**BDD - A Practicable Approach for Computerised System Validation?**

**Bachelor Thesis 2020**

Client: wega Informatik AG  
Author: Sabrina Leuenberger  
Lecturer: Stephan Jüngling  
City, Date: Basel, 23th of July 2020

|  |  |
| --- | --- |
| **BDD – A Practicable Approach for Computerised System Validation?** |  |
| **Author** |  |
| Sabrina Leuenberger Merlachfeld 54 3280 Murten +41 (0)78 935 19 99 saleuenberger@gmx.ch |  |
| **Lecturer** |  |
| Stephan Jüngling University of Applied Sciences and Arts  Northwestern Switzerland  stephan.juengling@fhnw.ch |  |
| **Client** |  |
| wega Informatik AG Mathias Fuchs und Evelyne Daniel Aeschengraben 20 4051 Basel +41 (0)61 270 87 87 info@wega-it.com www.wega-it.com |  |
| Basel, July 2020 |  |

Declaration of Authenticity

I the undersigned declare that I have prepared the present paper independently and without the use of sources other than those indicated in the reference list.

All statements and information contained herein are listed and indicated as quotations and / or paraphrases.

This Bachelor Thesis / Project Work / Student Research has not been published to date. It has thus not been made available to other interested parties or examination boards.

City, Date

Signature

Preface or Background of the Project or Acknowledgment

Acknowledgements:

* wega Informatik AG for hosting the project, access to Teams, Confluence and Jira as collaboration tools
* Evelyne Daniel and Mathias Fuchs from wega for all the support and valuable inputs, their time, ….
* Andreas Hosbach from Zühlke AG for all is support from the technical side, ……
* Stephan Jüngling for supervising this project from FHNW,……

Management Summary / Abstract

Table of Contents

[Declaration of Authenticity II](#_Toc37082258)

[Preface or Background of the Project or Acknowledgment III](#_Toc37082259)

[Management Summary / Abstract IV](#_Toc37082260)

[Table of Contents V](#_Toc37082261)

[1 Introduction 1](#_Toc37082262)

[1.1 Initial Situation 1](#_Toc37082263)

[1.2 Test Optimisation using Automated Testing 1](#_Toc37082264)

[1.3 BDD High Level Test Automation for OQs 2](#_Toc37082265)

[1.4 Tools 2](#_Toc37082266)

[1.5 Hypothesis and Research Questions 2](#_Toc37082267)

[1.6 Scope 3](#_Toc37082268)

[1.6.1 In Scope 3](#_Toc37082269)

[1.6.2 Out of Scope 3](#_Toc37082270)

[1.7 Approach 4](#_Toc37082271)

[2 Materials & Methods 6](#_Toc37082272)

[2.1 Analysis 6](#_Toc37082273)

[2.2 Prototyping 6](#_Toc37082274)

[2.2.1 Tools Included in the Evaluation 6](#_Toc37082275)

[2.2.2 Tools Used Outside the Evaluation Part 6](#_Toc37082276)

[2.2.3 Used tools 6](#_Toc37082277)

[2.2.4 Method 7](#_Toc37082278)

[2.3 Evaluations 7](#_Toc37082279)

[2.3.1 Tool-Evaluation 7](#_Toc37082280)

[2.3.2 Audit of the Prototype 7](#_Toc37082281)

[3 Computerised System Validation according to GAMP5 8](#_Toc37082282)

[3.1 GAMP5: An Overview 8](#_Toc37082283)

[3.1.1 What is GAMP5 8](#_Toc37082284)

[3.1.2 Key Concepts 8](#_Toc37082285)

[3.1.3 Life Cycle Approach 8](#_Toc37082286)

[3.1.4 Product Categories 8](#_Toc37082287)

[3.1.5 Project Phase for Category 5 Software 8](#_Toc37082288)

[3.1.6 Automated Testing 8](#_Toc37082289)

[3.2 OQs According to GAMP5 9](#_Toc37082290)

[3.2.1 Writing OQs 9](#_Toc37082291)

[3.2.2 Incorporating the Risk Assessment 9](#_Toc37082292)

[3.2.3 Traceability 9](#_Toc37082293)

[3.2.4 Specification- and Test Management 9](#_Toc37082294)

[3.2.5 Validation Process in Respect to OQs 9](#_Toc37082295)

[4 Behaviour Driven Development 10](#_Toc37082296)

[4.1 BDD a suitable Software Engineering Approach for Highly Regulated Environments 10](#_Toc37082297)

[4.2 The Approach: An Overview 10](#_Toc37082298)

[4.3 Writing Executable Specifications with Gherkin 10](#_Toc37082299)

[4.4 Automation Tools 10](#_Toc37082300)

[5 OQs using BDD – A combined process 11](#_Toc37082301)

[5.1 Organisation, Roles, Responsibilities 11](#_Toc37082302)

[5.2 QA Process 11](#_Toc37082303)

[5.3 Test Automation Tool Validation for Cucumber/Gherkin, Scenarioo, Selenium and their interactions 11](#_Toc37082304)

[6 Prototyping 12](#_Toc37082305)

[6.1 Tool-Validation 12](#_Toc37082306)

[6.2 Application Design 12](#_Toc37082307)

[6.3 Environmental Set-Up 12](#_Toc37082308)

[6.3.1 General Set-Up 12](#_Toc37082309)

[6.3.2 OQ Test App 12](#_Toc37082310)

[6.3.3 JBA Frontend (JavaBusinessApplication) 12](#_Toc37082311)

[6.3.4 JBA Backend (JavaBusinessApplication) 12](#_Toc37082312)

[6.3.5 Database 13](#_Toc37082313)

[6.4 Specification/Formulation 13](#_Toc37082314)

[6.4.1 From User Stories to Feature Files 13](#_Toc37082315)

[6.4.2 Risk Assessment 13](#_Toc37082316)

[6.4.3 Compliance 13](#_Toc37082317)

[6.5 Test Automation 13](#_Toc37082318)

[6.5.1 Glue Files 13](#_Toc37082319)

[6.5.2 Test Reports 13](#_Toc37082320)

[6.6 QA Processes 13](#_Toc37082321)

[6.6.1 Specification and Test Management 13](#_Toc37082322)

[6.6.2 Traceability 14](#_Toc37082323)

[6.6.3 Approval 14](#_Toc37082324)

[6.7 Implications of adding new functionalities on the OQ Test-Suite 14](#_Toc37082325)

[7 Results of the Prototype Audit 15](#_Toc37082326)

[7.1 xxxx 15](#_Toc37082327)

[7.2 xxxx 15](#_Toc37082328)

[8 Learnings & Discussion 16](#_Toc37082329)

[8.1 xxxx 16](#_Toc37082330)

[8.2 xxxx 16](#_Toc37082331)

[9 Outlook 17](#_Toc37082332)

[9.1 General 17](#_Toc37082333)

[9.2 Specific Topics 17](#_Toc37082334)

[9.2.1 PQs 17](#_Toc37082335)

[9.2.2 Category 4 Software 17](#_Toc37082336)

[9.2.3 DevOps 17](#_Toc37082337)

[10 Formatting 19](#_Toc37082338)

[10.1 Quotes 19](#_Toc37082339)

[10.2 Enumerations 19](#_Toc37082340)

[10.3 Footnotes 19](#_Toc37082341)

[10.4 Figures 19](#_Toc37082342)

[10.5 Tables 20](#_Toc37082343)

[10.5.1 Tables as illustration 20](#_Toc37082344)

[10.5.2 Word tables 20](#_Toc37082345)

[10.6 Changing the font 21](#_Toc37082346)

[References 22](#_Toc37082347)

[List of Figures 24](#_Toc37082348)

[List of Tables 25](#_Toc37082349)

[List of Abbreviations 26](#_Toc37082350)

[Appendix 27](#_Toc37082351)

# Introduction

## Initial Situation

To ensure patient safety, software that directly or indirectly influences product quality must be validated according to Good Automated Manufacturing Practices (GAMP) (Wikipedia, 2018; International Society for Pharmaceutical Engineering ISPE, 2008, pp. 14, 15 and 27). This means that a formal and objective proof must be provided that the software is compliant and that its intended use is achieved (International Society for Pharmaceutical Engineering ISPE, 2008, p. 14; Johner, 2017). In support of this validation process, GAMP5 is a guide on how to achieve computerised system validation (CSV) (International Society for Pharmaceutical Engineering ISPE, 2008, p. 14).

According to GAMP5, software validation includes the verification of user requirements (PQs) and of functional specifications (OQs) (International Society for Pharmaceutical Engineering ISPE, 2008, p. 38). Until today the wega CSV specialists team experienced that PQs and OQs are often performed manually, even though test tools like hp alm (Guru99, 2020a) are supporting testing documentations in regulated companies (Evelyne Daniel, personal communication, December 19, 2019 and April 1, 2020).

## Test Optimisation using Automated Testing

As described in several blogs, manual testing has the disadvantages that it is prone to mistakes and errors while performing the tests and that it is time consuming for human testers and therefore it generates high costs (Guru99b, 2020; Hoogenraad, 2017). This is especially true, when the same tests have to be performed several times (Guru99b, 2020). On the other hand, and in contrast to automated testing, manual testing allows to evaluate aspects like user friendliness and positive customer experience (Guru99b, 2020). Therefore this project will focus its investigations on test automation for OQs as testing on functional level does not include evaluations on usability aspects (Qualitest, n.d.).

Coming back to the situation in the pharmaceutical industry, Jae Burnett investigates in her paper ‘Practical Use of Automated Tool in Computer System Compliance’, the usage of test automation tools for validated pharmaceutical environments: She emphasises that life science companies should consider automated testing as an opportunity that could add significant value to the computer system compliance process (Burnett, 2009, p. 75).

She further suggests, that if a full test automation might not always be possible, as it might be difficult to integrate the formal approval and the control of test cases into the test automation system, hybrid approaches between automation and manual process could be considered (Burnett, 2009, p. 75).

In conclusion, OQs are interesting to investigate in respect of test automation, as they are so far mainly performed manually and have therefore a good potential of optimisation in respect of reliability and cost effectiveness. And in contrary to PQs there is no need to consider aspects of user friendliness and positive customer experience, for which automated testing is not suitable.

## BDD High Level Test Automation for OQs

OQs are considered as high level testing as they verify that the functional specifications are fulfilled (International Society for Pharmaceutical Engineering ISPE, 2008, p. 38).

Behaviour driven development (BDD), on the other side, is a software development approach that includes tools for high level test automation (Smart, 2015, p.25). One of such tool is Cucumber (Smart, 2015, p.25). As the test automation script is based on a formalised natural language, that is human and machine readable, it has the potential to be a powerful asset in the sense of a hybrid approach between automation and manual processes (Burnett, 2009, p. 75; Nagy & Rose, 2018, chapter 4.6).

In the same sense, Gáspár Nagy and Seb Rose state in their book ‘Discovery: Explore behaviour using examples’ that BDD is well suited for software development in regulated areas and refer thereby also to the U.S. Food and Drug Administration (FDA) (Nagy & Rose, 2018, chapter 4.6).

## Tools

* Cucumber/Gherkin which allow to automate the test cases using an automation script (gherkin feature file) (SmartBear Software, 2020).
* Selenium which simulates the user interaction with the web application and will be controlled by cucumber and the gherkin feature file (Selenium, n.d; Jain & Sawant, 2018).
* Scenarioo that is used to display test reports with screenshots (Scenarioo, n.d.).

## Hypothesis and Research Questions

This project will be led by following hypothesis: BDD with its activities from user stories to executable specifications (formulation) and automation is a practicable approach in respect of technical feasibility, taking into account Cucumber/Gherkin, Selenium and Scenarioo, and validation requirements according to GAMP5 for OQ test automation in highly regulated environments of the pharmaceutical industry.

To evaluate the above mentioned hypothesis, the project should find answers to following questions:

* Do the artefacts out of the BDD process satisfy the GAMP5 requirements in respect of OQs?
* Can automation tools like Cucumber/Gherkin, Selenium and Scenarioo (see chapter 2.2.1) be used together in validated environments?
* How can the test suite be adapted to the evolution of the application?
* How could be dealt with new versions of the automation tools in terms of validation?

## Scope

### In Scope

* Evaluation of a test automation for OQs based on BDD for a category 5 software (custom application) according to GAMP5.
* Proposal for a Validation Procedure for Cucumber/Gherkin, Scenarioo and Selenium and their interaction.
* Implementation of a prototype for the evaluation and illustration of a test automation for different functionalities. The test automation is based on the following tools: Cucumber, Selenium, Scenarioo.
* Are to be included: User Requirements, Specification/Test Management, Risk Management, Traceability, the validation process with regard to the OQs for a Category 5 product according to GAMP5.
* If possible, outlooks on related topics (PQs, category 4 software, DevOps), based on the findings obtained.

### Out of Scope

* Validation activities according to GAMP5 outside of OQs (e.g. process validation, IQs, PQs, design reviews, used infrastructure ...).
* BDD activities that are outside the chain from user stories to high-level test automation (e.g. the implementation technique TDD, Unit/Module Testing).
* Tool evaluations for implementation (e.g. Selenium vs. Cypress) are not part of this work.
* Risk evaluation regarding the implementation of the prototype: exemplary risks are considered, but without claiming that the risk evaluation was carried out correctly and completely from a practical point of view.
* Compliance of the prototype: if possible, an exemplary compliance requirement that is technically easy to implement will be considered. However, the prototype will not be compliant to regulations like FDA 21 CFR part 11 or EU GMP Annex 11.

## Approach

The starting point for the BDD part is the book "Discovery - Explore behaviour using examples" by Gaspar Nagy and Seb Rose (ISBN 978-1983591259). In chapter 4.6 the authors state that BDD leads to improved efficiency in software development for regulated environments.

The CSV part of the project is based on the widely-used CSV standard GAMP5.

The BDD approach and the GAMP5 methodology are analysed to develop a combined process. This process is then presented in an appropriate form (e.g. BPMN) so that it can be evaluated by a wega CSV specialist. The process will be regularly reviewed by a wega CSV specialist while being developed to assure its conformity with GAMP5. Based on the developed process a prototype will be implemented as proof of concept. Few exemplary user requirements are defined as a basis for its implementation. As the final input regarding the suitability of the developed model, the prototype is audited by a wega CSV specialist. This then forms the basis for the ‘Learnings & Discussion’ and the ‘Outlook’.

In summary following activities are planned in two different streams as shown in the figures 1 & 2:

* Development of a combined process between BDD and GAMP5 in respect to OQs
* Based on point 1 to implementation of a prototype
* Audit of the prototype by wega
* Analysis of the audit results including learnings, a discussion and an outlook
* Analysis of what is needed to be done in order to use Cucumber/Gherkin, Scenarioo and Selenium in a validated environment including following aspects
  + single tools
  + combination of the tools
  + updates of the tools

Figure 1 Process to investigate OQ test automation

Figure 2 Analysis of the usability of automation tools for regulated environments

# Materials & Methods

## Analysis

Literature search: Semantic scholar, google, google scholar, swissuniversities databases

BPMN: <http://www.bpmn.org/>

Draw io desktop app from draw.io

C4 model for software architecture (<https://c4model.com/>).

## Prototyping

### Tools Included in the Evaluation

* Cucumber/Gherkin which allow to automate the test cases using an automation script (gherkin feature file) (SmartBear Software, 2020).
* Selenium which simulates the user interaction with the web application and will be controlled by cucumber and the gherkin feature file (Selenium, n.d; Jain & Sawant, 2018).
* Scenarioo that is used to display test reports with screenshots (Scenarioo, n.d.).

### Tools Used Outside the Evaluation Part

The use of the following tools is anticipated but might be subject to changes. These tools will support the work but are not relevant for the evaluation itself.

* Front end: vue CLI 🡪 <https://cli.vuejs.org/>
* Back end: Spring 🡪 <https://spring.io/>
* Database: PostgreSQL 🡪 <https://www.postgresql.org/>
* Others:  
  Maven 🡪 [https://maven.apache.org/](https://maven.apache.org/guides/introduction/introduction-to-repositories.html)   
  GitHub 🡪 <https://github.com/>

### Used tools

OQ TEST APP

* AdoptOpenJDK 14 with HotSpot as JVM (200406 - <https://adoptopenjdk.net/index.html?variant=openjdk14&jvmVariant=hotspot>) was chosen as it is an open-source version of the Java Standard Edition platform (200406 - <https://en.wikipedia.org/wiki/OpenJDK>). There are different Open JDK distributors on the market (200406 - <https://en.wikipedia.org/wiki/OpenJDK>). AdoptOpenJDK was chosen as it is recommended by stackoverflow when no specific environmental or license requirement are needed and the most standard DK build would therefore be appropriate (200406 - <https://stackoverflow.com/questions/52431764/difference-between-openjdk-and-adoptopenjdk>) .
* IntelliJ IDEA 2019.2.4 Community Edition was used as development environment: 200406 - <https://www.jetbrains.com/idea/>
* Maven version 3.6.1 was used to allow dependency management: for description see also the oq-test-app POM on <https://github.com/sableu/BDD4OQ>.......
* JUnit Jupiter is used as testing framework on the Java Virtual Machine (JVM). It is used in the version as described in the oq-test-app POM on <https://github.com/sableu/BDD4OQ>.......
* Cucumber/Gherkin and Selenium (200415 - <https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java>) were imported via the maven pom file and were used in the versions as described in the oq-test-app POM on <https://github.com/sableu/BDD4OQ>.......
* The Cucumber-Scenarioo-plugin version 0.1.0 was downloaded from (200406 (<https://github.com/andreashosbach/cucumber-reporter>) , installed in the local maven repository by creating a maven build as described (200408 - <https://maven.apache.org/guides/getting-started/index.html>) and integrated as maven dependency as described in the oq-test-app POM on <https://github.com/sableu/BDD4OQ>.......
* Scenarioo release 5.0.2 was setup and used in the standalone application version as described in (200408 - <http://scenarioo.org/docs/master/tutorial/Scenarioo-Viewer-Web-Application-Setup.html#setup-1---running-as-standalone-application>)
* Chrome Browser version 80.0
* Chrome Driver version 80.0 downloaded from (200406 - <https://chromedriver.chromium.org/downloads>) to allow automated testing by Selenium (200406 - <https://chromedriver.chromium.org/>)

### Method

Incremental Software Engineering Approach in order to evaluate the question, what implication the extension of the application with new functionalities on the test suite (see chapter 6.6).

## Evaluations

### Tool-Evaluation

Tool Evaluation for Cucumber/Gherkin, Scenarioo and Selenium: An analysis will be performed on what is needed in order to be able to use these tools in a validated environment.

### Audit of the Prototype

The audit will be done by wega. The goal of the audit is to evaluate the documents and the prototype itself that were created within the scope of this project, in respect to the GAMP5 requirements for OQs of a category 5 software. The audit report is attached to this document.

# Computerised System Validation according to GAMP5

## GAMP5: An Overview

### What is GAMP5

### Key Concepts

### Life Cycle Approach

### Product Categories

### Project Phase for Category 5 Software

### Automated Testing

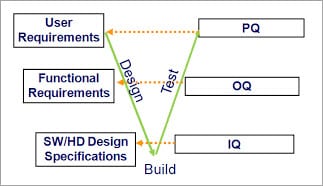
(International Society for Pharmaceutical Engineering ISPE, 2008, pp. 207)

## OQ Process According to GAMP5

GAMP5 defines OQs the following way: “Operational Qualification (OQ)” [...is a...] “documented verification that a system operates according to written and pre-approved specifications throughout specified operating ranges (GAMP5 p. 38). Whereas GAMP5 foresees following verification activity: “Testing or other verification of the system against specifications to demonstrate correct operation of functionality that support the specific business process throughout all specifies operating ranges” (GAMP5 p. 38). In the GAMP5 appendix D5 p. 212 for custom applications, this is explicitly linked to functional testing, as it states that these testings should focus on functionality that supports the specific business process based on risk and supplier assessment which exactly corresponds to the wording used to describe OQs as seen before.

The OQs must be integrated into an overall test strategy – also called test plan, which is dependent on the risks, the system to be tested and if given the faithfulness of the supplier (GAMP5, p196). As the test strategy is of higher level than the one investigated in this project, e.g. it concentrates only of OQ testing of the software and does not consider the whole system, the test strategy will not be further elaborated in the scope of this work.

For the purpose of this work following model will be used, as it highlights the detail level of the project scope and illustrates well the statements by GAMP5 as described in the first section of this chapter:



<https://www.softwaretestinghelp.com/iq-oq-pq-software-validation/>

### Roles and Responsibilities

GAMP5 foresees for software testing following roles and responsibilities (GAMP5, p 196), which might have some relevance in respect to OQs:

* The User is responsible for the system and therefore also for the approval of key test documents.
* The Test manager must plan the testings
* The Test analyst is responsible to develop test cases and – scripts
* The Testers are executing the defined test scripts. If possible, they should not have been involved earlier in the process in order that they can keep their independency.
* Test reviewer is responsible for reviewing the generated artefacts, meaning test cases, test scripts and test results. As they are reviewing the testers work, it is important, that they do not have the tester role at the same time.
* Persons from the Quality Unit have an independent role and are responsible for approval or audit of key documentations, such as reports (GAMP5 page 59).

As further stated on page 196 of GAMP5, the supplier and subject matter experts can take over some of the above-mentioned roles like tester, test manager or authoriser.

### Writing and Executing OQs

Writing OQs starts in parallel with the development of the functional specifications, by elaborating the corresponding test specifications (GAMP5 p199). GAMP5 does not mention which role is foreseen for this. The test specifications describe the overall purpose and a description of a set of test scripts (GAMP5 p198). For example, it defines which resources are needed, including tools for automated testing, the version of software under test, the test scripts to be carried out, methods, prerequisites, required reviews and approvals, etc. (GAMP5 p199). In addition, GAMP5 requires some metadata about the test specification document (GAMP5 p199).

Based on the test specification, the test analyst is responsible for developing test scripts that describe the tests to be performed in such a way that the testers can execute them consistently (GAMP5 p199). Next to some metadata, test scripts consist among others, of a title, a test description, the test objective, prerequisites, test steps, acceptance criteria and instructions about data to be recorded (GAMP5 p199 f.).

GAMP5 foresees, that the test specifications and the test scripts can be recorded in a single document if possible (GAMP5 p 200). From this statement it can be deduced, inter alia, that the test analyst is not only responsible for the test script elaboration (GAMP5 p 196), but also for writing the test specifications.

As already mentioned, the test scripts are executed by the testers (GAMP5 p 196). As a result of their testing, they have to deliver records that are reviewable. The records contain the result of the single tests (passed/failed) including needed descriptions and supporting documentations as defined in the test scripts for example screen shots (GAMP5 p 200).

Subsequent to the execution of the tests, the test results will be reviewed by the test reviewer and the results recorded in the test report (GAMP5 p. 196 and p. 200). A test report includes again some metadata like who executed and who reviewed the testings, and information about the effected testings like a summary of the test results, a summary of test failures and conclusions (GAMP5 p 200).

### Incorporating the Quality Risk Assessment

The above described OQ process, is, according to GAMP 5 accompanied by supporting processes (GAMP5 p32). One of them is risk management (GAMP5 p32), which in fact is not only a supporting process, but which is considered as one of the five key concepts of GAMP5 (GAMP5 p.20). The goal of this concept is to focus validation efforts on critical points of the computerised system (GAMP5 p.20). The quality risk management process according to GAMP5 includes in a first step the identification of functions with impact on patient safety, product quality and data integrity (GAMP5 p.50). Responsible for this step is a team of subject matter experts and key users (p106), or if regulatory compliance is concerned, it is the quality unit (p106). For the next step, the team perform a functional risk assessment and identify controls, based on the advice of the supplier, to eliminate or at least mitigate the risk to an acceptable level (p48 in combination with 50 and 106). Appropriate controls need then to be implemented GAMP p.50, e.g. by the supplier for additional software functionalities (p19) and verified (GAMP p.50) e.g. in the OQ process (p.38).

As the goal of OQ is the documented verification of software functionalities (GAMP p.38 in combination with p212), it is important to take the identified risks, their risk level and the defined control measures in respect to software functionalities into account, as the specific level of test efforts should than be determined according to the risk level and the system impact (GAMP p.50) and the controls themselves might be subject to the OQs (GAMP5, p38). The risk assessment documentation needs finally be approved by the system owner and/or the quality unit (p106)

### Specification- and Test Management

Next to the quality risk assessment, there are other supporting processes to be considered in respect to OQs for a custom application (p32). They include:

* Change management process: “Change management procedures also should be established. The point at which change management is introduced should be defined. Appropriate change processes should be applied to both project and operational phases.” (p.32) While performing OQ no change of the software is expected, as it is done on the version for which in a previous step the IQ were performed and approved (p.209). Therefore, the change management process will not be further considered.
* Configuration management: “Appropriate configuration management processes should be established such that a computerized system and all its constituent components can be identified and defined at any point” (p.32). Bringing this back to the level of the OQ process, it has to be clearly stated on which version of the software the OQ is performed, i.e. the same version as the preceding IQs (p.209).
* Traceability is the process to ensure that requirements are covered and traced to the corresponding functional specifications and design components, which then have to be linked further to the appropriate verification (p.33). This means for the OQ process, that each OQ test script needs to be traced back to the underlying functional specification, which on its turn needs to have a link to the user requirement (p.163), regulatory requirement (p.163) or risk controls (p.20) from which it is derived. 🡪 compare M5
* Document management process: “Management of documentation includes preparation, review, approval, issue, change, withdrawal, and storage”. (p. 33) 🡪 compare M9

### Validation Activities needed in Respect to OQ Testing Tools

Commercial or established testing tools are normally considered to be GAMP Category 1 🡪 compare appendix M4 but an assessment needs to be done according to the appendix M4. It could also be imaginable to be of Category 5 or anything inbetween, depending on the used tools.

### Exemplary OQ Process

# Behaviour Driven Development

Literature:

* The BDD Books: Discovery – Explore behaviour using examples   
  Gáspár Nagy and Seb Rose   
  ISBN: 978-1983591259
* BDD in Action – Behavior-Driven Development for the whole software lifecycle  
  John Ferguson Smart  
  ISBN: 9781617291654
* Writing Great Specifications – Using Specification by Example and Gherkin  
  Kamil Nicieja  
  ISBN: 9781617294105

## BDD a suitable Software Engineering Approach for Highly Regulated Environments

## The Approach: An Overview

## Writing Executable Specifications with Gherkin

* Basierend auf dem Buch: Writing Great Specifications: Using specification by example and Gherkin von Kamil Nicieja

## Automation Tools

BDD Automation Tool:   
- Cucumber/Gherkin: <https://cucumber.io/>

Further tools to complement Cucumber:  
- Scenarioo: <http://scenarioo.org/>  
- Selenium: <https://www.selenium.dev/>

# OQs using BDD – A combined process

Im folgenden ein paar Brain-storming Gedanken 🡪 gehört eigetnlich weiter unten im Kombinierten Prozess.

In the BDD Framework Functional Specifications and Functional Testautomation are based on the same Script, therefore writing OQs in BDD is writing functional specifications.

Functional Specification specifications are based on user requirements:

Validation according to GAMP5 means the formal proof for

Intended use 🡪 classical user requirements

Compliance 🡪 stakeholder requirement (regulatory affairs)

## Organisation, Roles, Responsibilities

## QA Process

## Test Automation Tool Validation for Cucumber/Gherkin, Scenarioo, Selenium and their interactions

# Prototyping

## Tool-Validation

Implication of updates of the used tools on validation

Category 1 (Selenium, Cucumber Gherkin nach GAMP5)

Category 3 (Scenarioo)

Category 5 (Cucumber Reporter Plugin)

Versionskontrolle ist wichtig 🡪 controlled by Maven, das hier über die Dependencies erlaubt es zu kontrollieren, deshalb ist Maven wichtig im Environmental Set-up

## Application Design

🡪 siehe dazu das draw.io diagram C4, level1

## Environmental Set-Up

### General Set-Up

For the development of the prototype following basic tools were used for all parts of the prototype:

Developement Environment

Maven

JUnit

Java14

### OQ Test App

Code is based on Code written by Andreas Hosbach (200406 -<https://github.com/andreashosbach/cucumber-reporter> )

### JBA Frontend (JavaBusinessApplication)

### JBA Backend (JavaBusinessApplication)

### Database

## Specification/Formulation

### From User Stories to Feature Files

### Risk Assessment

### Compliance

## Test Automation

### Glue Files

### Test Reports

## QA Processes

### Specification and Test Management

### Traceability

### Approval

## Implications of adding new functionalities on the OQ Test-Suite

# Results of the Prototype Audit

## xxxx

## xxxx

# Learnings & Discussion

## xxxx

## xxxx

# Outlook

## General

Selenium Testing Vor- Nachteile: Aufwändig wenn sich Oberfläche ändert, OQs können einfach auf verschiedenen Browsern ausgeführt werden 🡪 Testen der Applikation auf verschiedenen Browsern.

Digitization: Prozess möglichst klassisch gehalten, so wie er jetzt, manuell schon ist. Test automation passt sich diesem Prozess an. Denkbar wäre, OQs schon im Build-Prozess (inside Testing vs. Outside testing) durchzuführen. Würde aber bedeuten, dass die Reihenfolge IQ 🡪 OQ nicht mehr stimmen würde und dass die Qualificationen der Auditoren und der QA Menschen sich verschieben müssten 🡪 Müssten Code lesen und beurteilen können (vgl. Kleines BDD Buch)

## Specific Topics

### PQs

### Category 4 Software

### DevOps

# Formatting

## Quotes

Enable „Replace "Straight quotes" with “smart quotes”” in the AutoFormat section of the AutoCorrect options. Default language is „English (U.K.)“.

Variant: «…»

## Enumerations

Please use the „List Bullet“ template.

* Item 1
* Item 2

## Footnotes

Example for text containing a footnote.[[1]](#footnote-1)

## Figures

Please use the following option to insert graphical illustrations (e.g. Power Point charts):

Home Tab 🡪 Paste 🡪 Paste Special… 🡪 Picture (Enhanced Metafile).

Use the template „Figure“ for the formatting of illustrations. After inserting the illustration it must be anchored to the text. For this purpose, place the cursor in the target area of the figure (a paragraph which is formatted as „Figure“), do a right click on the figure and follow the instructions below:

Size and position 🡪 Text Wrapping 🡪 In Line with Text

The size of the illustration can also be changed in „Size and position“.

To **name** your illustration, place the cursor below your figure and choose „Insert Caption“ from tab „References“.

Caption: Figure number, colon, space, caption.

Please compare the following example: Figure 1



Figure 3: Example illustration

## Tables

### Tables as illustration

For tables which are imported as **illustration** or **object** e.g. from excel, use the same procedure like for figures (2.4) except that the illustration is formatted as “Figure Table” instead of “Figure”

For the caption use the same label “Table” as for word tables and format it as “Caption Table”.

Table caption: Below the table, like in our example: Table 1



Table 1: Example for a table imported as illustration

### Word tables

Choose our template „Table heading“ for title row and „Table text“ for table content. Please follow our instructions to caption your table:

Tab “References” 🡪 “Insert caption” (Position: Below selected item)

Reformat accordingly to „Caption Table“ (Styles).

Caption: Table number, colon, space, caption.

|  |  |
| --- | --- |
| Bezeichnung der  Grössenklasse | Anzahl Beschäftigte  (Vollzeitäquivalent) |
| Kleinunternehmen | 10 bis unter 50 |
| Mittlere Unternehmen 1 | 50 bis unter 100 |
| Mittlere Unternehmen 2 | 100 bis unter 250 |

Table 2: Example for a table created by Word

## Changing the font

The font for the whole document can be changed by the “normal” template (is not recommended). All other styles which are used in this document are linked to this template.

To change the font anyway:

Tab „Home“ 🡪 „Styles“ 🡪 Choose template „Normal“ in the dropdown-menu (right click) 🡪 „Modify“

References

Burnett, J. (2009). Practical Use of Automated Tools in Computer System Compliance. Journal of Validation Technology, 15(4), 73–76. Retrieved from https://www.ivtnetwork.com/journal-validation-technology/journal-of-validation-technology-3004

GitHub, Inc. (2020). Build for developers. Retrieved March 27, 2020, from https://github.com/

Guru99. (2020a, March 19). Introduction to HP ALM(Quality Center). Retrieved April 3, 2020, from https://www.guru99.com/hp-alm-introduction.html

Guru99. (2020b, March 23). Automation Testing Vs. Manual Testing: What’s the Difference? Retrieved April 3, 2020, from https://www.guru99.com/difference-automated-vs-manual-testing.html

Hoogenraad, W. (2017, October 11). Was ist manuelles Testen? Retrieved April 3, 2020, from https://de.itpedia.nl/2017/10/11/wat-is-handmatig-testen/

International Society for Pharmaceutical Engineering ISPE. (2008). GAMP 5 - A Risk-Based Approach to Compliant GxP Computerized Systems. North Bethesda, USA: ISPE.

Jain, N., & Sawant, T. (2018, February 5). Setup for Selenium with Cucumber Using Maven. Retrieved April 3, 2020, from https://www.axelerant.com/resources/team-blog/setup-for-selenium-with-cucumber-using-maven

Johner, C. (2017, April 5). Computer System Validation CSV. Retrieved March 29, 2020, from https://www.johner-institut.de/blog/regulatory-affairs/computer-system-validation-csv/

Nagy, G., & Rose, S. (2018). Discovery - Explore behaviour using examples. Victoria, Canada: https://leanpub.com/.

Nicieja, K. (2018). Writing Great Specifications. Shelter Island, NY, USA: Manning Publications.

Object Management Group. (1997 to present). Business Process Model and Notation. Retrieved March 27, 2020, from http://www.bpmn.org/

Qualitest. (n.d.). Functional Testing vs. Usability Testing. Retrieved April 3, 2020, from https://www.qualitestgroup.com/white-papers/functional-testing-vs-usability-testing/

scenarioo. (n.d.). Retrieved March 27, 2020, from http://scenarioo.org/

Selenium. (n.d.). Selenium automates browsers. That’s it! Retrieved March 27, 2020, from https://www.selenium.dev/

Smart, J. F. (2015). BDD in Action. Shelter Island, NY, USA: Manning Publications.

SmartBear Software. (2020). Tools & techniques that elevate teams to greatness. Retrieved March 27, 2020, from https://cucumber.io/

The Apache Software Foundation. (2002 to present). Maven. Retrieved March 27, 2020, from https://maven.apache.org/

The PostgreSQL Global Development Group. (1996 to present). PostgreSQL: The world’s most advanced open source database. Retrieved March 27, 2020, from https://www.postgresql.org/

VMware. (2020). Spring makes Java... Retrieved March 27, 2020, from https://spring.io/

Wikipedia. (2018, June 26). GxP. Retrieved March 29, 2020, from https://de.wikipedia.org/w/index.php?title=GxP&oldid=178646512

You, E. (2018 to present). Vue CLI. Retrieved March 27, 2020, from https://cli.vuejs.org/

List of Figures

[Figure 1 Process to investigate OQ test automation 4](#_Toc36826148)

[Figure 2 Analysis of the usability of automation tools for regulated environments 5](#_Toc36826149)

[Figure 3: Example illustration 19](#_Toc36826150)

Annotation: A convenient way to create default list of figures and list of tables in the text is to follow the sequence:

Tab „References“ 🡪 „Captions“ 🡪 „Insert Table of Figures“

Now, the caption is formatted automatically according to our template „caption“.

For the figures the template „Figure“ and for tables the templates „Table heading“ and „Table text“ should be used.

Further information can be found in chapters: 2.4 and 2.5

List of Tables

[Table 1: Example for a table imported as illustration 3](#_Toc416160904)

[Table 2: Example for a table created by Word 3](#_Toc416160905)

List of Abbreviations

|  |  |
| --- | --- |
| Abbreviation | Description |
| BDD | Behaviour Driven Development: A software development approach developed by Dan North covering the whole software lifecycle |
| CSV |  |
| GAMP |  |
| GAMP5 |  |
| GxP |  |
| JVM | Java Virtual Machine |
| OQ | Operational Qualification: …. |
| PQ | Performance Qualification …. |
| FDA | U.S. Food and Drug Administration ….. |
|  |  |
|  |  |
|  |  |

Appendix

References and appendix are formatted as „Heading 1 without numbering.“ Thereby this entry is listed in the table of contents (but without numbering). To structure your appendices use CAPITAL LETTERS (Appendix A, Appendix B etc.).

1. Example Footnote. [↑](#footnote-ref-1)