DELIVERABLE

| | |
|---|---|
| Project Acronym: | **Natural Europe** |
| Grant Agreement number: | **250579** |
| Project Title: | **Natural Europe: Natural History & Environmental Cultural Heritage in European Digital Libraries for Education** |

# D6.4.1 – Technical Validation Plan

**Revision: [final]**

**Authors:**

**Martin Wolpers (FIT)**

**Kerstin Schmidt (FIT)**

| | | |
|---|---|---|
| **Project co-funded by the European Commission within the ICT Policy Support Programme** | | |
| **Dissemination Level** | | |
| **P** | **Public** | **X** |
| **C** | **Confidential, only for members of the consortium and the Commission Services** | |

## Revision history:

| Revision | Date | Author | Organization | Description |
|---|---|---|---|---|
| 0.1 | 4/10/11 | M. Wolpers | FIT | First draft ToC |
| 0.2 | 20/10/11 | M. Wolpers | FIT | Final draft v1 including all partner feedback on very first version |
| 0.5 | 14/11/11 | M. Wolpers | FIT | Final daft v2 including all partner feedback on v1 |
| 1.0 | 16/11/11 | M. Wolpers | FIT | Final draft including all partner feedback on v2 |

## Statement of originality:

## Executive Summary

The present deliverable D6.4.1 presents the approach for the technical validation that will be taken by the NaturalEurope consortium. Based on review of appropriate literature, a number of criteria were identified that need to be addressed by technical validation. Through the study of the emerging NaturalEurope infrastructure, the respective interfaces are identified which need to validated technically to ensure correct functioning of the services and tools provided through NaturalEurope. Combining the knowledge of the infrastructure with approaches to technical validation, the respective testplan is deviced. Furthermore, measures to ensure a continuous high quality of the software provided as well as the availability of the NaturalEurope infrastructure, services and tools, are specified and respective usage processes described.

# Table of contents

# 1 Introduction

## 1.1 Scope

This deliverable presents the technical validation plan for the technical infrastructure of the NaturalEurope system. This presents a change from the original approach of carrying out technical validations before the end of month 12. The project plan did foresee implementations to be in a verifyable state only at month 15 and moth 18 respectively. Hence, the follow-up deliverable D6.4.2 Second Periodical Technical Validation Report will include the results of the ongoing validation tests once stable releases of the technical infrastructure become available. In this deliverable, we describe what methods, concepts and approaches we are going to use for evaluation and validation of service components in the middle and backend layers.

The Deliverables D4.1 and D4.2 "Specification of the NE Platform & Tools" provide the basis for developing the technical validation plan. Furthermore, we draw on the Joint Deliverable 6 "Evaluation of functional prototype for metadata tools and concepts" of the MACE project to build on the experiences made in the MACE project.

## 1.2 Audience

The deliverable outlines how we plan to validate the technical infrastructure of the NaturalEurope system. The outcomes will be transferred by WP2 Requirements Transfer & Analysis into the WPs WP3 Educational Design and WP4 Platform Design & Integration. Hence, the intended audience are the developers of the NaturalEurope technical infrastructure as well as the project manager and coordinator that use the found results for project-advancement and planning purposes. The wider public has access to this deliverable in order to provide insights into lessons learned.

## 1.3 Definitions

**CHO**: cultural heritage objects

**NHM**: natural history museum

**ESE/EDM**: metadata schemas used by Europeana

**CRUD**: Create/Retrieve/Update/Delete

## 1.4 Structure

**Chapter 1:** contains an overview of this document, providing its Scope, Audience, and Structure.

**Chapter 2:** describes the background knowledge relevant for technical validation

**Chapter 3:** provides an overview of possible validation methods applicable in the context of NaturalEurope.

**Chapter 4:** describes the various service tests

**Chapter 5:** describes the architectural elements of the NaturalEurope infrastructure that will be subject to tests

**Chapter 6:** describes the test and validation plan

**Chapter 7:** briefly discusses the impact on the NaturalEurope vision

**Chapter 8:** gives the references

# 2 Approaches to technical validation and evaluation

This chapter describes the methods and approaches we are using for assessing the quality of the work we did so far for the technical infrastructure. As the NaturalEurope system consists of several layers, we need different assessment methods for each layer: frontend services, middleware services and backend components. Frontend services will be dealt with in D6.1 FF of WP6. Here, we discuss the validation of middleware services and backend components. In addition, we need to assess why certain concepts were chosen and if they proved to be usable for the issues encountered.

Currently, common agreement on the term of "quality" [6] has not yet fully been achieved. In habitual language use, "quality" denotes the properties and goodness of a thing [4]. Quality is the entirety of attributes of a product regarding its ability to fulfil prescribed requirements (ISO 8402:1995). The approach of measuring quality in terms of fulfilment of expectations is emphasised in the ISO 9000 series standards.

In literature, five different views on quality are discussed:

- The transcendental approach sees quality in line with the philosophical principle of "beauty", which cannot be measured, but confesses to the experienced. It is a symbol of uncompromisingly high standards.
- The product-based approach considers exactly measurable quality variables. A product's quality is the sum of the quality of its ingredients.
- The definition of quality in the user-oriented approach is coined by the subjective experience of usefulness that buyers of the product have.
- The manufacturing approach considers variation in the production process bad quality and is comparable to the ISO 9000 view.
- In the value-based approach something is of high quality if the price of it is acceptable to both consumer and producer.

In the scope of NaturalEurope, we will use the product-based approach for services and backend components. We will use two methods for assessing the quality of our work, evaluation and validation. The middle and backend layer will be evaluated by experts and developers. Validation is more technically oriented and is based on quantifiable results. To validate e.g. a web service, we will define criteria and threshold values, then run tests and record results to compare these against the thresholds. This helps us to assess whether a service component works as intended or is compliant to a standard.

# 3 Validation methods of services and backend components

Since middle layer services and backend components are not directly accessible by end users, the traditional user evaluation method will not work in these cases. Instead we decided to validate the software quality of middle layer and backend components with software quality attributes as they are defined in [3].

The software quality attributes used are benchmarks describing the intended behaviour of a system within the environment for which it was built. We will use common quality attributes providing the means for measuring the fitness and suitability of a system to evaluate components in the MACE system. Below, some relevant quality attribute are discussed.

## *3.1 Performance*

Performance is concerned with how long it takes the system to respond when an event occurs. It can be measured by events per timeframe, for example technically successful metadata record retrievals per minute.

## *3.2 Reliability*

The amount of time for the system being up and running correctly; the amount of time between failures and the length of time needed to resume operation after a failure.

Reliability, within the perspective of application architectures, can be viewed as the degree to which a given system architecture is susceptible to failure at the system level in the presence of partial failures within components, connectors, or data.

We can assess performance and reliability by using automatically generated system logfiles, where events and errors are stored. To improve performance and reliability, we can filter logfiles for bottlenecks and most often occurring errors – by fixing these most problematic points first instead of working on small issues, we will improve the overall system best.

## *3.3 Interoperability*

The ability of two or more systems or components to exchange information and to use the information that has been exchanged [7]. Means of achievement: Simplicity of the components, their interaction mechanisms and protocols. Clean partitioning of responsibilities. Component interfaces have to be specified clearly and completely.

The NaturalEurope architecture is a modular service-oriented architecture built on the principle that interoperability and extensibility is best achieved by the integration of different interfaces as clearly defined modules. These interfaces interoperate based on a formal definition that is independent of the underlying platform. This definition hides the implementation of a language-specific service.

### *3.4 Modifiability*

Modifiability means the ease with which a software system can accommodate changes to its software. It can be further broken down into extensibility, configurability, and reusability, as described below.

### *3.5 Extensibility*

Extensibility is defined as the ability to add functionality to a system. Dynamic extensibility implies that functionality can be added to a deployed system without impacting the rest of the system.

### *3.6 Configurability*

Configurability is related to both extensibility and reusability in that it refers to post-deployment modification of components, or configurations of components, such that they are capable of using a new service or data element type.

### *3.7 Reusability*

Reusability is a property of a given system architecture if its components, connectors, or data elements can be reused, without modification, in other applications. The primary mechanisms for inducing reusability within architectures are reduction of coupling (knowledge of identity) between components and constraining the generality of component interfaces.

### *3.8 Portability*

Definition: the ability of a system to run under different computing environments. The environment types can be either hardware or software, but is usually a combination of the two.

### *3.9 Testability*

Software testability refers to the ease with which software can be made to demonstrate its faults through testing, typically execution based. For a system to be properly testable, it must be possible to control each components internal state and inputs and to observe its outputs. Frequently, this is done through use of a test harness, specialised software designed to exercise the software under test.

## 3.10 Scalability

Scalability refers to the ability of the architecture to support large numbers of components, or interactions among components, within an active configuration. Scalability can be improved by simplifying components, by distributing services across many components (decentralizing the interactions), and by controlling interactions and configurations as a result of monitoring.

## 3.11 Conclusion

First, we need to establish a baseline for the given parameters. This baseline will be comprised of the results of test first run, or if applicable, of the data available at the beginning of the project. Thereby, we will be able to show how each parameter advances (if needed) during the project.

# 4 Service tests

The NaturalEurope services enable front applications to create, query and manipulate (meta-)data that is managed within the NaturalEurope infrastructure. This infrastructure exposes its functionality through the services that are described within this section. At a general level we will check how these services perform against the following software quality attributes:

- Reliability: We will deploy a tool to measure the availability of these services. That is, at regular timestamps, the tool will check whether the service is available or not. The output of this tool will be used to generate statistics on the availability of these services.
- Performance: The service will be tested for performance by using stress test tools.
- Reliability will be measured by sending SOAP requests with invalid parameters, insufficient values or out-of-range values. To succeed, services must not crash when confronted with illegal values and resume normal operation afterwards.
- Redundancy is another critical point for several services others depend on0. We will critically assess the service and try to handle non-availability by artificially “crashing” services in a test environment and finding out how well this is handled by the whole NaturalEurope infrastructure.
- Portability: Most services were written in Java and use the Apache Axis container so they can be redeployed to another machine quickly in case of failure. We will check how well this works in practice.
- Interoperability: The NaturalEurope services use SOAP and should be WS-i compliant, which means they will be interoperable across a variety of other services and platforms. WS-i tests will be run regularly to ensure compliance with the WS-i standard (Web service interoperability, http://www.ws-i.org/).
- Testability: The web service code will be tested continuously with unit tests. These tests ensure that the service works as expected, reacts correctly to unexpected or missing input data, and provides continuity and constant functionality over changing software versions; i.e. new versions will deliver the same results.

## *4.1 Testing levels*

Testing is typically considered to take place on four different levels [10]:

1) Unit testing
2) Integration testing
3) System testing

4) Validation

From these four levels of testing, the unit testing is done by the developers themselves. Units are complete software modules like the MM authoring tool or the various types of repository implementations. To validate the correctness of the service implementation, unit tests are conducted for which for example the Junit toolkit is available. These tests are not reported here in detail as most of them are carried out automatically. Rather, we will report on the general availability and error-freeness of the implementations. Furthermore, “validation “ item 4 on above list refers to user validation and is reported in D6.1 and updates thereof.

Integration testing and system testing relate to the incorporation of new units into the framework. This is the main focus of the technical validation because here the various interfaces of the different units must be attuned. Errors at this level will clearly hinder the NaturalEurope system from correct function, either in part or as a whole.

The architecture of NaturalEurope bases on a service oriented approach. Hence, all interfaces will be represented through web service technology. We will rely on web service performance testing to determine the scalability and robustness of the web services. Most web services are implemented according to the Service-Oriented Architecture (SOA) paradigm. Testing here implies calling target web services with varying messages across a range of concurrent loading clients, taking into account the respect implementation methods, e.g. SOAP, HTTP (put/get in XML via URL) or SRU/SRW (Search Retrieve through URL).

Performance testing will be done using test software that creates multiple simultaneous processes that perform web service calls and then measure the time to receive the result. The result will tell basically how well a specific service copes with increasing load and at which point it does not fulfil performance targets.

Making complete tests for all methods on all services is a very major task so we choose to performance test those functions we inherently know might have performance problems since they contain more advanced functionality.

Performance testing of the complete system will be made in cooperation with the respective testbeds in order to be able to identify the respective problems that might occur at each of the testbeds.

Interoperability between services is the cornerstone of the NaturalEurope architecture. At each NHM site, the MM authoring tool must play together with the CHO and Educational Pathway repositories, while these repositories must interoperate with the central repositories at the same time.

The interoperability tests to be made in the NaturalEurope technical infrastructure can be divided into two main groups:

1. Testing done on WSDL files between services during development and integration.
2. More formal testing of services using WS-I tools. These test both design time interoperability (based on a WSDL file) and run-time interoperability (whether the web services responds according to WS-I at run-time). If deemed necessary, a manual test suite will be developed to deal with customized service implementations. The test suite will be integrated into the CruiseControl mentioned below.

The Web Services Interoperability Organization (WS-I) has developed testing tools that evaluate Web services conformance to profiles. These tools test web service implementations using a nonintrusive, black box approach. The tools focus is on the interaction between a web service and user applications, e.g. between the MM authoring tool and the local CHO repository. To test web services from the outside by simulating a client, thereby sending a series of web service requests to the service and validating the results sent back from the service against some pre-recorded values. If service results and pre-recorded values match, the service is assumed to work correctly. A tool capable of doing web service testing for SOAP-based services is soapUI [15].

The assumption is that tests have to be performed manually in which case the time to adequately test a complex application increases during the course of NaturalEurope. To remedy this situation the concept of “Continuous Integration” was invented by Fowler [11] where team members integrate their work frequently and this is supported by an automated build to check an error-free integration run. A tool to support that process is called “CruiseControl” [12]. CruiseControl monitors a source code repository for changes and as soon as it detects new or updated source code files it will automatically issue a build and run the associated tests. It can inform the developer who has committed changes automatically of a broken build. This short time span between commit and integration checking allows the developer to find the error quicker because he still remembers that changes he has made to the code.

The consortium agreed to use a SVN code repository for all software developed in the project. On top of the SNV repository [14], the CruiseControl will operate.

The bug and issue tracker like the JIRA tool [13] is being used to manage errors, bugs and their correction. In order to feed the bug tracker with input from end users, a help desk will be installed that answers to and manages issues of end users and translates them into the bug tracker. The bug tracker then maintains the list of open bugs and issues and who is resolving them. It will be the responsibility of the NaturalEurope helpdesk (http://www.natural-europe.eu/en/helpdesk) to ensure the provision of an answer to each submitted issue or bug.

# 5 NaturalEurope architecture

The architecture of NaturalEurope is described in more detail in D4.2 “Specification of Natural Europe Platform & Tools”. Here, we briefly summarize the main components that provide services and interfaces that need continuous monitoring and validation.

The NaturalEurope architecture bases on a federation of nodes. Each node exists at one NH museum and is used to create and store CHOs and related metadata as well as to store and create learning pathways. Respective services are implemented that provide access to these functionalities through interfaces. The functionality related to CHOs is captured in the Natural Europe NHM Cultural Environment (NECE), while all functionality related to pathways is contained in the Natural Europe NHM Learning Environment (NELE). The functionality to be tested will be described in the following two subsections.

## *5.1 Natural Europe NHM Cultural Environment (NECE)*

NECE refers to the set of modules and tools that will be deployed at each participating NHM in order to allow the complete metadata management life-cycle (at least for the contributed CHOs): i.e. ingestion, maintenance, curation, and dissemination of CHO metadata.

The **Multimedia authoring tool**, described in D4.2a, enables the creation, management, and administration of CHOs, CHO collections and CHO metadata. It offers the following functionality:

- Create/delete/manage CHO Collections
- Describe CHO Collections (with appropriate metadata)
- Import/create/delete/manage/export CHO Metadata records
- Import/publish/update/delete CHOs
- Create/delete/update/manage Users

The functions are split over the client side and the server side. At the client side, the tool runs within a web browser and provides, apart from GUI operations mainly business logic operations. The server side is represented by a web server that provides access to these services

- *CRUD Service:* Facilitates the creation, retrieval, update and deletion of a CHO, a CHO Metadata record, a CHO Collection, a user etc.
- *CHO Import Service:* Supports the importing of record-based xml metadata in order to be converted to SIPs using the SIP Transformation Module. The SIPs are then transferred to the Persistency Management Module for storage into the NHM CHO Repository.
- *Vocabulary Access Service:* Responsible for returning the taxonomic terms matching a specific term prefix.
- *Concurrency Service:* Provides the basic methods for acquiring/releasing/refreshing locks on a CHO Metadata Record or a CHO Collection.

The **local CHO repository** running at each NHM, as described in D4.2a, also runs at the server side. It provides the following functionality:

- CRUD services for CHO Collections
- CRUD services  CHO Metadata records
- Publishing/retrieving /deleting CHOs
- CRUD services for  Users
- Access to metadata via OAI-PNMH Interface

Finally, the **central NHM Metadata Repository** harvests all CHO metadata and provides this as central access node of the NaturalEurope Architecture to the Europeana services. All metadata exchanged via the publicly accessible interfaces is compliant with ESE/EDM.

## 5.2 The Natural Europe NHM Learning Environment (NELE)

The NELE is the set of modules and tools that will be deployed at each participating NHM in order to allow the authoring and consumption of educational pathways.

The Educational Pathway Authoring Tool, described in D4.2d, is used to create and maintain the educational pathways. It makes use of the cultural material search service that allows the retrieval of relevant content from the Natural Europe cultural federation and of Europeana's Open Search API.

Furthermore, NELE includes the Faceted Search Web Interface as described in D4.2f that also uses the cultural material search service. In addition, the Educational Pathway Navigation Web Interface as part of NELE provides a GUI for the EP Repository in order to discover, retrieve, render and present SCORM 2004 educational pathways.

The in-hall interactive installations will also use the search services of NaturalEurope, that are the Natural Europe cultural federation search API and the Europeana's Open Search API.

## 5.3 Specific architecture components

Based on the identified services, the following architecture components and their services need to be tested. The tests will be carried out in tight cooperation with the developers of each service. As only a few partners actually carry out the development of the architecture, the coordination will be done through simple meetings. Testing will occur after implementation of a service. CruiseControl will issue a notification that a software component has been replaced. If possible, automatic unit tests are run against this new component. If automatic tests are not possible or when errors are detected, the respective developer(s) will be notified to correct the issue(s).  In this case, the bug tracker will be updated; either directly automatically or manually through the help desk.

### 5.3.1 CHO and CHO metadata repositories

- CRUD API for the management of collections and objects

- OAI target service for exposing metadata to Europeana
- search API (materialising the Cultural Material Search service)

### 5.3.2 Educational Pathway repositories

- CRUD API for the management of collections and objects
- Batch import of media objects and metadata
- Harvesting OAI target service
- Search API used for querying repository content

### 5.3.3 Central CHO and CHO metadata repository including the bridge to Europeana

- CRUD API for the management of collections and objects
- OAI target service for exposing metadata to Europeana
- search API (materialising the Cultural Material Search service)
- Read: harvesting to Europeana

### 5.3.4 Central Educational Pathway repository

- Write: harvesting into repository
- Modify/write: existing pathways
- Read: Metadata for pathway authoring tool

### 5.3.5 Metadata validation services

- Natural Europe CHO validation service at the federated repository that will be activated by the user when he/she finishes the annotation process. This service should check the well-formedness and the validity of the record with respect to the ESE-CHO AP.
- Natural Europe CHO validation service at the federal repository that will be activated after the harvesting process.

# 6 Testplan

The Natural Europe Platform & Tools (D4.4) is due in month 15 of the project. Only afterward, extensive validation will take place. In order to ensure complete functionality, within a month after the release of the platform and tools, respective validation tests will take place. Found errors etc. will be fed back to WP4 for uptake, consideration and correction.

In month 18, the bridge to Europeana as described in D4.5 will be released. In order to ensure complete functionality, validation tests will take place within a month after the release of the bridge. Found errors etc. will be fed back to WP4 for uptake, consideration and correction.

During the remainder of the project, the correct function of the services will be tested continuously, e.g. through the CruiseControl and bug tracking systems. Results of these tests will be made available through a dashboard implementation.

| **Service providing implementations** | **Testing timetable** | **Validation Reports** |
|---|---|---|
| CHO and CHO metadata repositories | Extensive tests in M15-M16; afterwards continuous minute-based checks for uptime and checks for correct function and performance based upon redeployment schedule | M24, M36 |
| Educational Pathway repositories | Extensive tests in M15-M16; afterwards continuous minute-based checks for uptime and checks for correct function and performance based upon redeployment schedule | M24, M36 |
| Central CHO and CHO metadata repository | Extensive tests in M16-M17; afterwards continuous minute-based checks for uptime and checks for correct function and performance based upon redeployment schedule | M24, M36 |
| Central Educational Pathway repository | Extensive tests in M16-M17; afterwards continuous minute-based checks for uptime and checks for correct function and performance based upon redeployment schedule | M24, M36 |

| | | |
|---|---|---|
| Bridge to Europeana | Extensive tests in M18-M19; afterwards continuous minute-based checks for uptime and checks for correct function based upon redeployment | M24, M36 |
| Interactive Installation | Tests where required in M25-M26 | M36 |

**Table 6.1:** Testplan

In addition, all development partners have devised their tool and/or service related infrastructure and processes to ensure the correct operation of the provided software. In order to avoid inconsistencies and to ensure that users will have access to stable and tested version of the software, all developing partners use two environments i) a test and development environment (e.g. http://education.natural-europe.eu/natural_europe_test) where the development work and the tasting takes place and ii) a production environment (e.g. http://education.natural-europe.eu/) where the tested version of the software is deployed. These two environments use separate databases to ensure that data from the test environment cannot get into the database of the productive system. Data from the production database is copied to the test database frequently to be able to better reproduce errors that are reported by users.

The test environment is used for the development and testing of new functions of the services and tools provided by the respective partner. Based on the identified requirements and the corresponding use cases new functions are developed in the test environment. After the completion of the development task, groups of users test the new functions based on specific use case scenarios. After successful testing, the software is provided via SVN to the productive environment(s) of NaturalEurope . The deployment of new versions is performed in periods with low user activity i.e. weekend or during the night. After the deployment of the new version the users are informed about the new functionality.

When a problem is reported, the respective responsible partner is reproducing the problem in their test environment and identifies the solution. The solution is then tested again and, when finalized, provided via SVN to the productive environment. After the deployment of the new version, the users are informed about the new version of tool in which the problem is solved. Finally, the users test the provided solution and in case of errors/bugs they report them via the NaturalEurope helpdesk.

# 7 Impact on Natural Europe Vision

This section in each deliverable is dedicated to identifying its connections to the Vision of the Natural Europe project. The process in which the vision of the project is affected through the lifetime of the project is clearly documented in D2.1 “White Paper on Natural Europe Vision”. For each deliverable, its author answers the questions presented below, summarizing the outcomes in less than a page, which is then incorporated in the D2.1 document.

- How is this deliverable affecting the services that Natural Europe is deploying?
  - *How is it going to affect access to cultural content from museums of natural history?*
  - *How is it going to affect access to educational pathways coming from educators of the museums?*
  - *How is it going to affect searching for content, both educational and cultural, through the museum websites and interfaces set up by the project?*
  - *How is it going to affect setting up interactive installations in the museum exhibition floors?*
- How is this deliverable affecting the outreach of Natural Europe to its audiences?
  - *How is this deliverable going to extend the audiences to which Natural Europe is targeted?*
  - *How is this deliverable going to affect Natural Europe outreach to specific audiences (teachers, parents, pupils, etc)*

Obviously, the results of the validation ensure the correct function of technical infrastructure of NaturalEurope. As such, the impact on the NaturalEurope vision is limited to ensuring that the implemented architecture works as foreseen. Furthermore, the results provide indicators where urgent activities are required – might this be error correction or a redesign of parts of the architecture in order to accommodate the removal of errors and erroneous design decisions.

Looking at this from the point of the services that will be deployed, technical validation will surely help in collecting also input from the users of the tool that will lead to rethinking the envisaged services that the project can offer to its targeted audiences and thus refining them to serve the needs of the users in a more comprehensive way. In the same sense, technical validation can definitely provide some limited input to the categories of the envisaged audiences of the project as the results of the technical validation will allow for the creation/improvement of services and thus possibly to the inclusion of new audiences and stakeholders to the one that Natural Europe serves.

# 8 References

1. Joint Deliverable 6 "Evaluation of functional prototype for metadata tools and concepts", MACE Project (ECP 2005 EDU 038098), March 2008
2. http://portal.mace-project.eu/
3. (Bass, 2003) Len Bass, Paul Clements, Rick Kazman.: Software Architecture in Practice, Second Edition. Addison Wesley, April 11, 2003. ISBN 0-321-15495-9
4. (Brooks, 2003) Brooks, F. P.: Vom Mythos des Mann-Monats. MITP, 2003
5. (Fowler, 2007) M. Fowler, "Mocks Aren't Stubs", 2007, retrieved from http://www.martinfowler.com/articles/mocksArentStubs.html
6. (Geiger, 1988) Geiger, W.: Begriffe. In: MASING, Walter (ed.): Handbuch der Qualitätssicherung, Carl Hanser Verlag, 1988.
7. (IEEE, 1990) Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY: 1990.
8. (ISO 8402:1995) DIN EN ISO 8402:1995. Qualitätsmanagement. Beuth Verlag GmbH, 1986
9. (Meszaros, Gerard, 2007) xUnit Test Patterns: Refactoring Test Code, Addison-Wesley, 2007
10. B. Beizer (1990). Software Testing Techniques, International Thomson Press, second ed.
11. http://www.martinfowler.com/articles/continuousIntegration.html . Accessed 2010-10- 20
12. http://cruisecontrol.sourceforge.net/
13. http://www.atlassian.com/software/jira/
14. http://subversion.apache.org/
15. http://www.soapui.org