



# Web Engineering

## Quality Management

Prof. Dr. Alexander Knapp

Dr. Nora Koch

SS 07 (9) – 9.07.07

Ludwig-Maximilians-Universität München

# Software Quality and Web Quality

---

- Definition of **Quality** in Standard ISO/IEC 9126
  - The totality of features and characteristics of a software product that bears on its ability to satisfy stated or implies needs.*
  
- Quality is a complex multidimensional concept that cannot be measured directly
- A model and a process is required to measure quality of software products
- User have to be considered in the evaluation of quality
  
- Definition, models, evaluation process and criteria can be applied to Web software
- **Web Quality** focus on some Web (software) critical aspects
  - suitability
  - performance
  - security
  - legal compliance
    - standards, conventions, legal norms, intellectual property rights

# Quality Dimensions

---

- Quality characteristics
  - functionality
  - reliability
  - efficiency
  - usability
  - portability
  - maintainability
- Objects
  - content, structure, presentation
  - infrastructure and environment
  - functions
- Phases
  - requirements
  - design and implementation
  - acceptance and installation
  - operation and maintenance

# Quality Characteristics and Methods

Functionality	Functions	Content, structure and presentation	Infrastructure and environment
<b>Suitability</b>	reviews, inspections, test-driven development	checklists, lexical testing, style guides, reviews	
Accuracy	capture/replay, test-driven development	static analysis, link testing, lexical testing, reviews	static analysis, link testing
Interoperability	cross-browser and cross-platform compatibility testing	test printing, checklists, reviews, compatibility testing	cross-browser and cross-platform compatibility testing
Compliance	compatibility testing, style guides, test-driven development	checklists, compatibility testing, style guides, reviews	cross-browser and cross-platform compatibility testing
<b>Security</b>	analysis of common attacks, reviews, inspections		analysis of common attacks, forced error testing, ethical hacking

# Reliability and Efficiency

---

Reliability	Functions	Content, structure and presentation	Infrastructure and environment
Maturity	endurance testing		endurance testing
Fault Tolerance	forced-error testing, stress testing		forced-error testing
Recoverability	forced-error testing, failover testing		failover testing, forced-error testing, low-resource testing

Efficiency	Functions	Content, structure and presentation	Infrastructure and environment
Performance <small>(time behaviour)</small>	load and stress testing		load and stress testing, monitoring
Resource Utilisation	endurance testing	load testing	endurance testing, monitoring

<b>Usability</b>	<b>Functions</b>	<b>Content and Structure</b>	<b>Infrastructure and environment</b>
Understandability	usability studies, heuristic evaluation	static readability analysis, usability studies	
Learnability	usability studies, heuristic evaluation		
Operability	usability studies, heuristic evaluation		heuristic evaluation
Attractiveness		publicity testing	

# Performance: Basic Measures

---

- Throughput
  - number of requests completed per unit of time
- Response time
  - total time elapsed between submitting a request and end of the response
- Utilisation
  - fraction of time during the system was busy
  
- Negative examples (bad performance)
  - people tend to leave a Web site if the response time is above 8 seconds (eight second rule)
  - holidays imply high workload (# of requests by the user of the software) making a system slow
  
- Performance is a key success / failure factor

# Performance Evaluation Example

---

- Dynamic request service
  - Architecture: Client, Internet, Web system
  - Web System: HTTP interface, application logic, information source
- Architectures for generating dynamic content
  - scripting languages PHP
  - component-based technologies: Java servlets and EJBs
- Benchmark: 2 applications
  - online bookstore stresses the server back-end
  - auction site stresses the server front-end

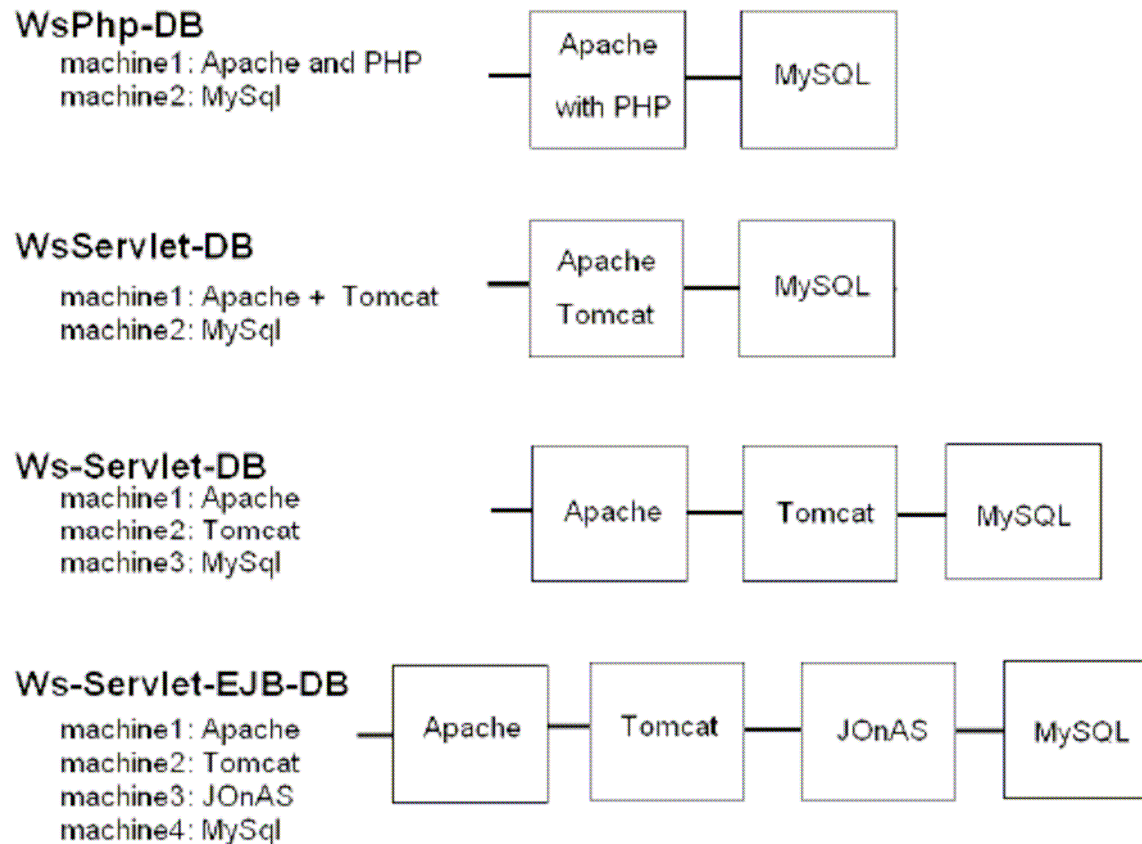
# Evaluation Techniques: Step-by-Step

---

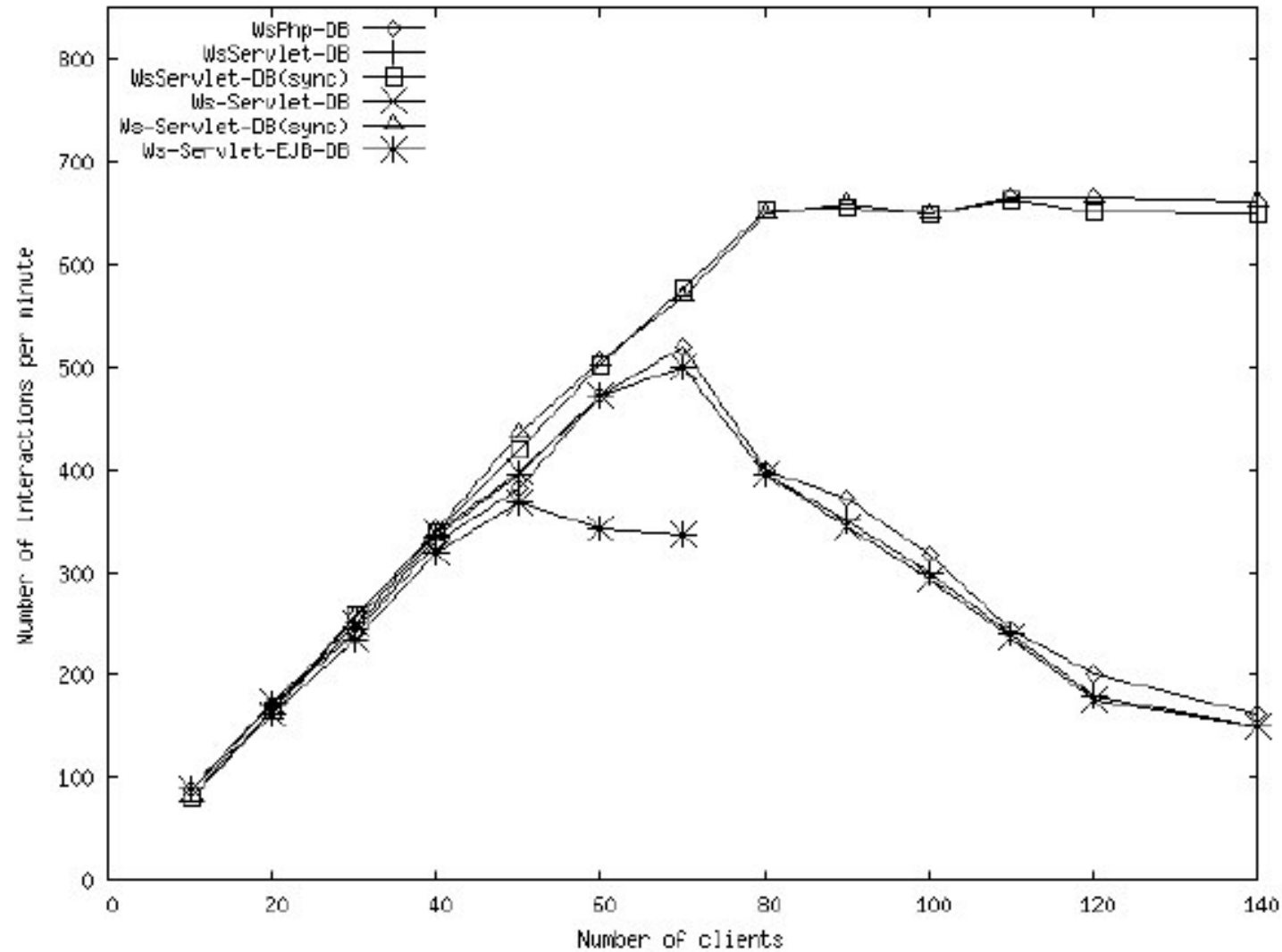
- Define of objectives and performance indices
  - comparison of technologies
  - interactions per minute and number of clients
- Select measurement technique
  - simulation
- Select type of test workload
  - Define behavioural characteristics
    - number and type of transactions (#bids)
    - database queries
  - Define quantitative characteristics
    - size of the system (# products, # auctions, # users, etc)
    - time intervals
- Evaluate the system
  - define the architecture
  - run the simulation
- Reduce and visualize data
  - collect data and produce graphics
- Validate and interpret results
  - conclusions of best technology
- Iterate (optional)

# Performance Evaluation Techniques

- Different architectures



# Performance Evaluation Example



# Performance Evaluation Mistakes & Performance Tuning

---

- Evaluation mistakes
  - Wrong evaluation technique (not applicable, high cost, ...)
  - Unrepresentative workload
  - Overlook of parameters
  
- Methods to improve performance
  - identify bottlenecks
  - classify bottlenecks in primary and secondary
  - identify measure to reduce bottlenecks
  - evaluate costs of improvements
  
- Performance tuning
  - tuning web application
  - configuring Web server software
  - increasing computing power of Web server hardware
  - increasing network bandwidth
  - caching and replication

# References

---

- Gerti Kappel, Birgit Pröll, Siegfried Reich, Werner Retschitzegger (eds.). *Web Engineering*. John Wiley & Sons, Chichester, 2006
- Rick Hower. *Web Site Test Tools and Site Management Tools*. 2007.  
<http://www.softwareqatest.com/qatweb1.html>
- Emmanuel Cecchet, Anupam Chanda, Sameh Elnikety, Julie Marguerite & Willy Zwaenetpoel: *Performance Comparison of Middleware Architectures for Generating Dynamic Web Content*, ACM, 2003