützung mehrerer Tracelevel für XCM Toolkit Skripte
 - Protokollierung von Anzahl und Dauer der Ausführung
- POD Dokumentation
 - Nutzung eines Skript „Templates“ für POD Dokumentation
 - erzeugt: „Usage“, „Man Page“, „Implementation Notes“

Beispiele für „Building Blocks“

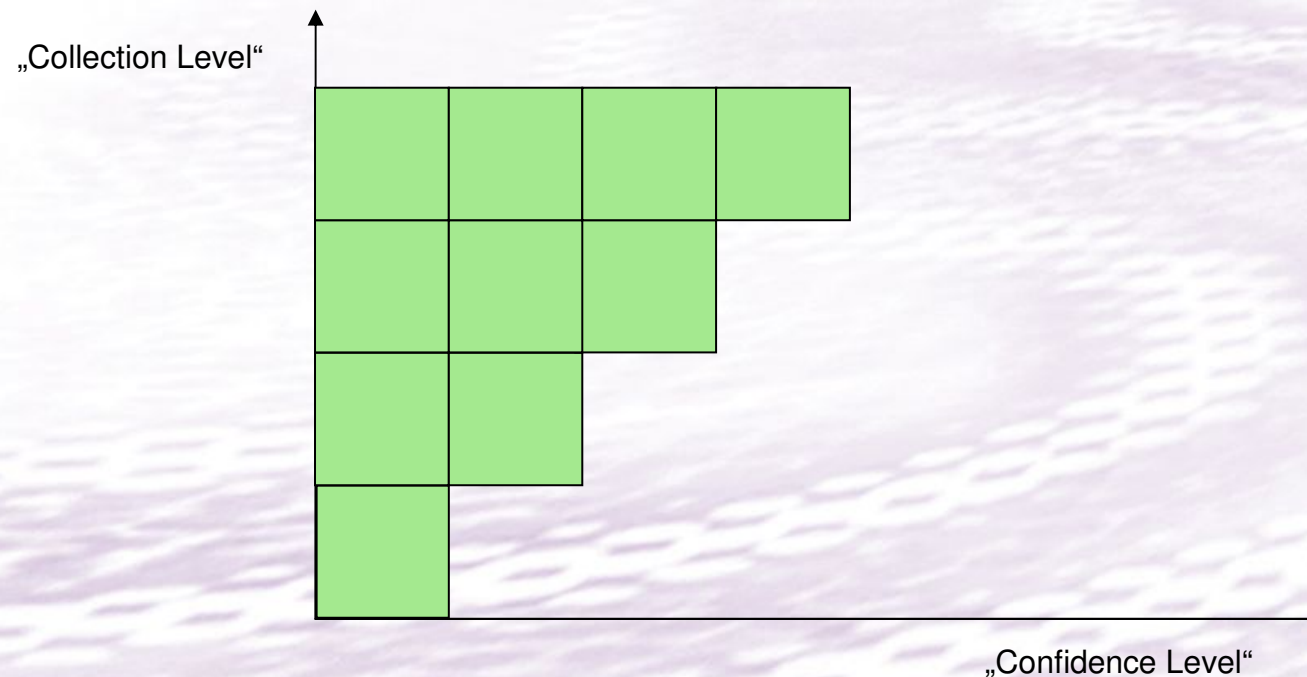
Kernel Building Block – Test Stages

- Freigabestufen kennt jede Softwareentwicklung, die Anzahl und die Benennung unterscheidet sich aber.
 - Freigabestufen haben gemeinsame Merkmale
 - geänderte Teile der Software werden auf einer Stufe gesammelt und bei Testerfolg an die nächste Freigabestufe gereicht
 - Anzahl der Freigabestufen im XCM Toolkit ist projekt-spezifisch
- „Collection Level“ - welche Änderungen sind zu sammeln
 - für eine Komponente oder eine Eigenschaft des Systems, für eine Gruppe von Eigenschaften, für zu integrierende oder auszuliefernde Eigenschaften
- „Confidence Level“ - welchen Qualitätskriterien genügen die gesammelten Änderungen der Software
 - sind sie übersetzbar im „daily build“, haben sie den „smoke test“ passiert, einen User Test oder schon den System Test, ...

Beispiele für „Building Blocks“

Kernel Building Block – Test Stages

- Nicht jeder „Confidence Level“ ist auf jedem „Collection Level“ sinnvoll
 - der „smoke test“ ist schon für eine einzelne testbare Eigenschaft möglich, der System Test erst, wenn alle geplanten Eigenschaften verfügbar sind



Beispiele für „Building Blocks“

Kernel Building Block – Baseline Process

Anforderungen

-  Baseline = getestete Konfiguration
-  Einfrieren des Inhalts einer beliebigen Test Stage
-  Zugrunde liegende Test Stage muss erkennbar sein
-  Tasks und deren Aufteilung in Folder müssen unverändert in der Baseline vorhanden sein
-  Unterstützung einer kompletten Projekt Hierarchie, einer Untergruppe oder eines einzelnen Projektes
-  Erzeugung „vorläufiger“ Baselines

Design Entscheidungen

-  Kopieren und Einfrieren von Foldern
-  Nutzung des Baseline Objects
-  Auschecken von der Test Stage oder letzten Baseline
-  Project History View, etc.

Beispiele für „Building Blocks“

Kernel Building Block – QuickCards und Stories

QuickCards

- Beschreibung häufig wiederkehrende Tätigkeiten
- Extrakt aus anderen Dokumenten
- rollenspezifisch, z.B. für developer, build_mgr

QuickCard Beispiel:

SIEMENS

QuickCard: XCM Developer Environment

Overview

This QuickCard describes how to set up the environment within Synergy/CM¹ for developers working in XCM development projects.

Developers are supposed to be using the "CM Synergy for Developers Client" on UNIX, so this document follows the terminology and usage of this client.

The following topics are covered:

- > [Start the Client](#)
- > [Use Help](#)
- > [Create a Working Project Hierarchy](#)

Start the Client

First the CM Synergy Client has to be started.

Step	Action
1.	Login on UNIX host (e.g. <code>krusty (xxx.12.345.678)</code>)
2.	If this is the first time that you use CM Synergy version 6.3, please check the version using the command <code>ccm version</code>. If the version is 6.3, please continue with step 3. If it is not, please update your default environment²: > Create a backup of your current personal profile: <code>cp \$HOME/.cshrc.private \$HOME/.cshrc.private.bak</code> > Open <code>\$HOME/.cshrc.private</code> with your favorite text editor and append the following lines to the end of <code>.cshrc.private</code> <pre>if (-f /tools/continuous/xcm/etc/cshrc.lcm) then source /tools/continuous/xcm/etc/cshrc.lcm endif</pre> > Save the file and close the editor. > Open a new shell console.
3.	Start the "CM Synergy for Developers - Client" with the command <code>cmsynergy &</code>
4.	Enter or select the correct CM Synergy database path (ask your build manager or team leader if in doubt) and the engine host (e.g. <code>krusty</code>) in the login dialog and continue with the client start-up.
5.	Set the environment variable <code>CCM_ADDR</code> to the correct Synergy session using the following command: <pre>setenv CCM_ADDR <hostname:portnumber:IP-address></pre> The correct CCM address can be found with the command: <pre>ccm status</pre>

¹ currently CM Synergy 6.3

² The commands assume that you use `tcsh` (default shell at SLC MP). For other shells (e.g. `ksh`, `bash`) please set the environment variables defined in `.cshrc.xcm` according to the syntax used in your shell.

Beispiele für „Building Blocks“

XCM interner Prozess - CCB Process

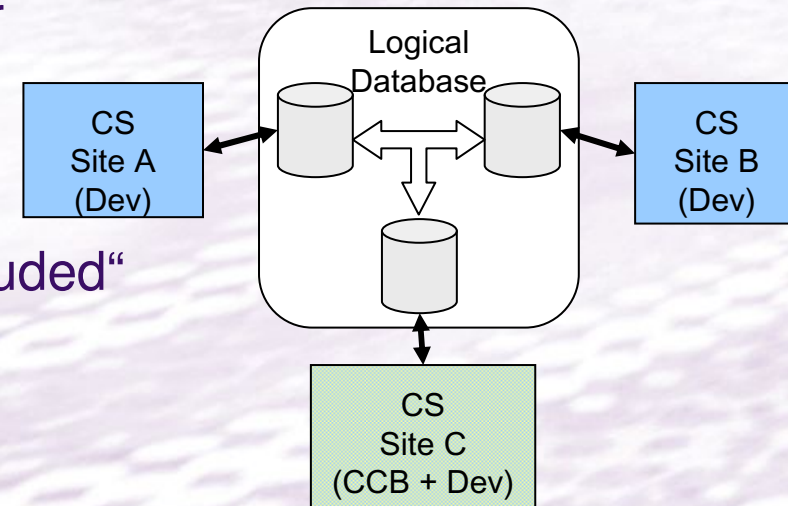
- hat sich in einfacherer Form schon in früheren Projekten und produktiven KM Systemen bewährt
- deckt Entwicklungsphase des XCM Toolkits und seinen produktiven Einsatz ab
- „Change Control Board“
 - Vertreter und KM-Experten des Kunden, Mitglieder von XCM
 - CR-Status: „evaluated“ | „deferred“ → “accepted”
- „Escalation Board“
 - weitere Eskalationsinstanz
 - Kunden-PLs und XCM Lead
 - CR-Status: „evaluated“ | „deferred“ → „in_escalation“ → „accepted“

Beispiele für „Building Blocks“

XCM interner Prozess - CCB Process

■ „Logical Database“

- Master Site für Erfassung und Freigabe durch CCB, CR-Status: ... → „accepted“
- Übergabe an die für Implementierung und Test verantwortliche Site, CR-Status: „accepted“ → „assigned“ → „resolved“ → „tested“
- Getesteter CR geht an die für das Deployment verantwortliche Master Site CR-Status: „tested“ → „concluded“



Inhaltsverzeichnis

- BenQ Mobile
- Mobile Software Entwicklung
- XCM Toolkit „on top“ der SYNERGY Produkte
- Organisation des XCM Toolkit Projekts
- Architektur des XCM Toolkits
- Status des XCM Toolkit Projekts
- Beispiele für „Building Blocks“ im XCM Toolkit
- Einführung des XCM Toolkits in die SE- Projekte
- Gesammelte Erfahrungen

Einführung in die SE- Projekte

Wünsch dir was...

Modularität

-  Kunde installiert die Teile des XCM Toolkits, die aus Projekt-
erfordernissen nötig sind
-  „Kernel“ ist obligatorisch, „Extensions“ sind fakultativ

Anpassbarkeit

-  „Extensions“ benötigen den „Kernel“, können aber projekt-
spezifisch angepasst werden
-  Kunde kann Änderungen und Erweiterungen des XCM Toolkits
über „CCB Process“ beauftragen

Support & Training

-  Support und Weiterentwicklung für (unmodifiziertes) XCM Toolkit
-  für jeden „Building Block“: „Usage Guide“ und „Reference
Manual“; blockübergreifend: „QuickCards“ und „Stories“
-  (rollenspezifische) Trainings

Gesammelte Erfahrungen

Wenn wir es wieder tun würden...

- hinsichtlich der verteilten Entwicklung von XCM
 - Vorteil - Einbindung verschiedener Standorte von Anfang an
 - Herausforderung - mehr Koordinationsbedarf (Zeitverschiebung)
 - Hilfreich -
 - tägliche Telefonkonferenz, SCRUM
 - verteilte Script- und Dokumentendatenbank
 - verteiltes Änderungsmanagement basierend auf SYNERGY/Change
- hinsichtlich des XCM Toolkits
 - mehr Zeit, die Architektur genauer vorzugeben
 - hinsichtlich der Abgrenzung von „Building Blocks“ zueinander
 - hinsichtlich der Schnittstellen zwischen den „Building Blocks“
 - Verfügbarkeit eines Basis-Glossars vor dem ersten Dokument
- hinsichtlich “die eigenen Kleider tragen”
 - fördert Test und Disziplin

Kontakt

DANKE für ihr Interesse!

Wenn Sie weitere Fragen haben, können Sie sich gerne an uns wenden

Barbara Neeb-Bruckner

BenQ Denmark ApS

barbara.neeb-bruckner@siemens.com

oder

info@barbara-neeb.de

www.barbara-neeb.de

Asad Raja

BenQ Mobile

asad.raja.ext@mch.siemens.de

oder

argumentum GmbH

raja@argumentum.de

www.argumentum.de

Vortragende

Dipl. Math. Barbara Neeb-Bruckner

Projektmitglied XCM Toolkit Entwicklung

- bei BenQ Denmark ApS

Senior Beraterin

- Spezialistin für Software-Konfigurationsmanagement und Qualitätsmanagement
- associated Professional Services Partner von 

- weitere Informationen unter info@barbara-neeb.de

Vortragende

Dipl. Inf. Asad Raja, argumentum

Projektleiter XCM Toolkit Entwicklung

- bei BenQ Mobile GmbH & Co. OHG

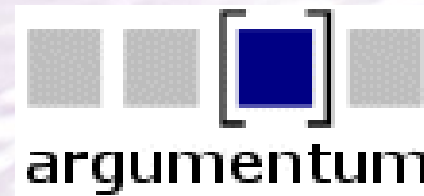
Senior Berater - argumentum GmbH

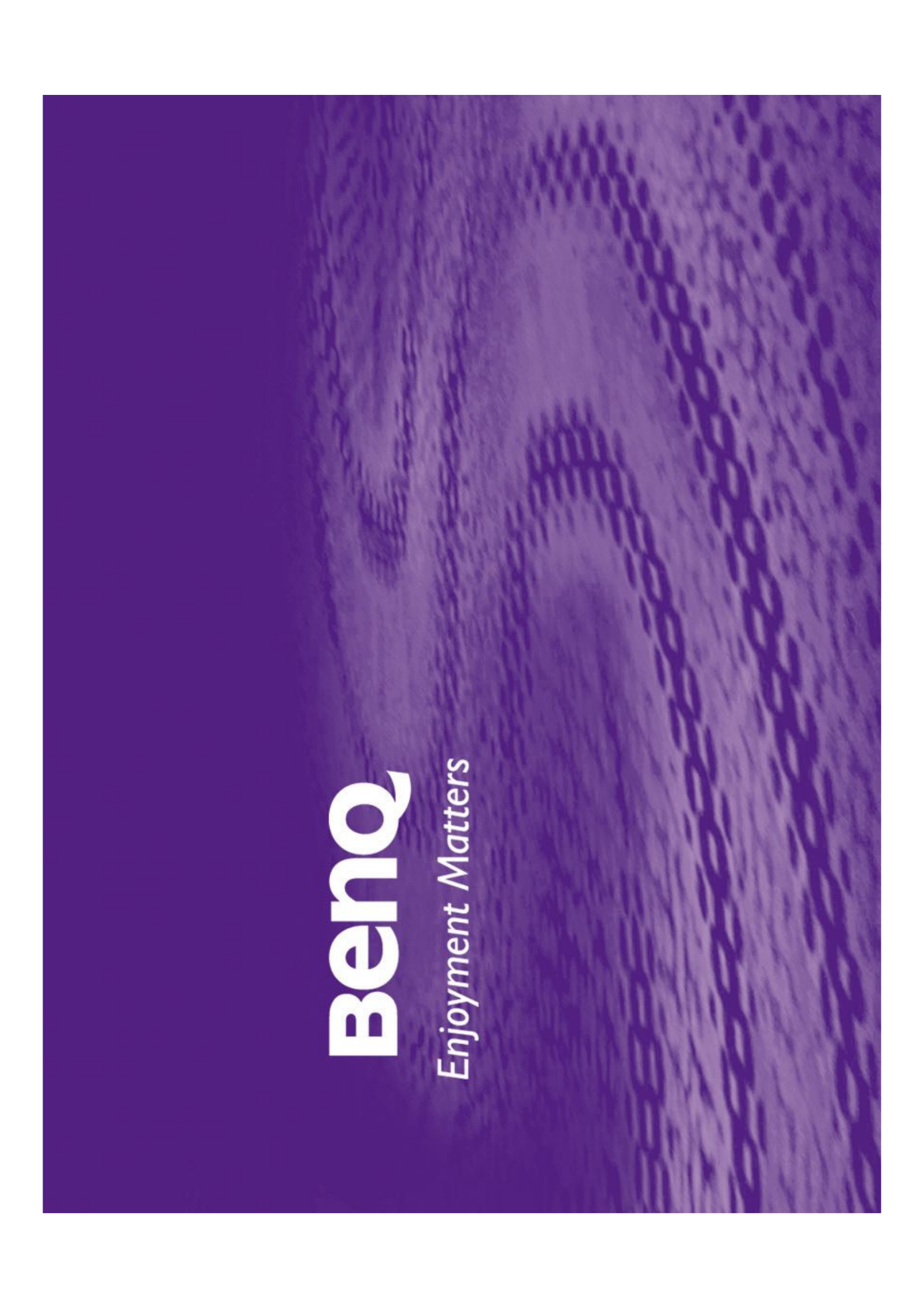
- Experten für Software-Konfigurationsmanagement und Test-Management
- Consulting Partner

von



- weitere Informationen unter <http://www.argumentum.de>





BenQ

Enjoyment Matters