



**Freie Universität Bozen**  
**Libera Università di Bolzano**  
**Università Lìdia de Bulsan**

Fakultät für Informatik  
Facoltà di Scienze e Tecnologie informatiche  
Faculty of Computer Sciences

## BACHELOR THESIS

adfasdf

Jan Schmid Niederkofler

Supervisor: Andrea Janes  
October, 2015

# Contents

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Einleitung . . . . .	1
1.2	Testdokumentation / Testautomatisierung . . . . .	2
1.3	Ausgangssituation . . . . .	2
1.4	Schwer bis unmöglich manuelle Testfälle zu warten . . . . .	3
<b>2</b>	<b>Grundlagen und Stand der Technik</b>	<b>4</b>
2.1	Hp Tool . . . . .	4
2.2	UFT . . . . .	4
<b>3</b>	<b>Herausforderungen</b>	<b>5</b>
3.1	Rechte in einen Großkonzern . . . . .	5
3.2	Semaphore / Visual Basic . . . . .	5
3.3	XML Parsing . . . . .	5
3.4	Zeigen von großen Datenmengen auf wenig Feldern . . . . .	5
3.5	Workflow in HPQC . . . . .	6
<b>4</b>	<b>Lösungen</b>	<b>7</b>
4.1	Object Repository . . . . .	7
4.2	UTF XML . . . . .	7
4.3	Workflow . . . . .	7
4.4	Binary Repräsentation . . . . .	8
<b>5</b>	<b>Problemstellung</b>	<b>9</b>
5.1	Problemstellung . . . . .	9
5.2	Warum HP Quality Center . . . . .	9
5.3	Requierments . . . . .	9
<b>6</b>	<b>Implementation</b>	<b>10</b>
6.1	OTA / tdconnect, Visual Basic Script . . . . .	10
6.2	Listen adden + Elemente . . . . .	10
6.3	Aus XML auslesen . . . . .	11

<i>CONTENTS</i>	ii
6.4 Ergebnis . . . . .	11
<b>7 Future</b>	<b>12</b>
7.1 Zukünftiger Einsatz und Ausbau . . . . .	12
7.2 Allen Testfällen . . . . .	12
7.3 Manuelle Eingabe aus der End-User Maske . . . . .	12
7.4 Direkt Automatisiert . . . . .	12
<b>8 Literature review</b>	<b>14</b>

# List of Tables

## List of Figures

# Listings

# Chapter 1

## Einleitung

### 1.1 Einleitung

Testfälle sind die einzelnen Tests die durchgeführt werden um sicherzustellen, dass Anwendungen funktionieren wie erwartet. Die Testfälle werden erstellt von den jeweiligen Verantwortlichen erstellt und besitzen dann einen fixen Ablauf. Mit HP Application Lifecycle Manager werden diese dokumentiert. Durch die Erweiterung dieses verbessert sich die Wartbarkeit der dokumentierten Testfälle. Ständige Veränderungen bei den MAX Anwendungen erschweren diese. Auch verbessert wird die Eingabe der dokumentierten Testfälle über vorgegebenen Dropdown Menüs.

Supply Chain Lufthansa Lufthansa ist mit fast 126.000 Mitarbeitern eine der größten Fluggesellschaften weltweit. Diese wird aufgeteilt in Passagierbeförderung, Fracht, Technik, Catering, IT- Dienstleistungen und Service –und Finanzgesellschaft. Dieses Projekt wurde in der Lufthansa Technik realisiert. Genauer im Bereich der Supply Chain. Der Einsatzbereich der Supply Chain ist die Versorgung von Materialien für Kunden. Diese werden unterteilt in Material das repariert und wieder im Umlauf gebracht wird, wie ein Flugzeuggetriebe, als auch Material das einfach aufgestockt wird wie Schrauben etc. Um den Kunden ein Interface bieten zu können über denen sie gewünschte Bestellungen abgeben können, bietet Lufthansa verschiedene Software-Lösungen an. Diese unterstehen regelmäßigen Updates (bis zu 7 Updates im Jahr)((source)). Um sicherzugehen das nach einem Update alles noch funktioniert wie es sollte wurden bis vor kurzen Integrationstests abgehalten, die daraus bestanden die Applikationen manuell durchzuklicken. Diese Arbeitszeit soll nun mit automatischen Tests reduziert werden. ((insert source about Lufthansa))

## 1.2 Testdokumentation / Testautomatisierung

Die Vorteile der Testautomatisierung sind in erster Linie Zeitersparnis durch die Abnahme der Testfälle per Computer. Automatisierte Tests laufen, ohne menschliche Interaktion, in bis zu 20 h durch((insert source, fact check with reality)). Manueller Testfälle hingegen dauern Tage und bedürfen Menschen die sie durchklicken. Durch die Reduzierung der menschlichen Interaktion auf ein Minimum entstehen auch weniger Probleme falls ein Experte für einen bestimmten Bereich nicht anwesend sein kann, sondern wird nur noch benötigt falls die Tests nicht erfolgreich abgeschlossen werden können. Ein Automatisierungszyklus beginnt mit der Auffindung aller Testfälle. Diese werden dann dokumentiert. Der erste Schritt der Dokumentation beginnt mit einem Tool namens „SAP Workforce Performance Builder“. Dieses ist ein Werkzeug um automatisch bei jeden User Input Screenshots zu erstellen. So kann ein Testfall in erster Instanz abgebildet sein. Ein Schritt zur Automatisierung ist die Dokumentation. Hauptsächlich liegt der Vorteil bei Dokumentation dabei, dass auch wenn ein Experte oder Wissensträger verloren geht, nicht das Wissen selbst verloren ist. Die Dokumentation beginnt mit Übertragung des Testfalls in das HP Tool „Application Lifecycle Manager“ ((insert screenshot)). Hier werden die Schritte einzeln in eine Tabelle eingegeben. Ein einzelner Testschritt besteht aus: wo genau die Aktion stattfindet, was für ein Input kommt, welches Objekt an der Maske angesprochen wird. Ein Testschritt kann auch eine SQL abfrage sein oder der Start eines Scripts. Diese Dokumentation wird später verwendet um Testfälle manuell durchzuspielen. Manuelle Tests bedarf es bei Tests die nicht zu Automatisieren sind. Parallel dazu werden manuelle Test verwendet bei fehlgeschlagenen automatisierten Tests den Fehler zu suchen.

Ist ein Testfall komplett übertragen und kontrolliert (4 Augen Prinzip) wird er nach Budapest ((Testfactory/ Testautomatisierung))f reigegeben, wo mit Hilfe von Visual Basic in Script erstellt wird, der den Testfall nachspielt. Dieses Script ist unabhängig von der Dokumentation im HPQC, auch wenn dieses darauf basiert. Änderungen an der Maske werden nur in den automatisierten Testfällen eingespielt.

Object Action genauer beschreiben + Beschreibungssprache und Formale Sprache (aus dem Wiki) + outsourcing probleme

## 1.3 Ausgangssituation

Die Übertragung der Testfälle in HP Quality Center funktioniert, ist durch Textfeld basierten Input aber Fehleranfällig. Noch dazu ist der Aufwand bei Änderungen zu hoch, so dass diese bei der großen Anzahl an Testfällen die Dokumentation Anpassung ganz wegbleibt. Die betrifft jetzt nicht automatisierte Tests. Hier das Visual Basic Script angepasst. ((Es funktioniert aber könnte besser sein, bei Pflege))



## 1.4 Schwer bis unmöglich manuelle Testfälle zu warten

((Anders Beginnen, kein geeigneter erster Unterpunkt, braucht mehr roten Faden))  
Ein entscheidender Punkt in der Dokumentation ist die Angabe wo ein bestimmtes Element sich befindet. Dies wird mit Object Path und Object Name angegeben. Bei der Übertragung in das HP tool wird dies manuell eingepflegt. Ändert sich die Maske, durch die regelmäßigen Updates, wird diese Angabe falsch. Dies richtig zu Pflegen beinhaltet alle Testfälle durchzugehen und wo auf die geänderte Oberfläche zugegriffen wird dementsprechend zu ändern. Folglich sind diese Tests nicht skalierbar und bereits bei wenigen Testfällen ist mit Änderungen großer Aufwand verbunden.

## Chapter 2

# Grundlagen und Stand der Technik

### 2.1 Hp Tool

Wenn du ein „HP Tool“ erwähnst, musst du dieses Tool kurz erklären. Das würde in „Grundlagen und Stand der Technik“ gut hineinpassen. Das HP Tool Application Life Cycle Manager oder kurz HP ALM ist eine Sammlung von Software Applikationen die den Lebenszyklus von Software unterstützt. In diesen Fall die Erstellung von automatisierten Testfällen. HP. Spezifisch wird hier HP Quality Center verwendet das eine einzige Plattform für Test- und Software Zyklus Management bereitstellt. Die Aufgabe der Software Factory ist nicht die genutzte Software nachzubasteln sondern eigentständige Programme schreiben die auf einer existierend Testumgebung die gewünschten Abläufe rekonstruiert.

TODO: -> Outsourcing theorie, zumindest a bissl 4) Grundlagen -> Formale Sprache, Object Path ersatz, find that document->gefunden

[insert source and more explanation about hp here + Cycles?]

### 2.2 UFT

## Chapter 3

# Herausforderungen

[In der Zusammenfassung so lange nicht aktive Probleme. Dargestellt wie in Zukunft ausbaufähig und könnte und den Umständen Probleme machen. Alles komplett abschließen, keine Offenen Enden]

### 3.1 Rechte in einen Großkonzern

Bei einen solchen Tool arbeiten zu können, kann man nicht auf Produktion arbeiten sondern ist es von Vorteil in einer „Sandbox zu arbeiten“. Diese verhindert das beim Arbeiten beim Tool Leute stört die gerade mit diesen Arbeiten.

### 3.2 Semaphore / Visual Basic

Semaphore sind eine Herausforderung da HPQC Zugriff auf die Daten mit semaphoren kann sperren und es an Lösungen gedacht werden muss die immer funktionieren.

### 3.3 XML Parsing

### 3.4 Zeigen von großen Datenmengen auf wenig Feldern

Die Auswahl von dem gewünschten Element wird schwierig wenn eine Liste tausende Elemente enthält. Eine Möglichkeit dies zu vereinfachen ist die Liste auf mehreren Listen aufzuteilen, die dynamisch generiert werden, abhängig von der vorigen Auswahl.

### **3.5 Workflow in HPQC**

Der Workflow in HPQC besteht aus großteil unkommentierten 2000 LOC ohne Dokumentation, alles auf nur eine Seite. Dies kann dazu führen das leicht die Übersicht verloren geht und Fehler passieren.

# Chapter 4

## Lösungen

### 4.1 Object Repository

Im Fokus steht ein Object Repository für Object Path und Object Name. Pro Eintrag in dieser Liste muss jedes Element genau identifiziert sein. Aus diesem soll der User später die Einträge vom Element in das HPQC nehmen. Zugriff hat dieser direkt im HPQC bei der Eingabe-Maske über ein Drop-Down Menü.

### 4.2 UTF XML

Die Liste wird dynamisch aus bereits automatisierten Tests erstellt. Zugriff darauf hat man mit UFT (Unified Functional Testing). Über diese Software kann man über bereits erstellte Scripts in VBS die angesprochenen Elemente auslesen. Hier kann man alle Test-Scripts ansprechen und hat somit alle Elemente die angesprochen werden abgedeckt. Die Ausgabe erfolgt in einem XML Format.

### 4.3 Workflow

Der Workflow ist eine Schnittstelle an der man sich anbinden kann um die Übertragung ins HPQC zu verändern. In unseren Fall die Eingabemaske. Einfache Textfelder (in HTML Editfields) werden durch Listen ersetzt. Aus diesen wählt man sich das Ergebnis gewünschte Ergebnis. So ist es möglich ein Element zu verändern und über alle Testfälle übergreifenden ändern mit einem einzelnen Eingriff in das Object Repository. (wo denn?). Parallel dazu wird die Fehlerquote reduziert, die mit manueller Eingabe vorkommen.

## 4.4 Binary Repräsentation

Der Object Path wird in Zukunft mit mehreren Feldern angezeigt, dass die Eingabe erleichtert. Wird eine Applikation ausgewählt gibt es im ersten Object Path Feld nur die Elemente auszuwählen die auch in der Maske sind. So geht es mit Unterseiten oder Reitern weiter bis wir beim gewünschten Objekt sind. Dieser wird mit der Element- Art verknüpft, sprich Button, Editfield usw. Das soll dazu führen das nur die Objekte dargestellt werden die auch ausgewählt werden die auch mit der vorher eingegeben Objekt Art übereinstimmen. Vor dieser Auswahl bleibt das Feld Object Name ausgegraut. Es entsteht eine Baumstruktur die das Navigieren in diesen einfacher gestaltet als eine große Liste.

# Chapter 5

## Problemstellung

### 5.1 Problemstellung

### 5.2 Warum HP Quality Center

Durch eine einzige Plattform die dieses Tool für Lebenszyklen und Test Management bereitstellt macht es für die Testautomatisierung geeignet. Die Testdokumentation und dessen Automatisierung folgt gewissen Zyklen [explaining here or somewhere else, or at all? ] die sich in HP QC unter „Status“ wiedergeben lässt. Das Projekt ist nicht zentral und wird teilweise outsourcend und bei Setzung dieses Status weiß wann welche Personen jetzt für den nächsten Schritt verantwortlich sind. Diese werden benachrichtigt sobald dieser in den eigenen Aufgabenbereich fällt.

### 5.3 Requierments

Im Rahmen dieser Arbeit wird über UFT eine XML aus den bereits automatisierten Testfällen alle bekannten Objekten ausgelesen und über in ein Object Repostory eingespielt. Auf diesen wird über den ALM Explorer mit Dropdown Menüs dem Endnutzer Zugriff gegeben. Der gesamte Pfad zu einem Objekt hin muss über dieselbe HP QC GUI erkenntlich sein. Es muss eine Möglichkeit bestehen Objekte die neu hinzugekommen sind trotzdem einzuspielen. Am besten ohne externe Anbindungen und direkt über den Input an der grafischen Oberfläche. Im Besonderen darf das Projekt HP QC nicht deutlich langsamer machen. Wird das Tool bis zur Unbrauchbarkeit langsamer überwiegen die Nachteile den Vorteilen und das Projekt wird überflüssig. [Noch nicht sicher wo das hingehört]

# Chapter 6

## Implementation

### 6.1 OTA / tdconnect, Visual Basic Script

Die Open Test Architecture API (OTA) erlaubt es mit tdconnect Zugriff auf HP Application Lifecycle Manager über externe Scripte zu haben. Die Schnittstelle wird mit Visual Basic Script angesprochen. Für diesen Zweck wurde Visual Basic Script C(Sharp) bevorzugt da diese Sprache für HPQC am geeignetsten ist. VBS ist gut dokumentiert und verfügt über eine viel größere Bibliothek was die Steuerung des Tools anbelangt. Der Workflow von HPQC ist auch mit VBS beschrieben, was eine Andockung mit der selbigen vereinfacht. C(Sharp) scheint zu diesen Zeitpunkt noch nicht für die Software ausgearbeitet genug zu sein. Das kann sich in Zukunft vielleicht ändern.

### 6.2 Listen adden + Elemente

```
1 ' Add list under costumization -> Project Lists;
2 Sub AddList(tdConnection)
3 MsgBox("adding lists like a champ")
4 ' custom is the connection with the elements, oList \
   are the lists access
5 Dim custom
6 Set custom = tdConnection.Customization
7 Dim oLists
8 Set oLists = custom.Lists
9 Dim nameList
10 Dim counter
11 counter = 0
12 While counter < 10
```



```
13 nameList = "LHT_Test_Nr " & counter
14 oLists.AddList (nameList)
15 'AddList RemoveList; Adding or removing List to the \
   queue on the list object
16 Call AddItemToList(tdConnection, nameList) ''adding an\
   item to thelist, does not work if node is in queue \
   and not commisted
17 counter = counter + 1
18 Wend
19 custom.Commit
20 End Sub
21 [Code beschreiben]
```

### **6.3 Aus XML auslesen**

### **6.4 Ergebnis**

# **Chapter 7**

## **Future**

### **7.1 Zukünftiger Einsatz und Ausbau**

### **7.2 Allen Testfällen**

Wird bei allen dokumentierten Testfällen benützt. [Anstatt dokumentierte Testfälle -> Dokumentation „Synonyme“ -> Standardisieren, einheitliche Beschreibung der Testfälle, Formalisieren-> sollen alle gleich Aussehen]

### **7.3 Manuelle Eingabe aus der End-User Maske**

-future Das Obeject Repository kann nicht nur über das Costumization Interface oder Scripten geändert werden sondern auch über eine End-User Maske.

### **7.4 Direkt Automatisiert**

-future Da alle Objekte bereits als Elemente vorhanden sind gibt es die Möglichkeit dies später zu benützten um zum Teil direkt schon zu automatisieren.

## **Fazit**

# **Chapter 8**

## **Literature review**

The goal: find out how micro-enterprises (synonyms?) how they develop, which processes they adopt, what problems they have its important to find out how they develop, which tools they use, which languages they use

since the goal is to adapt/use/configure sonarqube, we want to find out the needs of micro-enterprises

maybe the result is: - we have to install sonarqube - there are 3 typical problems everybody has - for this 3 typical problems we develop sonarqube plugins or we recommend a open source plugin and we recommend how to configure it

# **Bibliography**