

E-Quality

ACCU 2002

Nicolai Josuttis

Jutta Eckstein

The Project

- **Goal:**
 - mission critical banking application
 - 50.000 simultaneous users
 - low bandwidth
- **More than 100 software developers**
- **First approach failed after some years**
- **Thus, not much time and trust for new approach**
 - 3 months for architecture and „referential implementation“

Force Transparency

- **Communication and Transparency for**
 - developers
 - QA
 - controlling / revision
 - customers
- **Practices and Values:**
 - shared ownership
 - shared knowledge
 - shared skill
 - no head monopolies
 - honesty
- **Introduction of a Wiki-Web**
 - twiki.org

Copyright 2002 by N. Josuttis and J. Eckstein

3

NICOLAI M. JOSUTTIS
solutions in time

Jutta Eckstein
Objects in Action

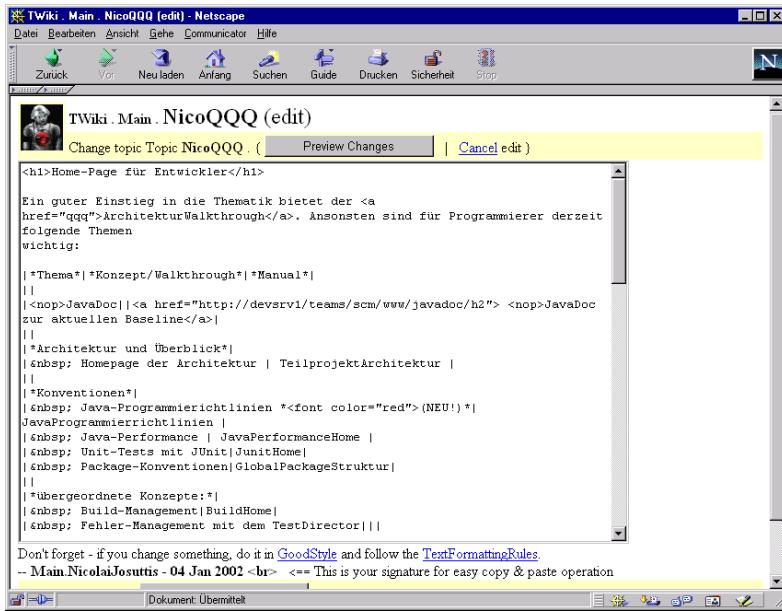
The screenshot shows a Netscape browser window with the title "TWiki, Main, NicoQQQ - Netscape". The address bar displays "Main . { Home | Users | Offices | Changes | Index | Search | Go [] }". The main content area is titled "Home-Page für Entwickler". It contains a brief introduction: "Ein guter Einstieg in die Thematik bietet der [ArchitekturWalkthrough](#). Ansonsten sind für Programmierer derzeit folgende Themen wichtig". Below this, there is a table with various links categorized under different sections like "Thema", "Architektur und Überblick", "Konventionen", and "übergeordnete Konzepte".

Thema	Konzept/Walkthrough	Manual
JavaDoc		JavaDoc zur aktuellen Baseline
Architektur und Überblick		
Homepage der Architektur	TeleprojektArchitektur	
Konventionen		
Java-Programmierrichtlinien (INFLU)	JavaProgrammierrichtlinien	
Java-Performance	JavaPerformanceHome	
Unit-Tests mit JUnit	JUnitHome	
Package-Konventionen	GlobalPackageStruktur	
übergeordnete Konzepte:		
Build-Management	BuildHome	
Fehler-Management mit dem TestDirector		
Vorgehensmodell	VorgehensModell	

Dokument: Übermittel

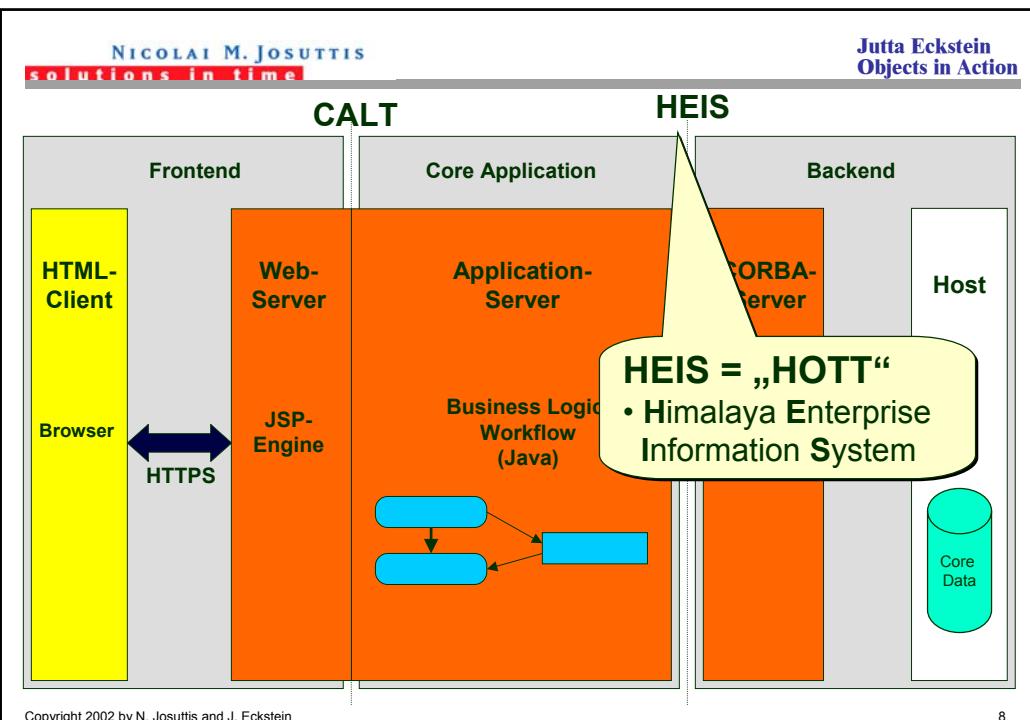
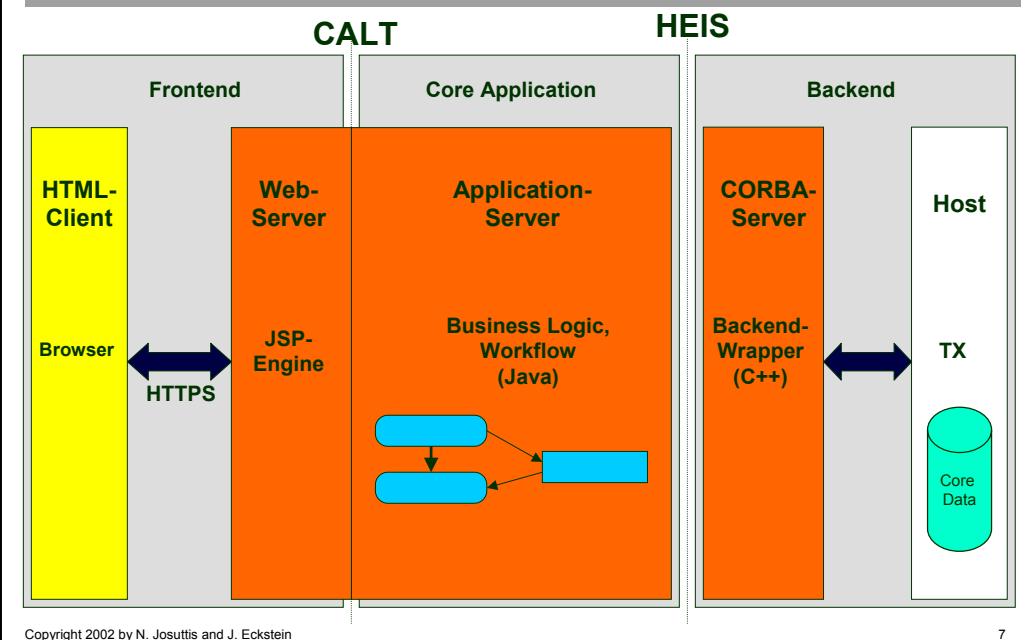
Copyright 2002 by N. Josuttis and J. Eckstein

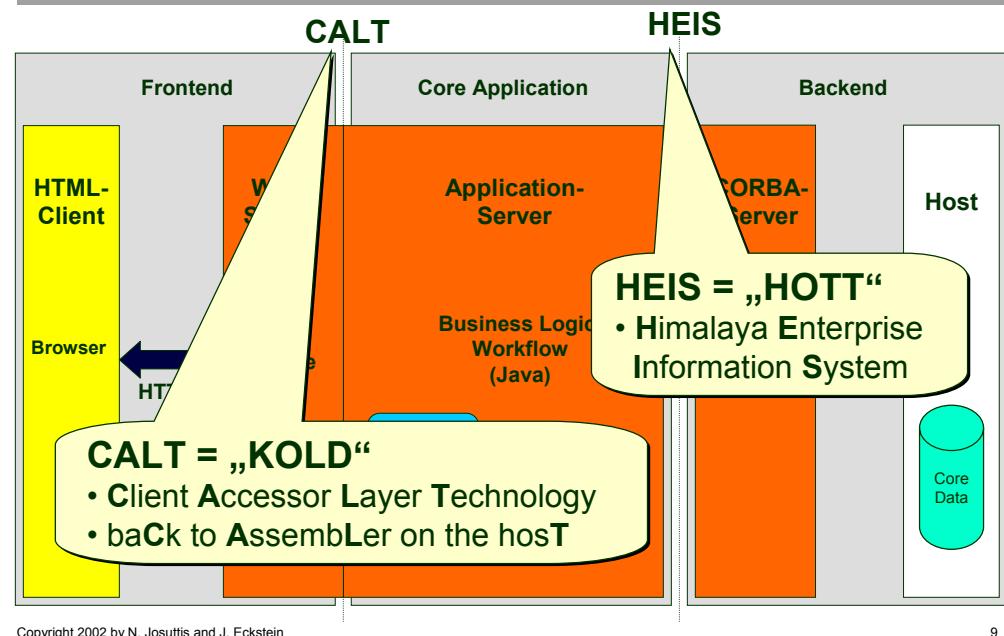
4



Force Fun

- **Regeneration**
 - force vacation
 - avoid overtime
- **Satisfaction**
 - results
 - customer feedback
- **Contests**





Force Simplicity

- **KISS**

- **„I built a lot of large systems, but I never built a complex system“**
[Kerth, Meszaros, Doble]

- **„Start stupid and evolve“**
[Kent Beck]

- **because simple systems**

- are better maintainable
- don't require smart programmers

Force Discipline

- **Guidelines**
 - good practices (essence of experience)
 - „no code without test code“
- **Deprecated interfaces**
 - and getting rid of them
- **Reviews and Code Inspection**
 - external
 - internal (by developers and scripts)
 - active marketing of concepts and interfaces
- **Timeboxing**

Testing

- **Unit-Tests**
 - there should/must be more test code than code
 - for each individual class, component, module, ...
- **JUnit**
 - junit.org
- **Note:**
 - Special support for tests necessary
 - dummy contexts
 - GUI tests are more difficult
 - Separation of presentation and core application supports easy tests of business logic

Timebox

Fixed short period for an internal release cycle

- **three weeks**
- **at the end:**
 - result (success or reasons for failure)
 - no overtime
 - retrospective
- **starts with planning session of teams**
 - according to functionality and business value
- **each week:**
 - integration and feedback

Force Courage

- **Errors are Part of the Process**
- **“Do It Right the First Time” sends the wrong message**
 - we can't be uncertain
 - we can't experiment
 - we can't learn from mistakes
 - we can't deviate from plan

⇒ **“Don't worry about getting it right the first time,
get it right the last time.”**
- **Change people and roles**
 - many projects fail due to people in wrong positions
- **Honesty**
 - bad news are good news

Force Communication

- **E-Mail and Meetings**
- **Plenary assembly and info meetings**
- **Parties and Food**
- **Wiki-Web**
- **„Floating desks“**
- **Avoid Bottlenecks**

Avoid Bottlenecks

- **Avoid technical bottlenecks**
 - avoid global resources
 - avoid singletons
 - avoid strong typing
- **Avoid communication bottlenecks**
 - modules/components instead of one BOM (business object model)
 - stable interfaces
 - clear roles
 - pragmatic instead of bureaucratic
- **Note: Project and team size influence architecture**

Therefore...

- Quality for Success means:
 - Transparency
 - Fun
 - Simplicity
 - Discipline
 - Courage
 - Communication

Quality in the E-Business

“You can rant and rave all you want about software quality (or lack thereof), but the marketing guys run the world and they want market share now... period, end of discussion.

My job is to deliver on time on budget, with the ‘appropriate’ quality metrics.”

[Telco Development Manager]

⇒ There is no explicit time for quality anymore

⇒ Make quality an integrated element of the process

Many Thanks!



Nicolai M. Josuttis
solutions@josuttis.com
www.josuttis.com



Jutta Eckstein
jutta@jeckstein.com
www.jeckstein.com