

M 156

Neue Services entwickeln und implementieren

Emanuel Duss; Semir Jahic

# Testprotokoll

Evaluation Ticketsystem



Projektbezeichnung	Ticketsystem evaluieren
Projektleiter	Emanuel Duss
Verantwortlich	Projektleiter
Erstellt	22.09.09
Letzte Änderung	2009-11-03 um 13:35:16
Zustand	Zur Prüfung
Pfad	X:\Schule\4_Lehrjahr\156_Neue_Services_entwickeln_und_implementieren\03_Evaluation_TTS\03_Testprotokoll\03_Testprotokoll.odt

## Projektmitglieder

Emanuel	Duss	Im Bienz 15	6170 Schüpfheim	emanuel.duss#gmail.com	EDU
Semir	Jahic	Moosmatte 32	6182 Escholzmatt	semir.jahic#gmail.com	SJA

## Änderungen

Datum	Version	Kapitel	Beschreibung	Autor
2009-10-20	0.1	ALLE	Grundgerüst vom Dokument	EDU

## Prüfungen / Reviews

Datum	Version	Kapitel	Beschreibung	Autor
	1	Alle		GAS

## Infos

Zuletzt bearbeitet	2009-11-03
Heute	2009-11-03
Bearbeitungszeit	00:04:55
Lehrjahr des Moduls	4. Lehrjahr; 2009 / 2010
Pfad	X:\Schule\4_Lehrjahr\156_Neue_Services_entwickeln_und_implementieren\03_Evaluation_TTS\03_Testprotokoll\03_Testprotokoll.odt
CC-Lizenz	Creative Commons Namensnennung-Keine kommerzielle Nutzung-Weitergabe unter gleichen Bedingungen 2.5 Schweiz <a href="http://creativecommons.org/licenses/by-nc-sa/2.5/ch/">http://creativecommons.org/licenses/by-nc-sa/2.5/ch/</a>



# Inhaltsverzeichnis

<b>1</b>	<b>Einführung.....</b>	<b>6</b>
1.1	Testwerkzeuge.....	6
1.2	Zweck des Dokumentes.....	6
1.3	Zielsetzung.....	6
1.4	Referenzierte Dokumente.....	6
<b>2</b>	<b>Testplanung.....</b>	<b>7</b>
<b>3</b>	<b>Liste der Testfälle.....</b>	<b>8</b>
<b>4</b>	<b>Teststufenplan.....</b>	<b>11</b>
4.1	Systemanforderung.....	11
4.2	ITIL-Prozessabbildung.....	12
4.3	Systemumgebung.....	13
4.4	Erweiterbarkeit.....	13
4.5	Schnittstellen zu LDAP.....	13
<b>5</b>	<b>Testfälle.....</b>	<b>14</b>
5.1	Testfall Nummer 1: Webbasiert.....	14
5.2	Testfall Nummer 2: OpenSource.....	14
5.3	Testfall Nummer 3: Komplette Onlineverwaltung.....	15
5.4	Testfall Nummer 4: Community.....	15
5.5	Testfall Nummer 5: Usability.....	15
5.6	Testfall Nummer 6: Schnell.....	16
5.7	Testfall Nummer 7: Einfach.....	16
5.8	Testfall Nummer 8: Intuitiv.....	17
5.9	Testfall Nummer 9: klare, übersichtliche und logische Strukturierung.....	17
5.10	Testfall Nummer 10: Erfassung von Störungen.....	18
5.11	Testfall Nummer 11: Bearbeitende Personen eintragen.....	19
5.12	Testfall Nummer 12: Dauer der Bearbeitung.....	19
5.13	Testfall Nummer 13: Beschreibung der vorgenommenen Arbeit.....	20
5.14	Testfall Nummer 14: Prioritätsstufen (Externe Dringlichkeit/Interne Priorität).....	20
5.15	Testfall Nummer 15: Kategorien.....	21
5.16	Testfall Nummer 16: History der Incidents.....	21
5.17	Testfall Nummer 17: Knowledge-DB.....	22
5.18	Testfall Nummer 18: automatische RfC-Erfassung (Request for Change).....	22
5.19	Testfall Nummer 19: vordefinierte Templates für Incidents.....	23
5.20	Testfall Nummer 20: Statistische Auswertung.....	23
5.21	Testfall Nummer 21: Automatisches Backup.....	24
5.22	Testfall Nummer 22: logische Konfiguration.....	24
5.23	Testfall Nummer 23: vielfältige Konfigurationsmöglichkeit.....	25
5.24	Testfall Nummer 24: Erweiterbarkeitsmöglichkeit.....	25
5.25	Testfall Nummer 25: Scriptsprache (PHP, Perl, Bash, ... ).....	26
5.26	Testfall Nummer 26: Plugin für schöne Oberfläche und Style und Themes.....	26
5.27	Testfall Nummer 27: Benutzerinfos aus LDAP.....	27
5.28	Testfall Nummer 28: Hardwareinfos aus LDAP.....	27
5.29	Testfall Nummer 29: Anbindung ans Finanzsystem (Kostenstellen).....	28
5.30	User-Cases.....	28
5.30.1	User-Case Nummer 1: Problemmeldung.....	28
5.30.2	User-Case Nummer 2: Incident eröffnen.....	29
5.30.3	User-Case Nummer 3: Weiterleiten.....	29
5.30.4	User-Case Nummer 4: User Information.....	29
5.31	User-Case Diagramm.....	30

**6**      **Glossar.....31**

## **Abbildungsverzeichnis**

# 1 Einführung

## 1.1 Testwerkzeuge

Sind die Werkzeuge, die für die diversen Tests benutzt werden. Dazu gehören alle benötigten Server und Netzwerkkomponenten, welche zum Einsatz kommen.

Zum Testen stehen uns folgende Werkzeuge zur Verfügung:

- PuTTY
- Wireshark
- Texteditor vim
- SSH-Client ssh

## 1.2 Zweck des Dokumentes

Dieses Dokument stellt das Testprotokoll für den Aufbau des TTS der Firma Kuhumus AG dar. In diesem Dokument werden alle durchgeführten Tests und deren Ergebnisse ausführlich und detailliert beschrieben. Die Testabläufe sind klar beschrieben und dokumentiert. So wird klar ersichtlich, was in der Umgebung alles funktionstüchtig ist.

## 1.3 Zielsetzung

- Es geht darum, Fehler festzustellen, die Leistungen gemäss Anforderungsspezifikation beeinträchtigen.
- Dazu wird nach dem Black-Box Testverhalten gearbeitet.

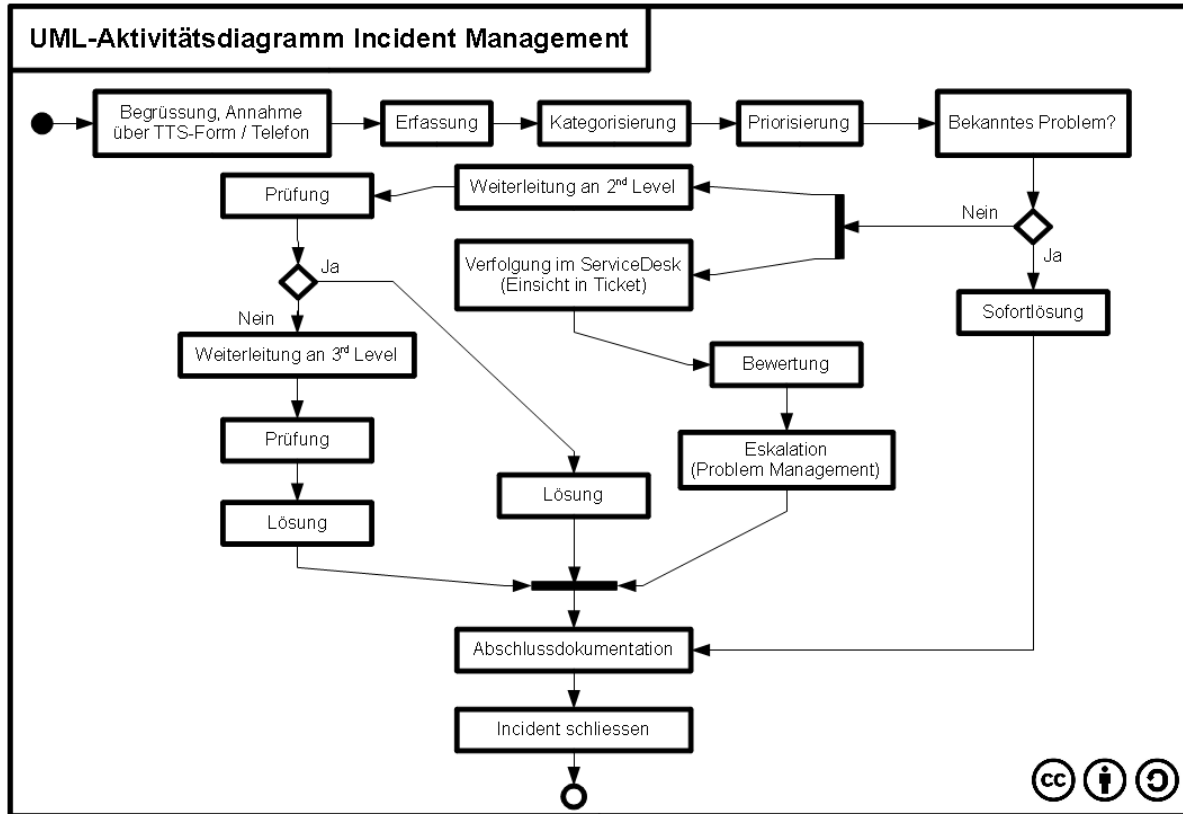
## 1.4 Referenzierte Dokumente

Referenzierte Dokumente werden in diesem Dokument nicht mit vollem Namen erwähnt, sondern mit den Zahlen in den eckigen Klammern aus der untenstehenden Liste deklariert. Alle Referenzen, welche im Text verwendet werden, sind in dieser Liste enthalten.

1. Anforderungsanalyse
2. Testkonzept

## 2 Testplanung

Grundsätzlich ist folgendes System zu erfüllen:



### 3 Liste der Testfälle

Nr.	Name	Beschreibung	K/M
1	Webbasiert	Es soll in den gängigen Browsern wie Firefox, Internet Explorer, Safari, Google Chrome, Mosaic und Opera lauffähig sein ohne jegliche grafischen Oberflächenfehler oder Verschiebungen aufzuzeigen.	Muss
2	OpenSource	Es soll frei sein, das heisst, wenn wir etwas weiterentwickeln, muss es erneut offengelegt werden und jeder muss so handeln.	Kann
3	Komplette Onlineverwaltung	Keine zusätzliche Software ist von Nöten, so garantiert man ein barrierefreie TTS-Verwaltung	Kann
4	Community	Hilfe soll man nicht nur durch eine potentiell teure Hotline erhalten, sondern auch in diversen Foren. Durch das ist nicht nur ein unbegrenztes Spektrum an freiwilligen Helfern im Internet zugänglich, es macht auch die Fehlersuche einfacher bzw. effizienter.	Kann
5	Usability	Das TTS soll einfach zu bedienen sein, Buttons sollen aussagekräftig beschrieben sein und Konfigurationsebenen sollten nicht allzu tief sein, das sonst schnell Verwirrung in einem Konfigurationsfensterjungle entsteht.	Kann
6	Schnell	Schnell muss die Software zwingend sein, durch langsame Software wird nur der Workflow unterbrochen und Arbeit wird kontraproduktiv.	Muss
7	Einfach	Es sind nicht 1000 Einstellungen nötig, dass eine Software gut ist, es soll alles vorhanden sein aber nicht unnötigt überlager aussehen.	Muss
8	Intuitiv	Grundsätzlich ergibt sich dies aus der Usability, denn je einfacher die Software ist, desto besser kann man diese bedienen. Ob es nun effektiv intuitiv ist, wird sich bei anfänglichen Tests zeigen.	Muss
9	klare, übersichtliche und logische Strukturierung	Durch ein ansprechendes Layout wird die Bedienung einfacher und mittels einer ordentlichen Oberflächen wirkt das ganze System professionell und ist schlussendlich produktiver im Einsatz.	Muss
10	Erfassung von Störungen	Mit dem Erfassen von Störungen erfüllt man eine Grundfunktion des Systems. Diese sogenannten Incidnet sollen in unserem TTS erfassbar sein.	Muss
11	Berbeitende Personen eintragen	Es müssen natürlich auch Personen eingetragen werden können, die für die Bearbeitung des Requests verantwortlich sind.	Muss
12	Dauer der Bearbeitung	Die Bearbeitungsdauer ist ein weiterer essentieller Punkt, der speziell für die Buchhaltung einen wichtigen Punkt darstellen kann, falls man Arbeitsstunden auf die Kostenstelle einer Abteilung verrechnen muss.	Muss
13	Beschreibung der vorgenommenen Arbeit	Ein IT-Mitarbeiter muss seine Arbeit auch in passender Form eintragen können. Die Beschreibung umfasst, was gemacht wurde und soll einfach sowie konzis sein.	Muss



14	Prioritätsstufen (Externe Dringlichkeit/Interne Priorität)	Da eine hohe Wahrscheinlichkeit besteht, dass es zu einem Zeitpunkt zu einer Anomalie an Incidents, müssen diese auch in einer Priorität geordnet werden.	Muss
15	Kategorien	Je nach Incident, wird ein anderer Experte prädestiniert für den Fall. Deshalb macht es Sinn Kategorien zu erstellen für eine optimale Verteilung der Incidents.	Muss
16	History der Incidents	Um Statistiken herstellen zu können und einzelne Objekte (PCs, Benutzer, etc.) nachzuverfolgen, ergibt sich die Idee einer History, die genau das beweckt.	Muss
17	Knowledge-DB	Mit einer Datenbank die alles Wissen zentralisiert wird die Problemlösung einfacher zugänglich und man kann auf Erfahrung zurückgreifen, statt Probleme mehrmals zu lösen.	Muss
18	automatische RfC-Erfassung (Request for Change)	Durch eine automatisierte Erfassung von RfCs wird es einfacher für dessens Verwaltung, dies kann anhand von Vorlagen geschehen.	Kann
19	vordefinierte Templates für Incidents	Auch für Incidents soll ein Template vorhanden sein, in welchem man schon bei der Erfassung festlegen kann, wie er Kategorisiert wird und kann dementsprechend schneller einsortiert werden.	Kann
20	Statistische Auswertung	Anhand der Datenbank kann man Auwertungen machen, welche Abteilung, Computer oder Benutzer am meisten Kosten verursachen. Diagramme sollen in Form von Balken und Kuchendiagrammen erstellbar sein.	Kann
21	Automatisches Backup	Zeitgesteuertes und vollautomatisiertes Backup soll im Falle eines Worst-case Szenarios die Verfügbarkeitsrate des Systems möglichst minimal verringern.	Muss
22	logische Konfiguration	Auch die Konfiguration des Grundsystems mit den Einstellungen soll einfach, sowie logisch aufgebaut sein, es soll keine verzwickte und nicht-verständliche Codemasse sein.	Muss
23	vielfältige Konfigurationsmöglichkeit	Es soll unsere Anforderungen ordnungsgemäss erfüllen aber ein zusätzliches, nützliches Feature ist natürlich nicht unwillkkommen.	Muss
24	Erweiterbarkeitsmöglichkeit	Durch zusätzliche Plugins sollen Funktionen optional erweitert werden können.	Muss
25	Scriptsprache (PHP, Perl, Bash, ...)	Das webbasierte TTS soll gängige Skriptsprachen beherrschen und dies auf der aktuellen Version, um eine hohe Sicherheit zu gewährleisten.	Muss
26	Plugin für schöne Oberfläche und Style und Themes	Auch soll uns eine optisch ansprechende Oberfläche begegnen auf der Webkonsole, bestenfalls sogar individuell anpassbar.	Kann
27	Benutzerinfos aus LDAP	Informationen zu Objekten im TTS sollen per LDAP ausgelesen werden, sodass die Objektverwaltung nicht doppelt geführt werden muss.	Muss
28	Hardwareinfos aus LDAP	Hardware Informationen zu Objekten im TTS sollen per LDAP ausgelesen werden, sodass die	Kann

		Objektverwaltung nicht doppelt geführt werden muss.	
29	Anbindung ans Finanzsystem (Kostenstellen)	Es soll, wenn möglich, mit einem Buchhaltungsprogramm verbindbar sein, so könnte die Buchhaltung ohne Probleme, Daten zu Incidents makellos verbuchen.	Kann

## 4 Teststufenplan

### 4.1 Systemanforderung

1	Webbasiert	Es soll in den gängigen Browsern wie Firefox, Internet Explorer, Safari, Google Chrome, Mosaic und Opera lauffähig sein ohne jegliche grafischen Oberflächenfehler oder Verschiebungen aufzuzeigen.	Muss
2	OpenSource	Es soll frei sein, das heisst, wenn wir etwas weiterentwickeln, muss es erneut offengelegt werden und jeder muss so handeln.	Kann
3	Komplette Onlineverwaltung	Keine zusätzliche Software ist von Nöten, so garantiert man ein barrierefreie TTS-Verwaltung	Kann
4	Community	Hilfe soll man nicht nur durch eine potentiell teure Hotline erhalten, sondern auch in diversen Foren. Durch das ist nicht nur ein unbegrenztes Spektrum an freiwilligen Helfern im Internet zugänglich, es macht auch die Fehlersuche einfacher bzw. effizienter.	Kann
5	Usability	Das TTS soll einfach zu bedienen sein, Buttons sollen aussagekräftig beschrieben sein und Konfigurationsebenen sollten nicht allzu tief sein, das sonst schnell Verwirrung in einem Konfigurationsfensterjungle entsteht.	Kann
6	Schnell	Schnell muss die Software zwingend sein, durch langsame Software wird nur der Workflow unterbrochen und Arbeit wird kontraproduktiv.	Muss
7	Einfach	Es sind nicht 1000 Einstellungen nötig, dass eine Software gut ist, es soll alles vorhanden sein aber nicht unnötigt überlager aussehen.	Muss
8	Intuitiv	Grundsätzlich ergibt sich dies aus der Usability, denn je einfacher die Software ist, desto besser kann man diese bedienen. Ob es nun effektiv intuitiv ist, wird sich bei anfänglichen Tests zeigen.	Muss
9	klare, übersichtliche und logische Strukturierung	Durch ein ansprechendes Layout wird die Bedienung einfacher und mittels einer ordentlichen Oberflächen wirkt das ganze System professionell und ist schlussendlich produktiver im Einsatz.	Muss

## 4.2 ITIL-Prozessabbildung

10	Erfassung von Störungen	Mit dem Erfassen von Störungen erfüllt man eine Grundfunktion des Systems. Diese sogenannten Incidents sollen in unserem TTS erfassbar sein.	Muss
11	Berbeitende Personen eintragen	Es müssen natürlich auch Personen eingetragen werden können, die für die Bearbeitung des Requests verantwortlich sind.	Muss
12	Dauer der Bearbeitung	Die Bearbeitungsdauer ist ein weiterer essentieller Punkt, der speziell für die Buchhaltung einen wichtigen Punkt darstellen kann, falls man Arbeitsstunden auf die Kostenstelle einer Abteilung verrechnen muss.	Muss
13	Beschreibung der vorgenommenen Arbeit	Ein IT-Mitarbeiter muss seine Arbeit auch in passender Form eintragen können. Die Beschreibung umfasst, was gemacht wurde und soll einfach sowie konzis sein.	Muss
14	Prioritätsstufen (Externe Dringlichkeit/Interne Priorität)	Da eine hohe Wahrscheinlichkeit besteht, dass es zu einem Zeitpunkt zu einer Anomalie an Incidents, müssen diese auch in einer Priorität geordnet werden.	Muss
15	Kategorien	Je nach Incident, wird ein anderer Experte prädestiniert für den Fall. Deshalb macht es Sinn Kategorien zu erstellen für eine optimale Verteilung der Incidents.	Muss
16	History der Incidents	Um Statistiken herstellen zu können und einzelne Objekte (PCs, Benutzer, etc.) nachzuverfolgen, ergibt sich die Idee einer History, die genau das beweckt.	Muss
17	Knowledge-DB	Mit einer Datenbank die alles Wissen zentralisiert wird die Problemlösung einfacher zugänglich und man kann auf Erfahrung zurückgreifen, statt Probleme mehrmals zu lösen.	Muss
18	automatische RfC-Erfassung (Request for Change)	Durch eine automatisierte Erfassung von RfCs wird es einfacher für dessens Verwaltung, dies kann anhand von Vorlagen geschehen.	Kann
19	vordefinierte Templates für Incidents	Auch für Incidents soll ein Template vorhanden sein, in welchem man schon bei der Erfassung festlegen kann, wie er Kategorisiert wird und kann dementsprechend schneller einsortiert werden.	Kann
20	Statistische Auswertung	Anhand der Datenbank kann man Auwertungen machen, welche Abteilung, Computer oder Benutzer am meisten Kosten verursachen. Diagramme sollen in Form von Balken und Kuchendiagrammen erstellbar sein.	Kann

## 4.3 Systemumgebung

Automatisches Backup	Zeitgesteuertes und vollautomatisiertes Backup soll im Falle eines Worst-case Szenarios die Verfügbarkeitsrate des Systems möglichst minimal verringern.	Muss
logische Konfiguration	Auch die Konfiguration des Grundsystems mit den Einstellungen soll einfach, sowie logisch aufgebaut sein, es soll keine verzwickte und nicht-verständliche Codemasse sein.	Muss
vielfältige Konfigurationsmöglichkeit	Es soll unsere Anforderungen ordnungsgemäss erfüllen aber ein zusätzliches, nützliches Feature ist natürlich nicht unwillkommen.	Muss

## 4.4 Erweiterbarkeit



24	Erweiterbarkeitsmöglichkeit	Durch zusätzliche Plugins sollen Funktionen optional erweitert werden können.	Muss
25	Scriptsprache (PHP, Perl, Bash, ...)	Das webbasierte TTS soll gängige Skriptsprachen beherrschen und dies auf der aktuellen Version, um eine hohe Sicherheit zu gewährleisten.	Muss
26	Plugin für schöne Oberfläche und Style und Themes	Auch soll uns eine optisch ansprechende Oberfläche begegnen auf der Webkonsole, bestenfalls sogar individuell anpassbar.	Kann

## 4.5 Schnittstellen zu LDAP



27	Benutzerinfos aus LDAP	Informationen zu Objekten im TTS sollen per LDAP ausgelesen werden, sodass die Objektverwaltung nicht doppelt geführt werden muss.	Muss
28	Hardwareinfos aus LDAP	Hardware Informationen zu Objekten im TTS sollen per LDAP ausgelesen werden, sodass die Objektverwaltung nicht doppelt geführt werden muss.	Kann
29	Anbindung ans Finanzsystem (Kostenstellen)	Es soll, wenn möglich, mit einem Buchhaltungsprogramm verbindbar sein, so könnte die Buchhaltung ohne Probleme, Daten zu Incidents makellos verbuchen.	Kann

## 5 Testfälle

### 5.1 Testfall Nummer 1: Webbasiert

<b>Nummer</b>	1
<b>Name</b>	Webbasiert
<b>Beschreibung</b>	Es soll in den gängigen Browsern wie Firefox, Internet Explorer, Safari, Google Chrome, Mosaic und Opera lauffähig sein ohne jegliche grafischen Oberflächenfehler oder Verschiebungen aufzuzeigen.
<b>Teststufenplan</b>	Systemanforderung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• TTS Installiert</li> <li>• Webserver</li> </ul>
<b>Testablauf</b>	TTS öffnen
<b>Erwartetes Ergebnis</b>	TTS muss geöffnet sein
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	



### 5.2 Testfall Nummer 2: OpenSource

<b>Nummer</b>	2
<b>Name</b>	OpenSource
<b>Beschreibung</b>	Es soll frei sein, das heisst, wenn wir etwas weiterentwickeln, muss es erneut offengelegt werden und jeder muss so handeln.
<b>Teststufenplan</b>	Systemanforderung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• Keine</li> </ul>
<b>Testablauf</b>	Lizenz beachten
<b>Erwartetes Ergebnis</b>	Lizenz unter einer OpenSource-Lizenz (z.B. GPL v3)
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

### 5.3 Testfall Nummer 3: Komplette Onlineverwaltung



<b>Nummer</b>	3
<b>Name</b>	OpenSource
<b>Beschreibung</b>	Keine zusätzliche Software ist von Nöten, so garantiert man ein barrierefreie TTS-Verwaltung
<b>Teststufenplan</b>	Systemanforderung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• TTS Installiert</li> <li>• Webserver</li> </ul>
<b>Testablauf</b>	Alles kann Online verwaltet werden
<b>Erwartetes Ergebnis</b>	Alles kann Online verwaltet werdenB. GPL v3)
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

### 5.4 Testfall Nummer 4: Community



<b>Nummer</b>	4
<b>Name</b>	Community
<b>Beschreibung</b>	Hilfe soll man nicht nur durch eine potentiell teure Hotline erhalten, sondern auch in diversen Foren. Durch das ist nicht nur ein unbegrenztes Spektrum an freiwilligen Helfern im Internet zugänglich, es macht auch die Fehlersuche einfacher bzw. effizienter.
<b>Teststufenplan</b>	Systemanforderung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• Internet</li> </ul>
<b>Testablauf</b>	Google konsultieren; Wikipedia
<b>Erwartetes Ergebnis</b>	Foren, Mailinglisten, Vereine, UerGroups, ...
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

### 5.5 Testfall Nummer 5: Usability

<b>Nummer</b>	5
<b>Name</b>	Usability

<b>Beschreibung</b>	Das TTS soll einfach zu bedienen sein, Buttons sollen aussagekräftig beschrieben sein und Konfigurationsebenen sollten nicht allzu tief sein, das sonst schnell Verwirrung in einem Konfigurationsfensterjungle entsteht.
<b>Teststufenplan</b>	Systemanforderung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• TTS installiert</li> </ul>
<b>Testablauf</b>	Eine Mutter sollte das Tool testen.
<b>Erwartetes Ergebnis</b>	Die Mutter konnte einen einfachen Incident eröffnen, ohne grosse Erklärungen zu geben.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

## 5.6 Testfall Nummer 6: Schnell

<b>Nummer</b>	6
<b>Name</b>	Schnell
<b>Beschreibung</b>	Schnell muss die Software zwingend sein, durch langsame Software wird nur der Workflow unterbrochen und Arbeit wird kontraproduktiv.
<b>Teststufenplan</b>	Systemanforderung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• TTS Installiert</li> </ul>
<b>Testablauf</b>	Zwischen einzelnen Seiten hin- und herwechseln.
<b>Erwartetes Ergebnis</b>	Keine Wartezeiten.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

## 5.7 Testfall Nummer 7: Einfach

<b>Nummer</b>	7
<b>Name</b>	Einfach
<b>Beschreibung</b>	Es sind nicht 1000 Einstellungen nötig, dass eine Software gut ist, es soll alles vorhanden sein aber nicht unnötigt überlager aussehen.
<b>Teststufenplan</b>	Systemanforderung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• TTS Installiert</li> </ul>
<b>Testablauf</b>	Beachten, ob etwas unnötiges im Wege steht



<b>Erwartetes Ergebnis</b>	Es sollte nicht unnötiges im Wege stehen.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

## 5.8 Testfall Nummer 8: Intuitiv



<b>Nummer</b>	8
<b>Name</b>	Intuitiv
<b>Beschreibung</b>	Grundsätzlich ergibt sich dies aus der Usability, denn je einfacher die Software ist, desto besser kann man diese bedienen. Ob es nun effektiv intuitiv ist, wird sich bei anfänglichen Tests zeigen.
<b>Teststufenplan</b>	Systemanforderung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Das Tool bedienen.
<b>Erwartetes Ergebnis</b>	Nach einer gewissen einarbeitungszeit (was bei jedem grösseren Tool nötig ist), soll das Tool ohne Probleme bedient werden können.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

## 5.9 Testfall Nummer 9: klare, übersichtliche und logische Strukturierung



<b>Nummer</b>	9
<b>Name</b>	klare, übersichtliche und logische Strukturierung
<b>Beschreibung</b>	Durch ein ansprechendes Layout wird die Bedienung einfacher und mittels einer ordentlichen Oberflächen wirkt das ganze System professionell und ist schlussendlich produktiver im Einsatz.
<b>Teststufenplan</b>	Systemanforderung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Menüeinträge sollen schnell gefunden werden
<b>Erwartetes Ergebnis</b>	Die Menüeinträge werden schnell gefunden.

<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

## 5.10 Testfall Nummer 10: Erfassung von Störungen

<b>Nummer</b>	10
<b>Name</b>	Erfassen von Störungen
<b>Beschreibung</b>	Mit dem Erfassen von Störungen erfüllt man eine Grundfunktion des Systems. Diese sogenannten Incidnet sollen in unserem TTS erfassbar sein.
<b>Teststufenplan</b>	ITIL-Prozessabbildung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• TTS Installiert</li> </ul>
<b>Testablauf</b>	Eine Störung erfassen
<b>Erwartetes Ergebnis</b>	Die Störung ist wieder abrufbar und bleibt gespeichert.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	


## 5.11 Testfall Nummer 11: Bearbeitende Personen eintragen

<b>Nummer</b>	11
<b>Name</b>	Bearbeitende Person eintragen
<b>Beschreibung</b>	Es müssen natürlich auch Personen eingetragen werden können, die für die Bearbeitung des Requests verantwortlich sind.
<b>Teststufenplan</b>	ITIL-Prozessabbildung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Ein Incident einer Person zuordnen
<b>Erwartetes Ergebnis</b>	Die zugeordnete Person kann das Ticket abrufen. Die Person kann danach das Ticket bearbeiten.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	



## 5.12 Testfall Nummer 12: Dauer der Bearbeitung

<b>Nummer</b>	12
<b>Name</b>	Dauer der Bearbeitung
<b>Beschreibung</b>	Die Bearbeitungsdauer ist ein weiterer essentieller Punkt, der speziell für die Buchhaltung einen wichtigen Punkt darstellen kann, falls man Arbeitsstunden auf die Kostenstelle einer Abteilung verrechnen muss.
<b>Teststufenplan</b>	ITIL-Prozessabbildung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Feld 'Dauer' vorhanden
<b>Erwartetes Ergebnis</b>	Das Feld 'Dauer' ist vorhanden.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	


### 5.13 Testfall Nummer 13: Beschreibung der vorgenommenen Arbeit

<b>Nummer</b>	13
<b>Name</b>	Beschreibung der vorgenommenen Arbeit
<b>Beschreibung</b>	Ein IT-Mitarbeiter muss seine Arbeit auch in passender Form eintragen können. Die Beschreibung umfasst, was gemacht wurde und soll einfach sowie konzis sein.
<b>Teststufenplan</b>	ITIL-Prozessabbildung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Ein Notizfeld ist sichtbar
<b>Erwartetes Ergebnis</b>	Das Notizfeld wird schnell gefunden.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	



### 5.14 Testfall Nummer 14: Prioritätsstufen (Externe Dringlichkeit/Interne Priorität)

<b>Nummer</b>	14
<b>Name</b>	Prioritätsstufen (Externe Dringlichkeit/Interne Priorität)
<b>Beschreibung</b>	Da eine hohe Wahrscheinlichkeit besteht, dass es zu einem Zeitpunkt zu einer Anomalie an Incidents, müssen diese auch in einer Priorität geordnet werden.
<b>Teststufenplan</b>	ITIL-Prozessabbildung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Felder sind vorhanden.
<b>Erwartetes Ergebnis</b>	Felder sind vorhanden.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	



## 5.15 Testfall Nummer 15: Kategorien

<b>Nummer</b>	15
<b>Name</b>	Kategorien
<b>Beschreibung</b>	Je nach Incident, wird ein anderer Experte prädestiniert für den Fall. Deshalb macht es Sinn Kategorien zu erstellen für eine optimale Verteilung der Incidents.
<b>Teststufenplan</b>	ITIL-Prozessabbildung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Felder sind vorhanden.
<b>Erwartetes Ergebnis</b>	Felder sind vorhanden.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	



## 5.16 Testfall Nummer 16: History der Incidents

<b>Nummer</b>	16
<b>Name</b>	History der
<b>Beschreibung</b>	Um Statistiken herstellen zu können und einzelne Objekte (PCs, Benutzer, etc.) nachzuverfolgen, ergibt sich die Idee einer History, die genau das beweckt.
<b>Teststufenplan</b>	ITIL-Prozessabbildung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Es wird angezeigt, was schon am Ticket gemacht wurde.
<b>Erwartetes Ergebnis</b>	Das Feld ist vorhanden.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	



## 5.17 Testfall Nummer 17: Knowledge-DB

<b>Nummer</b>	17
<b>Name</b>	Knowledge-DB
<b>Beschreibung</b>	Mit einer Datenbank die alles Wissen zentralisiert wird die Problemlösung einfacher zugänglich und man kann auf Erfahrung zurückgreifen, statt Probleme mehrmals zu lösen.
<b>Teststufenplan</b>	ITIL-Prozessabbildung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Es können neue Einträge in die Knowledge-DB hinzugefügt werden. In der Knowledge-DB sind mögliche Lösungen vorhanden.
<b>Erwartetes Ergebnis</b>	Bei einem neuen Incident kann man die Knowledge-DB durchsuchen und so schneller zu einer Lösung kommen.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	
	RT FAQ Manager

## 5.18 Testfall Nummer 18: automatische RfC-Erfassung (Request for Change)

<b>Nummer</b>	18
<b>Name</b>	Automatische RfC-Erfassung
<b>Beschreibung</b>	Durch eine automatisierte Erfassung von RfCs wird es einfacher für dessens Verwaltung, dies kann anhand von Vorlagen geschehen.
<b>Teststufenplan</b>	ITIL-Prozessabbildung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Aus dem Ticket heraus kann automatisch ein RfC erstellt werden.
<b>Erwartetes Ergebnis</b>	Der RfC ist danach ersichtlich.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	



## 5.19 Testfall Nummer 19: vordefinierte Templates für Incidents

<b>Nummer</b>	19
<b>Name</b>	Vordefinierte Templates für Incidents
<b>Beschreibung</b>	Auch für Incidents soll ein Template vorhanden sein, in welchem man schon bei der Erfassung festlegen kann, wie er Kategorisiert wird und kann dementsprechend schneller einsortiert werden.
<b>Teststufenplan</b>	ITIL-Prozessabbildung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Häufig verwendete Incidents sind als Vorlage verwendbar.
<b>Erwartetes Ergebnis</b>	Bei einem neuen Incident kann die Vorlage ausgewählt werden.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	


## 5.20 Testfall Nummer 20: Statistische Auswertung

<b>Nummer</b>	20
<b>Name</b>	Statistische Auswertung
<b>Beschreibung</b>	Anhand der Datenbank kann man Auswertungen machen, welche Abteilung, Computer oder Benutzer am meisten Kosten verursachen. Diagramme sollen in Form von Balken und Kuchendiagrammen erstellbar sein.
<b>Teststufenplan</b>	ITIL-Prozessabbildung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Es ist ein Menüpunkt für die Auswertung vorhanden.
<b>Erwartetes Ergebnis</b>	Der Menüpunkt ist vorhanden.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

## 5.21 Testfall Nummer 21: Automatisches Backup



<b>Nummer</b>	21
<b>Name</b>	Automatisches Backup
<b>Beschreibung</b>	Zeitgesteuertes und vollautomatisiertes Backup soll im Falle eines Worst-case Szenarios die Verfügbarkeitsrate des Systems möglichst minimal verringern.
<b>Teststufenplan</b>	Systemumgebung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Ein Backup soll automatisch jeden Tag erstellt werden. (evtl mit 'mysqldump' erstellen)
<b>Erwartetes Ergebnis</b>	Das Backup wird jeden Tag erstellt.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

## 5.22 Testfall Nummer 22: logische Konfiguration

<b>Nummer</b>	22
<b>Name</b>	Logische Konfiguration
<b>Beschreibung</b>	Auch die Konfiguration des Grundsystems mit den Einstellungen soll einfach, sowie logisch aufgebaut sein, es soll keine verzwickte und nicht-verständliche Codemasse sein.
<b>Teststufenplan</b>	Systemumgebung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Die Konfiguration erscheint dem Administrator „logisch“
<b>Erwartetes Ergebnis</b>	Der Admin ist in der Lage das System zu administrieren
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	





## 5.23 Testfall Nummer 23: vielfältige Konfigurationsmöglichkeit

<b>Nummer</b>	23
<b>Name</b>	Vielfältige Konfigurationsmöglichkeit
<b>Beschreibung</b>	Es soll unsere Anforderungen ordnungsgemäss erfüllen aber ein zusätzliches, nützliches Feature ist natürlich nicht unwillkommen.
<b>Teststufenplan</b>	Systemumgebung
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Es sind viele Optionen ersichtlich.
<b>Erwartetes Ergebnis</b>	Die Optionen können eingestellt werden.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	



## 5.24 Testfall Nummer 24: Erweiterbarkeitsmöglichkeit

<b>Nummer</b>	24
<b>Name</b>	Erweiterbarkeitsmöglichkeit
<b>Beschreibung</b>	Durch zusätzliche Plugins sollen Funktionen optional erweitert werden können.
<b>Teststufenplan</b>	Erweiterbarkeit
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Das System kann mit neuen Modulen erweitert werden.
<b>Erwartetes Ergebnis</b>	Die Module sind lauffähig.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	



## 5.25 Testfall Nummer 25: Scriptsprache (PHP, Perl, Bash, ...)

<b>Nummer</b>	25
<b>Name</b>	Scriptsprache
<b>Beschreibung</b>	Das webbasierte TTS soll gängige Skriptsprachen beherrschen und dies auf der aktuellen Version, um eine hohe Sicherheit zu gewährleisten.
<b>Teststufenplan</b>	Erweiterbarkeit
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Es können selber einfache Module programmiert werden.
<b>Erwartetes Ergebnis</b>	Die Module sind lauffähig
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	



## 5.26 Testfall Nummer 26: Plugin für schöne Oberfläche und Style und Themes

<b>Nummer</b>	26
<b>Name</b>	Plugin für schöne Oberfläche und Style und Themes
<b>Beschreibung</b>	Auch soll uns eine optisch ansprechende Oberfläche begegnen auf der Webkonsole, bestenfalls sogar individuell anpassbar.
<b>Teststufenplan</b>	Erweiterbarkeit
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Man kann mehrere Themes auswählen.
<b>Erwartetes Ergebnis</b>	Die Themes werden aktiv.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	



## 5.27 Testfall Nummer 27: Benutzerinfos aus LDAP

<b>Nummer</b>	27
<b>Name</b>	Plugin für schöne Oberfläche und Style und Themes
<b>Beschreibung</b>	Informationen zu Objekten im TTS sollen per LDAP ausgelesen werden, sodass die Objektverwaltung nicht doppelt geführt werden muss.
<b>Teststufenplan</b>	Schnittstellen zu LDAP
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Anbindung ans LDAP für Userinfos: Username, Telefonnummer. Diese Informationen sind im LDAP ersichtlich.
<b>Erwartetes Ergebnis</b>	Der User wird angegeben und die Telefonnummer automatisch vom TTS ausgefüllt.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

## 5.28 Testfall Nummer 28: Hardwareinfos aus LDAP



<b>Nummer</b>	28
<b>Name</b>	HW-Infos aus LDAP
<b>Beschreibung</b>	Hardware Informationen zu Objekten im TTS sollen per LDAP ausgelesen werden, sodass die Objektverwaltung nicht doppelt geführt werden muss.
<b>Teststufenplan</b>	Schnittstellen zu LDAP
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Anbindung ans LDAP für Userinfos: Welcher Computer verwendet der User?
<b>Erwartetes Ergebnis</b>	Das Computermodell wird angezeigt.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

## 5.29 Testfall Nummer 29: Anbindung ans Finanzsystem (Kostenstellen)



<b>Nummer</b>	29
<b>Name</b>	Anbindung ans Finanzsystem (Kostenstellen)
<b>Beschreibung</b>	Es soll, wenn möglich, mit einem Buchhaltungsprogramm verbindbar sein, so könnte die Buchhaltung ohne Probleme, Daten zu Incidents makellos verbuchen.
<b>Teststufenplan</b>	Schnittstellen zu LDAP
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>TTS Installiert</li> </ul>
<b>Testablauf</b>	Zu einem User werden Infos angezeigt wie z.B. Von welcher Firma er ist (CRM-Programmanbindung)
<b>Erwartetes Ergebnis</b>	Die Information ist schnell abrufbar.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

## 5.30 User-Cases



### 5.30.1 User-Case Nummer 1: Problemmeldung

<b>Auslöser</b>	User
<b>Voraussetzung</b>	Störung der Arbeit und computertechnisch bedingtes Problem
<b>Funktionsablauf</b>	Das Problem wird an die verantwortliche Stelle geleitet
<b>Endzustand</b>	Durch die erste Supportebene wird ein Incident eröffnet.
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	



### 5.30.2 User-Case Nummer 2: Incident eröffnen

Auslöser	1 <sup>st</sup> Level Supporter
Voraussetzung	Problemmeldung eines Users
Funktionsablauf	Eröffnung des Incidents
Endzustand	Ein neuer Incident ist eröffnet und wird in die DB aufgenommen
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

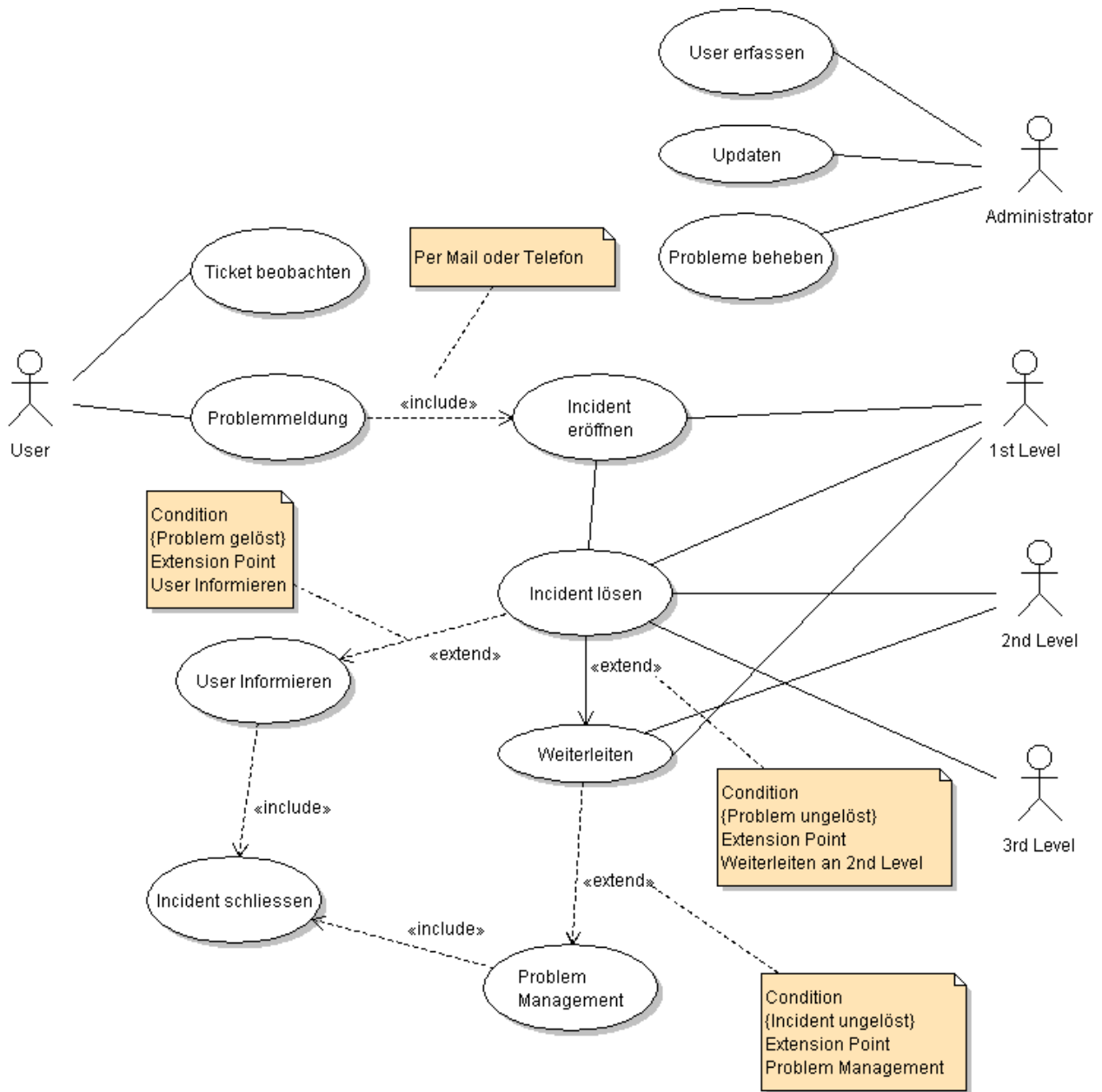
### 5.30.3 User-Case Nummer 3: Weiterleiten

Auslöser	Level n Supporter
Voraussetzung	Problem wurde durch untere Instanz bereits versucht zu lösen
Funktionsablauf	Weiterleitung des Incidents
Endzustand	Der Incident ist an der nächsten Stelle und wird dort bearbeitet
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

### 5.30.4 User-Case Nummer 4: User Information

Auslöser	Level n Supporter
Voraussetzung	Problem wurde gelöst und der Incident geschlossen
Funktionsablauf	Der geöffnete Incident konnte von einem Mitarbeiter gelöst werden
Endzustand	Der User kann wieder arbeiten, ohne jegliche Probleme
<b>OTRS::TISM::</b>	
<b>Best Practical: RT Request Tracker</b>	

### 5.31 User-Case Diagramm



## 6 Glossar

Folgende Begriffe sind für einen Normalsterblichen genauer erklärt:

<b>Begriff</b>	<b>Erklärung</b>
Glossar	Begriffserklärungstabelle