Part VII

Evolution...

continuous progress

Development  Evolution  Operation

© 1997-2005 Dr. Martin Gaedke

Chapter://1

Testing and Monitoring

Testing is extremely difficult
- Testing is a continuous process
- Starts during Assessment Phase
- Address Testing seriously! Prepare for Test Plans

Early:
- Requirements for testing
- Criteria for non-functional requirements – How to test for “good”?

Later:
- Define Test Cases
- Unit Tests etc.

Final:
- Release Test Criteria (e.g. ZBB, Customer Feedback, etc.)
- Based on Criteria an Internal Release becomes a Release
Testing Problems
- Things to look at – and define Test Plans for:
  - Spelling errors, broken links, buggy scripts
  - User and eCommerce night mares: overcharging accounts
  - Assumption of Correctness "site is correct because it looks and loads right"-Syndrome
  - The scenario: Browser Types x PlugIns x Script Engines x OS x Hardware x Network Connections
- BTW, Job of Testing – a social problem…
  - Testing is a difficult job and in many cases thankless
  - Proper testing is often not understood or appreciated and often seen as a boring task

Test Plans and Procedures
- Prepare Test Plans
  - Functionality Testing
  - Content Testing
  - User Testing
  - Security Testing
- If applicable try to use Test Labs
- Procedures for finding issues/problems
  - Report Problem Tracking System
  - Track, Handle, Finalize: Initiate Change Request
  - Be integral part of Configuration Management

Functionality Testing
- Site functions properly and meets specification
- Main Testing: Units, integration (all units together), browser
- Final Testing: User’s system configuration (e.g. speed of hard-drives, Java runtime with different processor speeds), delivery (network and server aspects)
- Test Labs may help in some cases, especially for final testing

Content Testing
- Content of site is correctly implemented
- Consider proof-reading, especially spelling of names and companies
- Check for copyright inclusion and legal disclaimers
- Check images and other media type (includes consideration of user’s system configuration)

User Testing
- Site meets user’s needs and is usable
- If available testing rooms
- Low-cost testing with some people and questionnaires
- Online-testing with feedback option ("send us your comments and you may win…")

Security Testing
- Should not be part of functionality testing
- Handle explicitly
- Include application-, server-, network-, physical site-security, and physical access by Staff; as well as many other issues
Handling Results Of Testing

- Use a Problem Tracking System
  - **Track**: Problem (id, problem description, discovered by, when, user’s system configuration, severity,…)
  - **Handle**: ProblemId, HandledBy, Status, …
  - **Finalize**: Initiate Change Request (CR)
- Should be part of a change request processing approach, cf. Requirements Engineering

Bobby

Web Application Test Tools

Evolution – Ongoing Process

Evolution: Plan For Change

© 1999-2006 by Dr.-Ing. M. Gaedke
Discovering Web Services

Universal Description, Discovery, and Integration (UDDI) – Specifies what the API for a Web-based Registry looks like.
- All about the “Yellow, White & Green Pages”
- Defines how to run and operate Registry Sites on the Web
- Defines how to pay for its Operation – encourages basic lookup services for free
- Further Information at http://uddi.org

Why a DNS-like Model?
- Enforces cross-platform compatibility across competitor platforms
- Demonstration of trust and openness
- Avoids tacit endorsement of any one vendor’s platform
- May migrate to a third party

Registry Operation

- Peer nodes (websites)
- Companies register with any node
- Registrations replicated on a daily basis
- Complete set of “registered” records available at all nodes
- Common set of SOAP APIs supported by all nodes
- Compliance enforced by business contract

UBR: UDDI Business Registry

- UDDI von Microsoft (http://uddi.microsoft.com) und IBM (http://uddi.ibm.com)
UDDI provides information...

- Who – Business Information
- What – Find the right Type of Business
- Where – To Access a Service
- How – Describes how a given Interface functions

Information provided at http://uddi.microsoft.com

Part VII ► Chapter://2 ► Operation and Maintenance: UDDI

UDDI – A Publisher View

UDDI and Web Services

UDDI and SOAP

Registry APIs (SOAP)

Service Type Registration

© 1999-2006 by Dr.-Ing. M. Gaedke
Introduction

Activities after site development:
- Promotion
- Maintenance
- This may imply new development cycles: moving through all stages again
- Promotion and Maintenance ...
- Process vs. Product View
  - Process-based: Life Cycle of a Web-Application
  - Product-based: Evolution of a Web-Application

Promotion

Question: How to find the Web application?
- Promotion for intranets and extranets
  - Do promote internal Web-Applications!
  - Pre-set browser home, add link on the internal company's homepage, email, trainings, ...
- Promotion for public Web applications
  - Cf. marketing approaches for a product
  - E.g. advertisements in print media, business cards, cross-link with existing customers, etc. → Careful due to access behavior
  - Register with search engines and public portals
Promotion: Search Engines

- People get better in using search engines
  - Term of the year 2002: “Google it”
  - Cf. IEEE Technical Speaking

- Search Engines
  - Content is not understandable by Machines

- Facilitate Process by Providing Meta-Information
  - In HTML: META-Tag
    - `<META NAME="description" CONTENT="WebE Homepage">`
    - `<META NAME="keywords" CONTENT="WebE, Vorlesung">`
  - And there are more secrets… to influence search engine algorithms (Goal: become a top-5 entry)

Promotion: Search Engines II

- Check Referer header of an HTTP-Request
- Log-File Analysis for http://www.webengineering.org

What Is High Availability?

- How much downtime can my organization afford without loosing productivity, profits, sales, etc.?
- The solution to High Availability is a combination of people, process, AND technology
  - Beware of 99.99% myth - The nines model does not take timing into account

Availability:

\[
A = \left(\frac{MTBF}{MTBF + MTTR}\right) \times 100
\]

- \(MTBF\): Mean Time Between Failure
- \(MTTR\): Mean Time To Recovery

Simple Example:

- 24/7 Web Site with two failure a week and each requires 1 hour
- On a year’s time:
  \[
  \left(\frac{52 \times 7 \times 24}{2}\right) \times \frac{100}{99.41%} = 99.41\%
  \]

Achieve High Availability?

- It’s deceptively simple …
  - Plan and prepare
- Key to high availability
  - Deploy systems to create redundancy – the key from a technology standpoint, e.g. replicate Web server application logic (scale out, DNS-round robin, Network Load Balancer)
  - Define processes for people to solve conflicts
  - Test, test, test
  - Monitor on a continuous basis

Technical Approach: NLB

- NLB = Network Load Balancing
  - E.g. NLB-Service or Windows Load Balancing Service (WLBS)
- Generally used for scalability
- Can be used with databases
  - Front end switch for log shipping role change
  - Warm standby server
  - Protect analysis services
Maintenance

- Web applications are like "living entities"
  - Like a Garden: Must be maintained to look nice.
- Maintenance is any event that yields to a new iteration of the life cycle (of a feature or the application as a whole)
  - Often you will find: Maintenance – Any development activity performed to modify or fix the Web application after it has been completed or reached some final milestone. Be aware of ad-hoc maintenance (code-and-fix approach)!
- Reasons for Maintenance
  - CR: Content, delivery / access, functionality
  - Maintenance is good – it is the beginning of evolution!

Aspects of Maintenance - I

Content Maintenance (Main Activities)
- Note: Never work on a live site! Use a staging approach
- E.g. Content stored in Database and is easily manipulated using dedicated Editors. Reviewed Content is updated on Production Server

Delivery Maintenance (Success Disaster)
- "Perfect" Web application fails – if traffic increases dramatically
- Prepare for scalability and availability

Aspects of Maintenance - II

Functionality Maintenance (Crisis Management)
- Corrective – Activities to fix application bugs and design flaws
- Adaptive – Activities to make application work for a "Problem-Browser" configuration
- Perfective – Activities to increase functionality (feature additions)
- Note: If too many Bugs hinder the functionality and can not be solved in a few minutes:
  - Provide a currently under maintenance page

Tools for Maintenance

- Monitoring
  - Web application statistics terminology
  - Logs of server, router, etc.
- Use: Log Analysis Software
- Your own unit tests
- Feedback Channels
  - E.g. contact information, forms for user feedback etc.
Further Readings

Literature

- Chapter 8: Thomas A. Powell, *Web Site Engineering*, Prentice Hall PTR

Further information available at [Lecture Web Site](http://example.com/lectureweb)

Important Links

- IBM's Ease of Use
- Jakob Nielsen's web usability website
- Microsoft Usability Home Page
- Sun's Usability Testing of Web Concepts:
- User Interface Engineering
  - [http://world.std.com/~uieweb](http://world.std.com/~uieweb)