Lotto Anwendungssystemen

**Übungsprojekt**

11.12.2018

IT-Umschulung FIAE

A. Wett

# Inhaltsverzeichnis

[Inhaltsverzeichnis 1](#_Toc398118298)

[Firmen Beschreibung 3](#_Toc398118299)

[Problemstellung 3](#_Toc398118300)

[Erklärung Fehler! Textmarke nicht definiert.](#_Toc398118301)

[Lastenheft SemOrg 4](#_Toc398118302)

[1. Visionen und Ziele 4](#_Toc398118303)

[2. Rahmenbedingungen 5](#_Toc398118304)

[3. Kontext und Überblick 5](#_Toc398118305)

[4. Funktionale Anforderungen 5](#_Toc398118306)

[5. Qualitätsanforderungen 6](#_Toc398118307)

[Aufgabenstellung 7](#_Toc398118308)

[Pflichtenheft 8](#_Toc398118309)

[1. Visionen und Ziele 8](#_Toc398118310)

[2. Rahmenbedingungen 8](#_Toc398118311)

[3. Kontext und Überblick 8](#_Toc398118312)

[4. Funktionale Anforderungen 8](#_Toc398118313)

[5. Qualitätsanforderungen 9](#_Toc398118314)

[6. Abnahmekriterien 10](#_Toc398118315)

[Qualitätskonzept 12](#_Toc398118316)

[1. Tauglichkeit (Capability) 12](#_Toc398118317)

[2. Benutzerfreundlichkeit(Usability) 12](#_Toc398118318)

[3. Leistung (Performance) 12](#_Toc398118319)

[4. Korrektheit (Correctness) 12](#_Toc398118320)

[5. Verlässlichkeit/Ausfallsicherheit (Reliability) 12](#_Toc398118321)

[6. Wartbarkeit (Maintainability) 12](#_Toc398118322)

[7. Portierbarkeit (Portability) 12](#_Toc398118323)

[Testkonzept 14](#_Toc398118324)

[1. Komponententests 14](#_Toc398118325)

[1.1. Passwort 14](#_Toc398118326)

[1.2. Daten Einlesen 14](#_Toc398118327)

[1.3. Grafiken einbinden 15](#_Toc398118328)

[2. Integrationstests 15](#_Toc398118329)

[3. Systemtests 15](#_Toc398118330)

[4. Abnahmetest 16](#_Toc398118331)

[Anhänge 18](#_Toc398118332)

[1. USE CASE Allgemein grob 18](#_Toc398118333)

[2. USE CASE System Umwelt 18](#_Toc398118334)

[3. USE CASE Kundenbereich Darstellung 18](#_Toc398118335)

[4. USE CASE Seminarbereich 18](#_Toc398118336)

[5. ER Modell einfach 18](#_Toc398118337)

[6. ER Modell Ausführlich 18](#_Toc398118338)

[7. Klassendiagramm SemOrg 19](#_Toc398118339)

[8. Zustandsdiagramm Veranstaltungsbereich 19](#_Toc398118340)

[9. Sequenzdiagramm erstellen TN Liste 19](#_Toc398118341)

[10. Sequenzdiagramm Alternative Kundenbuchung 19](#_Toc398118342)

[Glossar 20](#_Toc398118343)

[Abkürzungsverzeichnis 21](#_Toc398118344)

# Firmen Beschreibung

**EasyGamBling**

Was macht die EasyGamBling

Anbieter von Gewinnspielsystemen. Als Unternehmen vertreibt Sie Software für Spielautomaten in verschiedenen Bereichen. Die Software wird als Lizensiertes Produkt an die Automatenaufsteller vertrieben.

# Problemstellung

Die Software wird Inhouse an mehreren Standorten entwickelt. Hauptsächlich in der Programmiersprache Java. Aufgrund der universellen Nutzbarkeit der Sprache in verschiedenen Systemen. Die entwickelte Software bleibt Eigentum der EasyGambling.

Da immer mehr Anbieter in diesem Bereich auf den Markt drängen möchte EasyGambling jetzt auch im Lotteriegeschäft ein Standbein aufbauen. Hierzu soll innerhalb eines Projektes eine Lottosimulation entwickelt werden, welche nicht nur eine automatiserte Generierung möglicher Gewinnzahlen, sondern auch die Ermittlung möglicher Gewinne erfolgen soll.

Regeln:

Gesamt Reihen: 5

Gesamt Zahlen Menge: 49

Zu verwendende Zahlen: genau 6

Gewinn Ermittlung:

6 Richtige einsatz \* faktor \* 6

5 Richtige einsatz \* faktor \* 5

4 Richtige einsatz \* faktor \* 4

3Richtige einsatz \* faktor \* 3

2 Richtige einsatz \* faktor \* 2

1 Richtige einsatz \* faktor

Einsatz ist frei wählbar

Faktor ist 0,75

Hierzu wurde folgendes Lastenheft ausgearbeitet.

# Lastenheft SemOrg

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Autor | Quelle | Status | Datum | Kommentar |
| 0.1 | M. Mustermann | Geschäftsführer | In Bearbeitung | 02.09.2014 | --------------- |

**Managementattribute:**

**Priorität aus Auftraggebersicht** = {hoch, **mittel**, niedrig}

**Priorität aus Auftragnehmersicht** = {hoch, **mittel**, niedrig}

**Stabilität der Anforderungen** = {fest, **gefestigt**, volatil}

**Kritikalität der Anforderungen** = {hoch, mittel, **niedrig**, keine}

**Entwicklungsrisiko der Anforderungen** = {hoch, mittel, **niedrig**}

## Visionen und Ziele

**/LV10/** Es soll eine weitere Produktgruppe im Bereich der Lotteriespiele dem Portfolio hinzugefügt werden

**/LV20/** Es sollen neue Kundengruppen angesprochen werden die außerhalb des Automatengeschäftes angesiedelt sind

**/LZ10/** Die neu generierten Kunden sollen das Lotteriesystem in Onlineplattformen anbieten können.

## Rahmenbedingungen

**/LR10/** Easy49 ist ein Gewinnspiel orientiertes System.

**/LR20/** Zielgruppe sind Nutzer die dem Gesetz nach volljährig sind

## Kontext und Überblick

**/LK10/**Das System soll eine Verfügbarkeit von 24/7 haben.

**/LK20/** Das System muss generierte Zahlen dem Benutzer zuweisen und die Gewinne Nutzer bezogen auswerten können.

## Funktionale Anforderungen

**/LF10/** Das System soll aus den verfügbaren Zahlen 6 auswählen und diese dem Benutzer zuweisen

**/LF20/** Das System soll dem Nutzer die Möglichkeit bieten zu entscheiden ob er eigene Zahlen eingibt, oder sich zufällige Zahlen generieren lässt.

**/LF30/** Das System muss 6 Gewinnzahlen generieren

**/LF40/** Das System muss entsprechend dem Faktor, unter Berücksichtigung des Einsatzes, und der Anzahl der vorhandenen Nutzer Zahlen den Gewinn ermitteln

**/LF50/** Das System muss den Einsatz speichern und entsprechend dem Gewinn verfügbares Guthaben erweitern.

**/LF60/** Das System soll eine Möglichkeit bieten aus den maximalen Reihen entweder alle oder weniger, aber mindestens eine auszuwählen.

**/LF70/** Das System muss den Nutzer nach dem Einsatz fragen.

## Qualitätsanforderungen

/LQ10/ Alle Reaktionszeiten auf Benutzeraktionen müssen unter 5 Sekunden liegen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Systemqualität | Sehr gut | gut | normal | Nicht relevant |
| Funktionalität | **X** |  |  |  |
| Zuverlässigkeit | **X** |  |  |  |
| Benutzbarkeit |  | **X** |  |  |
| Effizienz |  |  | **X** |  |
| Wartbarkeit |  |  | **X** |  |
| Portabilität |  |  |  | **X** |

# Aufgabenstellung

* Erstellen Sie basierend auf den im Lastenheft genannten Anforderungen ein Pflichtenheft. Und nach Erstellung des Pflichtenheftes die entsprechend den Phasen zugehörigen Dokumente.
* Das Klassendiagramm wird mit allen zusammen erstellt und ist bindend für die Umsetzung
* Die Vorgänge der Entwicklung sollen mit geeigneten Modellen der UML visualisiert werden.
* Die Ausarbeitung der notwendigen UML Diagramme wird Standortweise verteilt und der gesamten Projektdokumentation angehängt
* Dauer des Gesamten Projektes : 36 Stunden (12.12. –18.12.)
* In Absprache werden die jeweiligen Klassen Standortweise bearbeitet und am Klassendiagramm orientiert. Also Benennung der Methoden und Variablen wie im Klassendiagramm festgelegt.
* Die notwendigen Umsetzungsthematiken werden abwechselnd gemeinsam besprochen und dann Standortweise umgesetzt.
* Stellen Sie eigene Regeln für die Kommunikation untereinander auf und jeder sollte sich verpflichten sich an diese zu halten.

# Pflichtenheft

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Autor | Quelle | Status | Datum | Kommentar |
| 0.1 | M. Mustermann | Geschäftsführer  Teachware | In Bearbeitung | 04.09.2014 | Verfeinerung des Lastenheftes **VO.1** |

*Alle Anforderungen im Pflichtenheft, die einen Bezug zu einer Anforderung im Lastenheft haben, müssen eine entsprechende Referenz auf das Lastenheft haben. Da das Pflichtenheft eine Verfeinerung und Erweiterung des Lastenhefts darstellt, gibt es neue Anforderungen, von denen kein Bezug zum Lastenheft hergestellt werden kann. Es gibt dann keine entsprechende Referenz.*

## Visionen und Ziele

***Spezifikationen EasyGamBling***

/V10 (LV10) Es soll eine weitere Produktgruppe im Bereich der Lotteriespiele dem Portfolio hinzugefügt werden

/V20 (LV20) Es sollen neue Kundengruppen angesprochen werden die außerhalb des Automatengeschäftes angesiedelt sind

/Z10 (LZ10) Die neu generierten Kunden sollen das Lotteriesystem in Onlineplattformen anbieten können.

## Rahmenbedingungen

*Beschreibung der organisatorischen Rahmenbedingungen: Anwendungsbereiche, Zielgruppen, Betriebsbedingungen*

*/R10/ Anwendungsbereich 1 des Systems usw.*

*/R20/ Zielgruppe 1 des Systems usw.*

*/R30/ Physikalische Umgebung des Systems*

*/R40/ Tägliche Arbeitszeit des Systems*

*/R50/ Ständige Beobachtung des Systems durch Bediener oder unbeaufsichtigter Betrieb*

*Bei der technischen Produktumgebung sind folgende Festlegungen zu treffen:*

*/R60/ Eingesetzte Software auf der Zielmaschine*

*/R70/ Eingesetzte Hardware (Komponenten) einschl. Konfiguration auf der Zielmaschine*

*/R80/ Organisatorische Randbedingungen und Voraussetzungen*

*Bei den Anforderungen an die Entwicklungsumgebung sind folgende Festlegungen zu treffen*

*/R90/ Software auf dem Entwicklungssystem*

*/R100/ Hardware auf dem Entwicklungssystem*

*/R110/ Orgware des Entwicklungssystems*

*/R120/Entwicklungsschnittstellen*

***Spezifikationen EasyGamBling***

/R10 (LR10) Easy49 ist ein Gewinnspiel orientiertes System.

/R20 (LR20) Zielgruppe sind Nutzer die dem Gesetz nach volljährig sind

## Kontext und Überblick

Festlegung der relevanten Systemumgebung (Kontext) und Überblick über das System:

/K10/ Kontext 1 des Systems usw.

***Spezifikationen EasyGamBling***

/K10 (LK10) Das System soll eine Verfügbarkeit von 24/7 haben.

/K20 (LK20) Das System muss generierte Zahlen dem Benutzer zuweisen und die Gewinne Nutzer bezogen auswerten können.

## Funktionale Anforderungen

*Die Kernfunktionalität des Systems ist aus Auftraggebersicht auf oberster Abstraktionsebene zu beschreiben. Auf Detailbeschreibungen ist zu verzichten.*

*/F10/ Funktion 1 des Systems usw.*

*Wenn bereits möglich, sind die funktionalen Anforderungen nach Statik, Dynamik und Logik zu gruppieren. Erfolgt die Spezifikation natürlich sprachlich, dann sollten* ***Sprachschablonen*** *verwendet werden.*

***Spezifikationen EasyGamBling***

/F10 (LF20) Das System soll dem Nutzer die Möglichkeit bieten zu entscheiden ob er eigene Zahlen eingibt, oder sich zufällige Zahlen generieren lässt.

/F20 (LF10) Das System soll aus den verfügbaren Zahlen 6 auswählen und diese dem Benutzer zuweisen

/F21 (LF10) Die verfügbaren Zahlen sind 1 - 49, wobei die 0 ausgeschlossen werden muss und keine der zugewiesenen/ausgewählten Zahlen doppelt vorkommen darf.

/F30 (LF30) Das System muss 6 Gewinnzahlen generieren

/F31 (LF30) Die verfügbaren Zahlen sind 1 - 49, wobei die 0 ausgeschlossen werden muss und keine der generierten Zahlen doppelt vorkommen darf.

/F40 (LF40) Das System muss entsprechend dem Faktor, unter Berücksichtigung des Einsatzes, und der Anzahl der vorhandenen Nutzer Zahlen den Gewinn ermitteln

/F41 (LF40) Aus den Einsätzen der Nutzer wird ein Gewinntopf generiert, aus dem die Gewinne ausgezahlt werden.

/F50 (LF50) Das System muss den Einsatz speichern und entsprechend dem Gewinn verfügbares Guthaben erweitern.

/F60 (LF60) Das System soll eine Möglichkeit bieten aus den 5 verfügbaren Reihen entweder alle oder aber mindestens eine auszuwählen.

/F70 (LF70) Das System darf dem Benutzer die Auswahl/Vorgabe der Zahlen, sowie die Auswahl der Reihen erst nach Eingabe eines Einsatzes ermöglichen.

## Qualitätsanforderungen

*Es sollte anhand einer Tabelle eine verfeinerte Qualitätszielbestimmung für das System vorgenommen werden. Es eignen sich dazu die Qualitätsmerkmale der ISO / IEC 9126-1 (in separater pdf Datei).*

*Einzelne Qualitätsanforderungen können unter Bezug auf die ISO/IEC9126-1 wie folgt festgelegt werden:*

*/QFS10/ Qualitätsanforderungen zur Funktionalität ( Sicherheit) des Systems*

*/QBE10/ Qualitätsanforderungen zur Benutzbarkeit(Erlernbarkeit) des Systems usw.*

***Spezifikationen EasyGamBling***

Die Qualitätsanforderungen sind in der Tabelle 0.1 aufgeführt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Systemqualität | SEHR GUT | GUT | NORMAL | Nicht relevant  **Tabelle 0.1** |
| Funktionalität |  |  |  |  |
| Angemessenheit | X |  |  |  |
| Genauigkeit | X |  |  |  |
| Interoperabilität |  | X |  |  |
| Sicherheit | X |  |  |  |
| Konformität | X |  |  |  |
| Zuverlässigkeit |  |  |  |  |
| Reife | X |  |  |  |
| Fehlertoleranz | X |  |  |  |
| Wiederherstellbarkeit |  | X |  |  |
| Konformität |  | X |  |  |
| Benutzbarkeit |  |  |  |  |
| Verständlichkeit | X |  |  |  |
| Erlernbarkeit | X |  |  |  |
| Bedienbarkeit | X |  |  |  |
| Attraktivität | X |  |  |  |
| Konformität | X |  |  |  |
| Effizienz |  |  |  |  |
| Zeitverhalten | X |  |  |  |
| Verbrauchsverhalten | X |  |  |  |
| Konformität | X |  |  |  |
| Wartbarkeit |  |  |  |  |
| Analysierbarkeit |  | X |  |  |
| Änderbarkeit |  | X |  |  |
| Stabilität |  | X |  |  |
| Testbarkeit |  | X |  |  |
| Konformität |  | X |  |  |
| Portabilität |  |  |  |  |
| Anpassbarkeit |  |  |  | X |
| Installierbarkeit |  |  |  | X |
| Koexistenz |  |  |  | X |
| Austauschbarkeit |  |  |  | X |
| Konformität |  |  |  | X |

**Tabelle 0.2**

## Abnahmekriterien

*Abnahmekriterien legen fest, wie Anforderungen bei der Abnahme auf ihre Realisierung überprüft werden können.*

*Bsp.:*

***Anforderung : /F12/*** *Wenn ein Kunde oder eine Firma sich von einer bereits gebuchten Veranstaltung später als X Wochen vor der Veranstaltung abmeldet, dann muss das System Stornogebühren in Höhe der Veranstaltungsgebühr berechnen oder nach einem Ersatzteilnehmer fragen.*

***Abnahmekriterium***

***Ausgangssituation:*** *Ein Kunde, eine Firma, ein Seminar und eine Veranstaltung sind angelegt. Der Kunde und die Firma haben die Veranstaltung gebucht.*

***Ereignis:*** *Der Kunde und die Firma stornieren die Veranstaltung X Wochen vor der Veranstaltungsbeginn (wobei festgelegt sein muss, ab wann Stornogebühren in Höhe der Veranstaltungsgebühr anfallen, X muss in diesen Zeitraum fallen).*

***Erwartetes Ergebnis:*** *Das System fragt, ob der Kunde und die Firma einen Ersatzteilnehmer stellen (Name usw. wird angefordert). Wenn nein, dann wird mitgeteilt, dass der volle Veranstaltungsbetrag fällig wird. Die Stornierung wird bestätigt*

***Spezifikationen EasyGamBling***

# Qualitätskonzept

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Autor | Quelle | Status | Datum | Kommentar |
| 0.1 | M. Mustermann | Geschäftsführer  Teachware | In Bearbeitung | 07.09.2014 | Verfeinerung des Lastenheftes **VO.1** |

Basierend auf dem Qualitätsstandard der DIN ISO 9126 hat die zu entwickelnde Software nachfolgenden Punkten insoweit zu entsprechen, wie es unter jedem Punkt genauer beschrieben ist.

## Tauglichkeit (Capability)

*Die Tauglichkeit des zu entwickelnden Systems muss den Anwendern und Nutzern des Systems die Funktionalitäten entsprechend ihrer Rollen gewährleisten.*

## Benutzerfreundlichkeit(Usability)

*Die Anwendersicht muss leicht verständlich und einfach zu navigieren sein. Durch einheitliche Benennung der Menüpunkte muss der Anwender / Nutzer schnell zu den gewünschten Funktionalitäten gelangen und zwischen diesen navigieren können. Etwaige Fehlermeldungen sollten für jeden klar verständlich formuliert sein und Optionen zur Wiederholung bzw. Abbruch der Aktion ermöglichen.*

## Leistung (Performance)

*Die im Pflichtenheft festgelegten Leistungen sollten erreicht werden und durch geeignete Tests belegt werden.*

## Korrektheit (Correctness)

*Die Software sollte entsprechend der festgelegten Funktionen die erwarteten Ergebnisse ausführen. Z.B.: Löschen von Daten, sollen redundanzfrei gelöscht werden. Bei unterschiedlichen Rechten bedingt durch Rollen soll nur auf entsprechende Funktionalitäten zugegriffen werden können.*

## Verlässlichkeit/Ausfallsicherheit (Reliability)

*Die erwartete Leistung gemäß Pflichtenheft muss mindestens erreicht werden.*

## Wartbarkeit (Maintainability)

*Die Software sollte im Hinblick auf etwaige Änderungen von ihrem strukturellen Aufbau zukünftige Änderungen / Erweiterungen einfach anpassbar sein.*

## Portierbarkeit (Portability)

*Software muss entsprechend der Anforderungen auf verschiedene Systeme (Hardware, Software Organisatorische) portierbar sein und durch geeignete Auswahl der Programmiersprachen erstellt werden.*

# Testkonzept

## Komponententests

*Auch Modultest, Klassentest oder Unit Tests genannte Möglichkeiten, sind die zuerst geplanten Tests der erstellten SW. Durch diese Tests werden die einzelnen Komponenten voneinander isoliert betrachtet und Wechselwirkungen (cross relations) mit anderen Komponenten werden erstmal ausgeschaltet. Hierdurch wird sichergestellt, dass auftretende Fehler in der getesteten Komponente auftreten und nicht von außen beeinflusst werden.*

*Ziel des Komponententests ist zu prüfen, ob der vorliegende Software-Baustein die in der Spezifikation niedergelegten funktionalen und nicht-funktionalen Anforderungen erfüllt.*

* *Unter funktionalen Anforderungen versteht man dabei diejenigen Anforderungen, die das Ein- bzw. Ausgabeverhalten bzgl. der Korrektheit der gelieferten Ergebnisse spezifizieren.*
* *Nicht-funktionale Anforderungen dagegen umfassen z.B. Performanz Aspekte oder den Speicherverbrauch während der Laufzeit.*

*Im Komponententest identifizierte funktionale Defekte sind z.B. Berechnungsfehler durch fehlerhafte Algorithmen. Ein nicht-funktionaler Defekt wäre z.B. eine zu lange Ausführungszeit der Komponente aufgrund eines für den Anwendungsfall falsch eingesetzten Algorithmus.*

*Beim Komponententest können prinzipiell zwei Strategien verfolgt werden: Black-Box Tests und White-Box Tests.*

* *Beim Black-Box Test werden sämtliche Testfälle nur anhand der Spezifikation des Software-Bausteins gebildet, d.h. der Tester hat keinerlei Einsicht in die interne Realisierung der zu testenden Komponente. Die Kommunikation mit der Komponente während des Tests erfolgt ausschließlich über die extern sichtbaren Schnittstellen.*
* *Beim White-Box Test werden die Testfälle anhand des vorliegenden Programmcodes gebildet, d.h. der Tester hat Zugang zu und ein entsprechendes Verständnis von den internen Details der zu testenden Komponente. Der White-Box Test ermöglicht insbesondere auch den Einsatz von Debuggern, um interne Zustände zu beobachten und gezielt Veränderungen während Testlaufs vornehmen zu können.*

*In beiden Fällen müssen passende Testtreiber geschrieben werden, die das Testobjekt mit den Testfällen aufrufen und die Reaktionen des Testobjekts registrieren.*

## Passwort

* *Funktioniert die Passwortabfrage mit gewähltem / vergebenen Passwort*
* *Wird die entsprechende Nutzeransicht gemäß Rolle angezeigt / kann auf andere Funktionalitäten zugegriffen werden*
* *Etc.*

## Daten Einlesen

* *Bei Auswahl von Datensätzen werden die richtigen Datensätze angezeigt*
* *Werden alle ausgewählten Daten angezeigt*
* *Bei Weiterleitung an andere Schnittstellen, werden die Daten vollständig und korrekt übergeben*
* *Etc.*

## Grafiken einbinden

* *Werden vorhandene Grafiken sauber (farbecht) angezeigt, in der richtigen Größe und der richtigen Stelle.*
* *Bei Veränderung der Auflösung passen sich die Grafiken richtig an oder verschieben sie sich*
* *Etc.*

## Integrationstests

*Voraussetzung für die Überprüfung der Beziehungen zwischen den Komponenten ist die Überprüfung aller einzelnen Komponenten für sich durch einen vorangegangenen Komponententest. Mit Hilfe des im Anschluss an die Integration durchzuführenden Integrationstests wird somit das fehlerfreie Zusammenspiel der beteiligten Systemkomponenten überprüft. Dabei stehen vor allem das Testen der realisierten Schnittstellen und das Zusammenwirken der Komponenten im Vordergrund der Betrachtung.   
In welcher Weise sind die verschiedenen Komponenten bzw. Module in das gesamte System integriert.   
Im Kontext von Integrationstests bei Komponenten werden i. d. R. folgende Tests durchgeführt:*

* *Usability-Tests (Test der Nutzbarkeit)*
* *Funktionstest (Test der Methoden) und*
* *Scenario-Tests (Tests von Business-Szenarien bzw. Use-Cases).*

*Bei den Usability-Tests werden in erster Linie die Verlinkung, das Design und die Akzeptanz aus Nutzersicht getestet. Bei den Funktions- und Szenario-Tests werden auf der Basis von Business- bzw. Use-Cases unterschiedliche Anfangs- und Endbedingungen variiert und im Rahmen von Test-Cases getestet.   
  
Die Integrationstests sollten nach Möglichkeit in der Betriebsumgebung stattfinden.*

## Systemtests

*Mit dem erfolgreich abgeschlossenen Integrationstest liegt das komplett zusammengebaute System vor, d. h. es sind alle Systemkomponenten im Gesamtsystem integriert und einem Integrationstest unterzogen worden. Im folgenden Systemtest wird das System als Ganzes betrachtet. Der Systemtest ist der abschließende Test der Entwickler und Qualitätssicherer, welcher idealerweise in einer Testumgebung, die der späteren Einsatzumgebung möglichst nahe kommt, durchgeführt wird.   
Unterscheidet sich die Entwicklungsumgebung von der Testumgebung, dann muss das System vor Beginn des Systemtests auf die Testumgebung portiert werden. Dies sollte idealerweise automatisiert erfolgen, um bei möglichen Fehlern die Portierung effizient wiederholen zu können.   
  
Innerhalb des Systemtests, bei dem das System als Ganzes in seiner Laufzeitumgebung getestet wird, wird die Erfüllung sowohl von funktionalen wie auch nichtfunktionalen Anforderungen geprüft. Die Erfüllung der funktionalen Anforderungen, wie z. B. Sicherheit und Interoperabilität, werden i. d. R. durch Durchführung folgender Teiltests geprüft:*

* *Funktionstest,*
* *Sicherheitstest,*
* *Interoperabilitätstest.*

*Mit Hilfe des Funktionstests wird überprüft, ob alle geforderten und definierten Funktionen des Systems vorhanden sind und auch wie geplant realisiert wurden. Hierbei werden aus dem Pflichtenheft die Testsequenzen übernommen und mit Hilfe von funktionalen Testverfahren systematisch hergeleitet.   
  
Neben der Erfüllung der geforderten funktionalen Eigenschaften muss das System auch nichtfunktionale Anforderungen, wie z. B. Effizienz, Benutzbarkeit und Zuverlässigkeit einhalten.   
Dies kann u. a. durch folgende Teiltests überprüft werden:*

* *Leistungstest,*
* *Performanz Test (performance test)*
* *Test des Speicherverbrauchs und der Auslastung des Prozessors*
* *Recovery Testing*
* *Benutzbarkeitstest (Usability-Test).*

*Unter dem Begriff* ***Leistungstest*** *werden i. d. R. der Massentest, der* ***Zeittest****, der* ***Lasttest*** *und der* ***Stresstest*** *subsummiert. Allgemein wird mit dem Leistungstest das in der Produktdefinition festgelegte Leistungsverhalten des Systems überprüft.   
  
Bei der Einführung des Systems werden vor allem Funktions- und Leistungstest, hier insbesondere der Lasttest, benötigt, um die angestrebte Leistungsfähigkeit des Systems überprüfen zu können. In diesem Kontext ist es besonders wichtig, dass die Art und der Umfang dieser Tests rechtzeitig geplant werden. Das beinhaltet, dass aus den Anforderungen und Leistungsbeschreibungen die Testfälle und die benötigten Testdaten abgeleitet werden. Der erfolgreiche Systemtest ist Voraussetzung für den abschließenden Abnahmetest durch den Kunden bzw. Anwender.*

## Abnahmetest

*Der Abnahmetest ist das zentrale Element der Abnahme. Der Abnahmetest dient zum Nachweis, dass die zwischen Auftraggeber und Auftragnehmer vereinbarten Leistungen erbracht wurden.*

*Der Abnahmetest besteht aus einer Reihe von Einzeltests verschiedener Art, d.h. Teile des Abnahmetests beinhalten z.B.:*

* *Test auf Benutzerakzeptanz*
* *Test auf vertragliche Akzeptanz*
* *Feldtest*
* *Zielrechnertest*
* *Lasttest, Volumentest*
* *Robustheitstest*
* *Dokumentationstest*
* *Oberflächentest*

*Um auftretende Mängel klar einordnen zu können, muss eine Fehlerklassifikation vereinbart und vertraglich fixiert werden. Dabei ist auch festzuhalten, wann der Auftraggeber die Abnahme verweigern kann. Eine mögliche Fehlerklassifikation könnte folgendermaßen aussehen:*

* ***abnahmehinderliche Fehler****: Wesentliche Teile, die für den Betrieb und die Nutzung des Systems erforderlich sind, können nicht verwendet werden. Die Abnahme kann beim Auftreten eines solchen Fehlers verweigert werden.*
* ***schwere Fehler****: Wesentliche Teile des Systems, die nicht zwingend zum Betrieb, aber zur Nutzung des Systems notwendig sind, sind nicht nutzbar. Ein Betrieb des Systems ist eingeschränkt möglich. Es dürfen maximal 3 Fehler dieser Art auftreten. Andernfalls kann die Abnahme verweigert werden.*
* ***mittlere Fehler****: Teile des Systems, die unmittelbar weder für den Betrieb noch für die Nutzung des Systems notwendig sind, können nicht verwendet werden. Ein Betrieb und die Nutzung des Systems sind mit geringen Einschränkungen möglich. Es dürfen maximal 10 Fehler dieser Art auftreten. Andernfalls kann die Abnahme verweigert werden.*
* ***leichte Fehler****: Es sind Mängel in der Darstellung, im Layout oder in der Dokumentation enthalten. Es dürfen nicht mehr als 30 Fehler dieser Art auftreten. Andernfalls kann die Abnahme verweigert werden.*

# Anhänge

## USE CASE Allgemein grob

## USE CASE System Umwelt

## USE CASE Kundenbereich Darstellung

## USE CASE Seminarbereich

## ER Modell einfach

## ER Modell Ausführlich

## Klassendiagramm Easy GamBling

## Zustandsdiagramm Veranstaltungsbereich

## Sequenzdiagramm erstellen TN Liste

## Sequenzdiagramm Alternative Kundenbuchung

# Glossar

|  |  |
| --- | --- |
| **Begriff** | **Erklärung** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Abkürzungsverzeichnis

|  |  |
| --- | --- |
| **Abkürzung** | Erläuterung |
| **/LF10/** | Lastenheft Funktion 10 – man benutzt meist zehnerschritte damit es einfacher ist eventuell zu einer Funktion hinzukommende Unterpunkte einfacher zu ergänzen, z.B.: LF11, LF12 – diese Wären dann sinngemäß und funktional der Übergeordnete Funktionsanforderung LF10 zugeordnet |
| **/LK10/** | Kontextanforderung 10 des Lastenheftes |
| **/LR10/** | Lastenheft Rahmenbedingung |
| **/LV10/** | Lastenheft Vision |
| **/LZ10/** | Lastenheft Ziel |
| **SemOrg** | Seminar Organisation – Name der zu erstellenden Anwendung |
| **TBD** | To be determined – Wird noch festgelegt |
| **VO.1** | Version 0.1 |

**Bei der Erstellung des Pflichtenheftes wird auf die Anforderungen des Lastenheftes zurückgegriffen und im Pflichtenheft folgendermaßen kenntlich gemacht:**

|  |  |
| --- | --- |
| Lastenheft | Pflichtenheft |
| /LF10/ | /F10/ (LF30) |