AssignmentType

University of Applied Sciences Zittau/Görlitz

University of Applied Sciences Zittau/Görlitz

Faculty of ...

...

AssignmentType

This is the Title of this Work

Max Mustermann,

Görlitz, DDth Month 20JJ

Bibliographic Description

MUSTERMANN, MAX,
This is the Title of this Work,
University of Applied Sciences Zittau/Görlitz, Faculty of ...
AssignmentType, 20JJ
-PP pages, -FF figures, -TT tables, -CC citations, -AA algorithms

submitted by:	Max Mustermann born on DD$^{\text{th}}$ Month JJJJ in MusterCity mmustermann@hszg.de
submitted on:	DD$^{\text{th}}$ Month 20JJ
Advisors:	Prof. Dr. ... [1] Prof. Dr. ... [2]
Defence:	prospectively DD$^{\text{th}}$ Month 20JJ

[1] University of Applied Sciences Zittau/Görlitz, Faculty of ...
[2] University of Applied Sciences Zittau/Görlitz, Faculty of ...

Abstract

In the ...

Keywords: Keyword 1, Keyword 2, Keyword 3

Referat

In der ...

Schlagworte: Schlagwort 1, Schlagwort 2, Schlagwort 3

TABLE OF CONTENTS

1 INTRODUCTION

Write something about IT Service Management, its goals and why software supports achieving these goals. Also describe goals of your project here - why are we doing this? (About one page in sum.)

2 Evaluation of General Characteristics of the Software

Write a few introductory lines.

2.1 Basis facts about the software solution

Please describe the basic characteristics of the software. Is is a web-based solution? Is it installable or just available as a service? How much effort does the installation take (describe and give a time for the installation)? How old is the software? When did its development start? Which version is it (0.1?)? Is it well established? How many customers are using it? Is it still further developed - are new releases planned and when was the last release?

2.2 Versions and price

Please describe versions, prices and cost models of the software. Is the software open source?

2.3 Usability

Does the user interface and layout look up to date? Is it easy to understand and learn the functionalities of the software? Is the user interface efficient? Are the functionalities intuitive? Is the choice of the colors reasonable? Please answer the questions and grade the usability of the software - 5-extraordinary good - 1-very poor.

2.4 Performance

Describe if the performance of the software. Is it satisfying? Are the response times in order? Are there any performance critical operations? Please answer the questions and evaluate the performance between 5-extraordinary good and 1-the screen war frozen with the first klick :)

2.5 Documentation

Describe how well the software is documented on a scale between 5-extraordinary well to 1-no documentation available. Please also describe which tutorials, seminars or certification courses are available.

2.6 Support

How is the software supported? Is there a number to call, a web form/ email address, a forum (and if so, how are the replies - fast and well explaining?)? Does the support have to be payed? Try to ask the support a question and report the reaction. If you got a reply - was the reply competent? Evaluate the support between 5-extraordinary good to 1-no support.

2.7 Errors and Bugs

Describe errors and bugs of the software which you found during testing.

3 Evaluation of the ITSM Specific Functionalities

3.1 ITSM Processes

Describe the support of the processes with a number from 1-5 (5 for fully supported), describe also "what" it supported and "what not". Use our scribe notes with detailed information about the processes to have an idea about what could be supported. Give further descriptions "how" it is supported and about the usability of the process features in the column "comments". Describe as many details as possible.

3.2 IT Service Management Roles

Described the support of the ITSM roles individually.

3.3 Scenarios

3.3.1 Design of an Email Service

Design an email service as a new service within the service management software which has to be evaluated. Please consider if supported:

- Creating and formalizing a *Service Design Package* (SDP)
- Creating and formalizing *Service Acceptance Criteria* (SAC)
- Creating and formalizing *Service Level Requirements* (SLR) which develop to the *Service Level Agreement* (SLA) within the design process.

Lookup all necessary information about the documents in your scribe notes. Are measurable goals for the services supported by the software? How about SLA-Monitoring (SLAM) and Service Improvement Plans? Are different service levels for the same service supported? Are different SLA types supported (service-based, customer based, multi-level)?

Describe how your software supports this procedure. Use also screen shots to visualize that.

3.3.2 ITIL Roles

Check which of the ITIL roles (`http://wiki.en.it-processmaps.com/index.php/Roles_within_ITIL_V3`) are supported by your software or how it is possible to define ITIL roles for your service. Then establish as many ITIL roles as possible in your test system.

Table 3.1: Supported ITSM Processes

ITSM Process	Supported (1-5)	Comments
Service Strategy Strategy Generation Demand Management Service Portfolio Mgmt. Financial Management		
Service Design Service Catalogue Mgmt. Service Level Management Capacity Management Availability Management IT Service Continuity Mgmt. Information Security Mgmt. Supplier Management		
Service Transition Transition Planning and Support Change Management Service Asset and Configuration Mgmt. Release and Deployment Mgmt. Service Validation and Testing Evaluation Knowledge Management		
Service Operation Event Management Incident Management Request Management Problem Management Access Management		
Continual Service Improvement 7-Step Improvement Service Reporting Service Measurement		
General Service Desk Raci Authority Matrix		

3.3.3 Service Catalogue

Check if the life cycle phases *Service Pipeline*, *Service Catalogue* and *Retired Services* are supported by your software. Add the new Email Service as new service to the service pipeline, then transfer it to the service catalogue and finally retire the service. Is it possible to get information about all services in the life cycle? If yes describe how that works. Which information are available for each service? Is there a division in *business service catalogue* and *technical service catalogue*?

Describe how your software supports this procedure. Use also screen shots to visualize that.

3.3.4 Further Management of the Email Service

Note that you need to preserve capacities for your email service. Is there a capacity management information system? If yes then develop a capacity plan and reserve capacities for your email service. How about planning of personnel resources?

Also develop an availability plan for your email service if this is supported by the software. Is the ITIL role of the Availability Manager supported?

Is it supported by the software to plan capacities and availability in an alternating way?

How about a continuity plan for your Email service - is it possible to establish that in the software? Develop an exemplary continuity plan for the email service.

How is information security management and supplier management supported. If supplier management is supported, add exemplary suppliers with underpinning contracts for basis functionality of your email service - e.g. use cloud services to store the emails or your service. Safe the contract in the supplier and contract data base if available (SCD - Supplier and Contract Database).

3.3.5 Configuration Management

Add all necessary configuration items to implement your email service as configuration items in the Configuration Management Database (CMDB) of the system. Note that it is normal to have thousands if not millions of configuration items within the system. This is of course not the case in our test system but consider adding different CI-types as necessary infrastructure and servers, routers, documents, chairs, IP-adresses, other hardware and services, buildings, persons, etc. Add for each of the items a configuration record with all describing information. Are agents supported, which write infrastructure automatically in the CMDB? Is a CMS available to host the CMDB?

3.3.6 Change Management

Simulate all three types of changes which you know from the lecture. Rise a RFC (request for change) first. Then initiate a normal change to your service, e.g. migrating the email service to a new server. Also increase the storage capacity of your email service from the initial value of x MB to y MB. Since this change is not very risky, establish it as *standard change*. Define the CAB (Change Advisory Board) and a ECAB (Emergency Change Advisory Board). Simulate the situation that your email system got hacked - how is the handling of that situation supported by your software?

Now think about developing your email service further, e.g. by adding IMAP or POP or by adding a web interface. Plan those changes in a *Forward Schedule of Changes* (FSC) if available. Are the 7Rs supported somehow to make sure that the right questions are asked before the change is realized?

3.3.7 Definite Media Library

Is there a *Definite Media Library* (DML) to represent media CIs in different versions? All media items should also be available as CI in the CMDB. Think about which software you need to realize your email service and create configuration items with Master copies of this software which are also represented in the DML.

3.3.8 Knowledge Management

Consider to safe further information about your email service, as current users, status information about the service utilization, down times, etc. Safe those information to the *Service Knowledge Management System* (SKMS) if possible. Is it supported to safe status information about the service automatically?

3.3.9 Event, Incident and Problem Management

Since we did not consider Service Operation in the lecture so far, we need to have a brief introduction. Note that events can be any change of status - informative events where no action is required, hitting some predefined level for some parameter, exceptions that some resource or service is not in operation as usual. Incidents are events which interrupt or potentially interrupt a service. Events which are incidents have to be reported using an interface of the *Event Management System* to the *Incident Management*. Also the breakdown of an configuration item which has no direct impact for a service is an incident. Note that incidents need to be treated and prioritized according to *impact* and *urgency*. Then, a problem is a not known cause for one or several incidents. There are *problem records* with the complete history of a problem and also a *Known Error Data Base* (KEDB) with all of these known errors and their records.

The single point of access for the customer is always the *service desk*. Each call of a customer is an incident.

Consider different events, incidents and problems for your email service and check how they can be formally treated with your software system. How is the work of the service desk supported? Treat events, incidents and problems based on a few examples.

4 Conclusion

Conclude your evaluation with some overall statements (about 1/2 page).

A This is the First Appendix Chapter

A.1 Appendix Section

B This is the Second Appendix Chapter

B.1 Another Appendix Section

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF LISTINGS

LIST OF SYMBOLS

Notations

$\Box$	end of a proof
$\Diamond$	end of an example or a remark
x	scalar
$\mathbf{x}$	vector
$\lvert\mathbf{x}\rvert$	norm of vector $\mathbf{x}$, $\lvert\mathbf{x}\rvert = \sqrt{\mathbf{xx}}$
$\lvert\mathcal{S}\rvert$	number of elements in a set $\mathcal{S}$
$\mathbf{xy}$	scaler product of the vectors $\mathbf{x}$ and $\mathbf{y}$
$\boldsymbol{X}$	matrix
$\boldsymbol{X}^{\mathrm{tr}}$	transposed matrix
procedure()	some procedure in an algorithm
data	some data structure used in algorithms

Greek Symbols

α	state in the state space
$\boldsymbol{\alpha}$	generalized state
β	state in the state space
$\boldsymbol{\beta}$	generalized state
$\Gamma^{t}_{\boldsymbol{\alpha\beta}}$	transition probability from population $\boldsymbol{\beta}$ to $\boldsymbol{\alpha}$ at time t

Latin Symbols

a_i	a certain constant
$A(I)$	algorithm A started with instance I

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimization
AJAX	Asynchronous JavaScript and XML
B2B	Business-to-Business

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Görlitz, den DD.MM.20JJ

Mustermann, Max