

# Adaptive Online-Lernhilfe

Seminar Technologiegestütztes Lernen (TGL)

Seminararbeit

von

Jeremy Schmidt, Dane Leube

An der Fakultät für Informatik

Lehrstuhl der Systeme für Informationsverwaltung

Betreuer: Dr. Simone Braun  
Tatjana Streit  
Christian Krakowski

31.07.2015

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	CAS Campus . . . . .	1
1.2	Motivation . . . . .	2
1.3	Weiteres Vorgehen . . . . .	2
<b>2</b>	<b>Stand der Forschung</b>	<b>3</b>
2.1	Lernen . . . . .	3
2.1.1	Problembasiertes Lernen . . . . .	3
2.1.2	Differenzielles Lernen . . . . .	3
2.2	Game-Based Learning . . . . .	4
<b>3</b>	<b>Grundlagen</b>	<b>5</b>
3.1	Online-Lernhilfe . . . . .	5
3.1.1	Begriffsdefinition . . . . .	5
3.1.2	Verwendung in der Arbeit . . . . .	5
3.2	Adaption . . . . .	6
3.2.1	Begriffserläuterung . . . . .	6
3.3	Adaption an den Nutzer . . . . .	6
3.4	Gamification . . . . .	6
3.4.1	Definition . . . . .	6
3.4.2	Verwendung in unserer Arbeit . . . . .	6
3.5	Recommendersysteme . . . . .	7
3.5.1	Begriffserläuterung . . . . .	7
3.5.2	Verwendung in unserer Arbeit . . . . .	7
3.6	Personas . . . . .	7
3.6.1	Begriffserläuterung . . . . .	7
3.6.2	Verwendung in unserer Arbeit . . . . .	7
<b>4</b>	<b>Nutzer</b>	<b>8</b>
4.1	Nutzerrollen . . . . .	8
4.1.1	Professoren . . . . .	8
4.1.2	Studenten . . . . .	9
4.1.3	Studierendensekretariat . . . . .	9
4.1.4	Studiengangsbearbeiter . . . . .	9
4.1.5	Admins . . . . .	9

4.2	Nutzer Klassifizierung . . . . .	10
4.2.1	Powernutzer . . . . .	10
4.2.2	Gelegenheitsnutzer . . . . .	10
4.2.3	Spezielle Nutzer . . . . .	11
<b>5</b>	<b>Lösungsansätze</b>	<b>12</b>
5.1	Bewertungskriterien . . . . .	12
5.1.1	Begründung der Kriterien . . . . .	13
5.1.2	Wirtschaftlich . . . . .	13
5.1.2.1	Anfangskosten . . . . .	13
5.1.2.2	Wartungskosten . . . . .	13
5.1.2.3	Nutzen . . . . .	13
5.1.3	Technisch . . . . .	13
5.1.3.1	Wartbarkeit . . . . .	13
5.1.3.2	Datenschutz . . . . .	14
5.1.3.3	Usability . . . . .	14
5.2	Klassische Hilfe . . . . .	15
5.2.1	Erläuterung . . . . .	15
5.3	Lernspiel-App . . . . .	16
5.3.1	App . . . . .	16
5.3.2	Lernspiel . . . . .	16
5.3.3	Lösungsansatz . . . . .	16
5.4	Expertensystem . . . . .	17
<b>6</b>	<b>Empfehlung</b>	<b>19</b>
6.1	Bewertung der Lösungsansätze anhand der aufgestellten Nutzerklassen . . .	19
6.2	Bewertung der Lösungsansätze anhand der aufgestellten Kriterien . . . . .	19
6.3	Konkrete Empfehlung . . . . .	19
<b>7</b>	<b>Zusammenfassung</b>	<b>20</b>
<b>8</b>	<b>Ausblick</b>	<b>21</b>
	<b>Literaturverzeichnis</b>	<b>23</b>
	Literatur . . . . .	23



# 1. Einleitung

”Ein Mensch lernt sein Leben lang.”

Dieser weise Spruch ist allgemein gültig. Jeder Mensch, sei es im Beruf oder im privaten Umfeld muss neue Inhalte erlernen. Wie schnell er dies tut, das hängt von seinem individuellem Lerntempo ab. Damit er eine Lernhilfe erhält und neue Inhalte interessant aufbereitet, gibt es computerbasierte Hilfesysteme. Diese Hilfesysteme sind dafür da, um neue Inhalte schnell zu lernen. Wie ein solches Hilfesystem aussehen kann wird in dieser Arbeit vorgestellt. Die neuen Lerninhalte beziehen sich auf eine spezielle Software, welche bei der Ausschreibung des Seminarthemas vorgeschrieben wurde. Diese Software soll im Folgenden kurz erläutert werden.

## 1.1 CAS Campus

Die Zielsoftware, worauf unsere spätere Empfehlung basiert, ist die Software CAS Campus. Diese Software ist von der CAS Software AG aus Karlsruhe. Das Ziel der Software ist es einen kompletten Student-Lifecycle (siehe Abbildung 1.1) nachzubilden. Der Lifecycle beginnt mit der Bewerbung des Studenten an einer Hochschule und endet mit dem Alumni Zustand<sup>1</sup>. Bevor ein Student zu dem Alumni-Zustand kommt, muss er zuvor einige Stationen begleiten. Die Software CAS Campus soll den Studenten genau auf diesem Weg begleiten, indem alle Prozesse abgebildet werden. Die Prozesse sind hierbei sehr vielschichtig. Sie reichen vom einfachen Eintragen der Noten bis hin zu einer komplexen Modellierung von ganzen Studiengängen. Weitere Prozesse sind beispielsweise Terminmanagement. Hier müssen Vorlesungen zu Räumen zugeordnet werden und auf Konflikte bei der Belegung geprüft werden. Im Mittelpunkt dieser Prozesse steht immer der Student. Er ist sozusagen der Endkunde der Anwendung und steht im Zentrum eines modernen Campus-Management-Systems.

---

<sup>1</sup>Quelle: <http://cas-education.de> entnommen am 27.07.15



Abbildung 1.1: Student-Lifecycle

## 1.2 Motivation

Das Ziel dieser Arbeit ist es, eine mögliche Lösung zu finden, wie eine adaptive Online-Lernhilfe aussehen kann. Hierzu müssen passende Lösungsansätze gefunden werden. Die Lösungsansätze müssen auf die Zielsoftware CAS Campus angewandt werden können, so dass eine spätere Implementierung der Lösung erfolgen kann. Wichtig hierbei ist die Zentrierung auf den Nutzer. Diese Zentrierung wurde als Vorgabe zur Adaptivität der Lösung vorgegeben und soll sich in allen Lösungsansätzen widerspiegeln.

## 1.3 Weiteres Vorgehen

Im folgenden Text wird zuerst der aktuelle Status-Quo in Sachen Lernen und Online-Hilfe erläutert. Darauf aufbauend wird erklärt, was genau Adaptiv für unsere Arbeit bedeutet und worauf wir unseren Schwerpunkt beziehen. Zur besseren Einführung in die Thematik werden die Nutzer und deren Probleme in Form von Personas vorgestellt. Diese Nutzer werden anschließend klassifiziert, damit bessere Lösungsansätze gefunden werden können. Anhand der Klassifizierung werden wir unsere Lösungsansätze vorstellen, diese kritisch bewerten und eine anschließende Empfehlung daraus generieren. Zum Abschluss der Arbeit möchten wir ein kleines Fazit mit Ausblick auf ein weiteres Vorgehen bei der Arbeit geben.

## 2. Stand der Forschung

In diesem Kapitel wollen wir den aktuellen Stand der Wissenschaft vorstellen. Hierbei beziehen wir uns auf das Thema Lernen. Welches Wissen gibt es hierbei bereits und auf welchem Stand sind diese? Das wollen wir in diesem Kapitel klären, bevor die Grundlagen der Arbeit erläutert werden.

### 2.1 Lernen

Das Thema Lernen ist ein wichtiger Bestandteil dieser Arbeit. Aus diesem Grund wollen wir uns mit dem aktuellen Stand des Lernens befassen, so dass bei unseren Lösungsansätzen das Lernen besser begründet werden kann.

#### 2.1.1 Problembasiertes Lernen

”Problem based Lerning (PBL) is a learner-centred approach for designing learning environments in authentic, ill-structured problems” entnommen aus (Zumbach, o. J.).

Das Problembasierte Lernen, wie im Zitat deutlich wird, bezieht sich auf das Erarbeiten von selbstständigen Lösungsansätzen zum Lösen eines Problems. Der Lernende soll ein Thema selbstständig erarbeiten, ohne das jemand anderes einen Lösungsweg vorzeichnet. Auf dem Weg zu der Lösung eines Problems muss er evtl. mehrere Ansätze probieren um sein Ziel zu erreichen. Wie im Zitat bereits erwähnt steht der Lernende im Mittelpunkt. Der Lernende muss hierbei selbst die Initiative ergreifen. Ein Lehrer steht beim Lernprozess nur als Helfende Hand zur Seite. Er dient dabei lediglich als Coach, der evtl. aufkommende Fragen des Lernenden beantwortet und ihn in eine spezielle Richtung leitet, falls der Lernende nicht mehr weiterkommt. Beim Thema Software haben wir eine ähnliche Problemstellung. Ein möglicher Anwender kommt an einem Punkt in der Software nicht mehr weiter und muss selbstständig eine Lösung für das konkrete Problem finden.

#### 2.1.2 Differenzielles Lernen

Ein weitere Ansatz zum Thema Lernen gibt es in der Sporttheorie. Die Idee des differenziellen Lernens bezieht sich auf das Erlernen von Bewegungen im Sport. Beim differenziellen

Lernen werden Bewegungen variiert. Beim Tischtennis beispielsweise wird ein Schlag auf verschiedene Wege durchgeführt. Hierbei werden vor allem Körperwinkel variiert.<sup>1</sup> Es werden drei Bälle vom Lernenden gespielt und jede Bewegung sollte hierbei auf einem anderen Weg durchgeführt werden. Das Ziel dieser Form von Lernen ist es kein ideales Technikleitbild für eine Bewegung vorzugeben, also keine Lösung sondern die Bewegung so vielfältig wie möglich zu streuen. Wenn er dieses Training durchführt, dann kann ein Lernender in jeder Situation die korrekte Bewegung anwenden. Diese Form des Bewegungslernens ist sehr stark verwandt mit dem problemorientierten Ansatz. Das Differenzielle Lernen gibt ebenfalls ein Problemstellung vor, die mit verschiedenen Lösungswegen gelöst werden soll. Der Unterschied hierbei besteht in der Vorgabe von Lösungsansätzen im Vergleich zum komplett selbständigen problematisierten Lernen. Von einem Lehrer wird ein konkretes Problem vorgegeben. Dieses Problem soll im Anschluss auf verschiedene Wege gelöst werden. Durch die Differenzierung entsteht ein grobes Bild im Kopf, wie ein Lösungsansatz aussehen soll. Im Techniktraining im Sport zum Beispiel wie eine konkrete Technik im Sport aussehen soll. Wenn es in Zukunft zu einem solchen Problem kommt, so kann der Lernende aus verschiedenen Lösungsmöglichkeiten die passende auswählen. Diese Möglichkeit gibt es bei der Bedienung von Software auch. Hier ist es ebenfalls möglich eine konkrete Problemstellung, wie beispielsweise das Anlegen eines Adressdatensatzes auf verschiedene Wege zu lösen. Aus diesem Grund ist der differenzielle Lernansatz in unserer Arbeit interessant.

## 2.2 Game-Based Learning

Ein aktueller Ansatz um schnell neue Inhalte zu Lernen ist das sogenannte "Game-Based Learning". Dieser Ansatz geht von einem spielerischen Lernen aus. Neue Lerninhalte werden in einem digitalen Spiel aufbereitet. Digitale Spiele sind nach Wagner /cite ein regelbasiertes, interaktives Medium, das Spielende „emotional bindet und innerhalb eines von der objektiven Realität abgegrenzten Raums stattfindet“ und dessen „zugrunde liegende Interaktionstechnologie rein digitaler Natur ist“. Dieser Ansatz verspricht Spaß beim Lernen von neuen Inhalten. Dieser Spaß fördert das kreative Denken und hilft neue Lerninhalte einfacher zu verstehen und vor allem besser behalten und verinnerlichen zu können.

---

<sup>1</sup>Quelle: (Wolfgang I. Schöllhorn, 2009)



## 3. Grundlagen

Um ein besseres Verständnis dieser Arbeit zu gewährleisten, sind gewisse Grundlagen zu dem Thema erforderlich. Diese Begriffe werden nachfolgend noch einmal aufgeführt, definiert und in den Zusammenhang mit dieser Arbeit gesetzt. Beginnen möchten wir dabei mit dem Ersten der vier Begriffe, nämlich der Online-Lernhilfe.

### 3.1 Online-Lernhilfe

Wir werden eine kurze Begriffsdefinition geben und anschließend die Verwendung in unserer Arbeit erläutern. Dies hilft beim Verständnis des Problems.

#### 3.1.1 Begriffsdefinition

Die Online-Lernhilfe setzt sich aus zwei Begriffen zusammen, der Online-Hilfe und der Lernhilfe. Die Online-Hilfe bezeichnet eine „*Hilfefunktion, die nur über eine direkte Verbindung mit dem Internet genutzt werden kann*“ (Duden, Zugriff am 20.07.2015c). Die Lernhilfe beschreibt eine „*Hilfe, Mittel oder Anhaltspunkt beim Lernen von etwas*“ (Duden, Zugriff am 20.07.2015b). Somit ist die Online-Lernhilfe ein Hilfe-, Mittel- oder Anhaltspunkt, der eine Verbindung mit dem Internet voraussetzt.

#### 3.1.2 Verwendung in der Arbeit

Unter Online-Hilfen verstehen wir die klassischen Hilfsmodule von Software. Diese bietet dem Nutzer Hilfe zu einem bestimmten Themengebiet in der Software. Die klassische Form ist ein Handbuch das zu der Software beigelegt wird. Dieses Handbuch erklärt die Software mit ihren Funktionen. Früher waren die Handbücher sehr umfangreich, so dass man zuerst mehrere Seiten dieses Handbuch lesen musste, um die Software überhaupt erst verstehen zu können. Heutzutage sind die Benutzeroberflächen vereinfacht, so dass die Software im Idealfall selbsterklärend ist und keine weitere Hilfestellung notwendig. Das Software immer mehr Prozesse abbilden kann, wird die Komplexität einer Software jedoch immer zunehmen. Damit die Komplexität weiterhin von einem User bedient werden kann, muss

er zum einen neue Bedienelemente einer Software lernen und zum Anderen eine schnelle Hilfe erhalten, wenn er an einem bestimmten Punkt in der Software nicht weiterkommt.

## 3.2 Adaption

Wie aus dem Titel unserer Arbeit ersichtlich, benötigen wir eine Definition des Adaptions-Begriffs, um unsere Arbeitsgebiete klar abzugrenzen.

### 3.2.1 Begriffserläuterung

Unter dem Begriff Adaption, welcher von dem lateinischen Wort „*adaptare*“ stammt und „*anpassen*“ bzw. „*passend*“ herrichten (Duden, Zugriff am 20.07.2015a) bedeutet, versteht man die Anpassung eines Objekts an einen Zweck.

## 3.3 Adaption an den Nutzer

Der Begriff der Adaption ist sehr breit auf den Bereich Online-Lernhilfe anwendbar. Eine Adaption ist deshalb an sehr vielen Stellen möglich. Aus diesem Grund wurde in Absprache mit den Betreuern festgestellt, dass die Lösungsvorschläge nur eine Adaption an den Nutzer beinhalten sollen. Eine Adaption an verschiedene Endgeräte, wie Tablets, Smartphones oder Computer sollen deshalb aus der Betrachtung und Bewertung ausgeschlossen werden. Lösungsvorschläge, welche im Verlauf der Arbeit vorgestellt werden und adaptiv sind, stellen folglich ausschließlich eine Anpassung an den Nutzer dar. Darunter kann man sich zum Beispiel einen grafisch aufgearbeiteten Hilfetext für den Benutzer der Software vorstellen, der sich dynamisch an den derzeit verwendeten Teil der Software anpasst.

## 3.4 Gamification

Als zweite Grundlage haben wir den Ansatz Gamification. Dieser ist für unsere Erläuterung sehr wichtig und wird deshalb im Folgenden aufgeführt.

### 3.4.1 Definition

*„Gamification ist die Übertragung von spieltypischen Elementen und Vorgängen in spiel-fremde Zusammenhänge mit dem Ziel der Verhaltensänderung und Motivationssteigerung bei Anwenderinnen und Anwendern. Zu den spieltypischen Elementen gehören Beschreibungen (Ziele, Beteiligte, Regeln, Möglichkeiten), Punkte, Preise und Vergleiche. Zu den spieltypischen Vorgängen zählt die Bewältigung von Aufgaben durch individuelle oder kol-laborative Leistungen.“* (Bendel, 2015)

### 3.4.2 Verwendung in unserer Arbeit

Da es sich bei unseren Lösungsvorschlägen hauptsächlich um verschiedene Arten von Hilfen für ein konkretes Problem handelt, verwenden wir nur einen Teil aus dem Bereich des Gamification in unseren Lösungsvorschlägen. Es handelt sich bei dem verwendeten Teil um die spieltypischen Elemente, wie Punkte, Preise und Vergleiche und die spieltypischen Vorgänge. Wie diese dann konkret in unseren Lösungsvorschlägen verwendet werden, lässt sich im Kapitel der Lern-App nachlesen.

## 3.5 Recommendersysteme

Ein Teil unseres Lösungsansatzes verwendet ein Recommendersystem.

### 3.5.1 Begriffserläuterung

*„Ein Recommendersystem sammelt Empfehlungen von Benutzern, aggregiert diese Empfehlungen und leitet sie an geeignete Adressaten weiter.“*(aus (recommender, o. J.))

### 3.5.2 Verwendung in unserer Arbeit

In unserer Arbeit wollen wir die Recommendersysteme in Form eines Expertensystems verwenden. Der Experte kennt ein bestimmtes Gebiet der Software und generiert später die Empfehlung. Diese kann im Kontext des Lernens entweder ein Tipp zur Lösungsfindung sein oder aber auch die direkte Lösung zu einem Problem. Im Softwareumfeld kann dies besser differenziert werden. Entweder der Experte zeigt einen möglichen Weg auf um das Problem zu beseitigen oder aber er erarbeitet mit dem Lernenden zusammen als Coach die Lösung des Problems. Diese beiden Möglichkeiten können mithilfe eines Recommendersystems begangen werden und werden in unserem dritten Lösungsansatz weiter vertieft.

## 3.6 Personas

Um eine Klassifizierung der Nutzer vornehmen zu können, müssen zuerst sogenannte Personas erstellt werden.

### 3.6.1 Begriffserläuterung

„Personas (lat. Maske) sind Nutzermodelle, die Personen einer Zielgruppe in ihren Merkmalen charakterisieren. Sie können z. B. einem Entwicklerteam aufgrund ihrer umfangreichen Beschreibung helfen, sich in die Lage der potenziellen Nutzer zu versetzen und diese Perspektive während des gesamten Designprozesses leicht zu vertreten.“ (onlinemarketing praxis.de, 2015)

### 3.6.2 Verwendung in unserer Arbeit

Wir haben Personas in Absprache mit unseren Betreuern erstellt, um mögliche Rollen bei der Nutzung von CAS Campus herauszufiltern. Das primäre Ziel bei der Verwendung von Personas ist es, die Adaptivität auf den Nutzer klar herauszustellen.

## 4. Nutzer

Eine klare Vorgabe in unserer Arbeit war die Adaptivität auf Nutzer. Damit die Zielgruppe der Software klar wird, werden im Folgenden die Personas erklärt, die für unsere Zielgruppe der Software bildet. Aufgrund dieser Personas werden wir eine Klassifizierung des Nutzers durchführen. Diese Klassifizierung soll uns dabei helfen die Lösungsansätze passend auszuwählen und später zu bewerten.

### 4.1 Nutzerrollen

Das Anwendungsfeld der Software ist wie in Einleitung beschrieben eine Hochschule. Im Mittelpunkt der Anwendung steht der Student, der sozusagen der Endkunde ist. Alle Dienstleistungen, die von der Software übernommen werden sind für den Studenten. Hierbei werden verschiedene Prozesse durchlaufen, die von verschiedenen Personen während des sogenannten Studen-Lifecycles gesteuert werden. Im Folgenden werden wir erklären, wie die gegebene Nutzerrolle die Zielsoftware verwendet und welche Ziele bei der Verwendung verfolgt werden. Die einzelnen Personas sind in der Diskussion mit unseren Betreuern und auf von der CAS bereitgestellten Materialien entstanden.

#### 4.1.1 Professoren

Beginnen wollen wir mit der Persona Professor. Er ist für die Lerninhaltsaufbereitung des Studenten zuständig. Ein Professor hat eher weniger mit der Software zu tun, da die meiste Arbeit die Sekretärin abnimmt. Die Verwendung beschränkt sich auf das Lesen eines Termines oder das Eintragen einer Lehrveranstaltung. Weiterhin auf ein eventuelles Einladungsmanagement. Ein Professor ist somit ein Nutzer, der sehr wenig Zeit mit der Verwendung der Software aufbringt. Somit muss er, wenn er in einem Problem bei der Verwendung der Software steckt schnell eine Lösung haben. Weiterhin ist die Zeit des Professors aufgrund seiner Aufgabengebiete begrenzt, womit eine lange Einarbeitungszeit ebenfalls vermieden werden sollte.

### 4.1.2 Studenten

Die Studenten sind wie Eingangs erwähnt die Profiteure der Software. Dadurch, dass alle Prozesse, die von der Software abgebildet werden nur für den Studenten da sind, steht er im Fokus. Die Verwendung der Software hält sich allerdings vom zeitlichen Aufwand her im Rahmen. Er verwendet zwar viele Bereiche der Software allerdings nur sehr kurz und schnell. Sein primäres Ziel bei der Verwendung ist Ähnlich dem Professor. Er möchte schnell an Informationen kommen und sich nicht lange in eine Software einarbeiten, da er andere Inhalte für sein Studium lernen muss. Somit sollte er komplizierte Lerninhalte einfach aufbereitet bekommen, so dass er schnell ein gutes Verständnis erlangen kann.

### 4.1.3 Studierendensekretariat

Das Studierendensekretariat oder auch Help-Desk genannt ist für die komplette Verwaltungsarbeit eines Studenten zuständig. Die Nutzer dieser Rolle verwenden die Software im täglichen Gebrauch und dies für mehrere Stunden am Tag. Aus diesem Grund müssen sie die Software möglichst effektiv und schnell beherrschen. Komplexe Prozesse bei der Bedienung der Software muss den Arbeitern geläufig sein, um effizient arbeiten zu können. Weiterhin muss bei einem konkreten Problem beim Verwenden der Software schnell Abhilfe geschaffen werden.

### 4.1.4 Studiengangsbearbeiter

Eine besondere Rolle im Lebenszyklus der Software nimmt die Rolle des Studiengangsbearbeiters ein. Diese ist für die Modellierung von Studiengängen zuständig. Diese Art von Nutzer verwenden die Software sehr intensiv und müssen sehr effizient arbeiten, da sie die Software jeden Tag intensiv mit vielen Inhalten nutzen. Die Modellierungsinhalte, welche mit der Software getätigt werden sind sehr komplex und erfordern ein hohes Maß an Konzentration. Damit man diesen Wünschen gerecht werden kann, muss unsere Lösung viel Lerninhalt in kurzer Zeit aufbereiten und für den Nutzer sinnvoll erklären. Natürlich kann die Hilfe nur einen kleinen Teil der Inhalte lernen. Die Basis sollte weiterhin eine intensive Schulung sein, bevor der Nutzer die Software verwendet. Die adaptive Online-Lernhilfe sollte den Nutzer bei der Lösungsfindung in diesem Fall unterstützen und ihm helfen, die Inhalte der Schulung evtl. nachträglich noch einmal zu wiederholen und zu vertiefen.

### 4.1.5 Admins

Eine weitere Rolle, die nicht missachtet werden sollte sind die Administratoren der Software. Diese sind für die Konfiguration der einzelnen Bereiche in der Software zuständig und normalerweise erster Ansprechpartner, wenn Probleme bei der Verwendung der Software auftreten. Der Admin muss normalerweise entscheiden, ob es ein Software-Fehler ist oder ein anderes Problem besteht. Bei kleineren Hochschulen kann diese Rolle auch wegfallen, da nicht genügend Kapital für die Finanzierung einer solchen Stelle vorhanden ist. Da die Zielsoftware für größere Hochschulen mit teils sehr technischen Schwerpunkten (Informatik, Maschinenbau, etc.) konzipiert ist, ist anzunehmen, dass es gesonderte Stellen zur Administration der Software gibt. Die Schwerpunkte hier beim eigentlichen Verwenden

der Software liegen in dem Deployment von Updates oder dem Lösen von Konfigurationsproblemen. Für diese Zielgruppe soll das Hilfesystem vorwiegend schnell Informationen bereitstellen, so dass der Admin unverzüglich weiterarbeiten kann.

## 4.2 Nutzer Klassifizierung

Nachdem die einzelnen Personas vorgestellt wurden, folgt die Klassifizierung in verschiedene Nutzerklassen. Die Klassifizierung erfolgte anhand der in Abbildung 4.1 sichtbaren Kriterien: schnelle, effektive Nutzung sowie lösungsorientierte und informationsorientierte Nutzung. Unter informationsinteressiert verstehen wir eine Nutzung der Software um an Informationen wie Stundenplan oder Rauminformation schnell zu gelangen. Diese Kriterien haben wir den jeweiligen Rollen zugeordnet. Durch die Verknüpfung in einem ungerichteten Graphen sind wir zu der Nutzerklassifizierung gekommen.

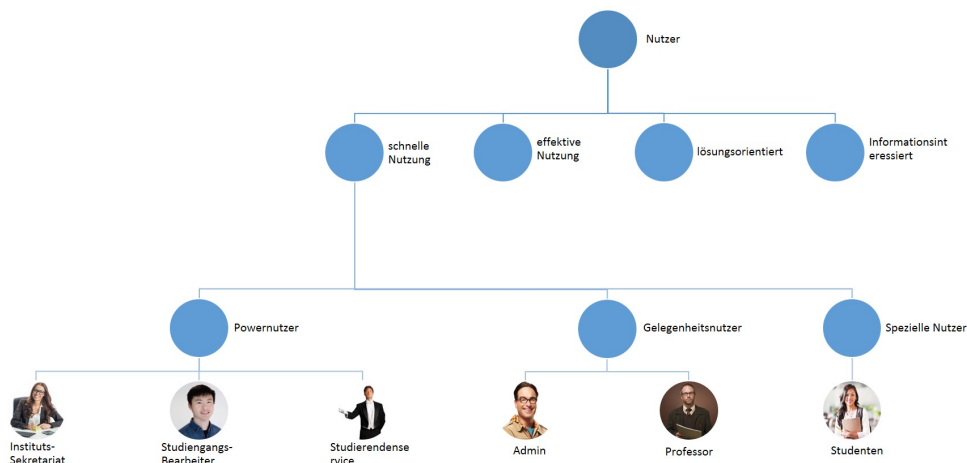


Abbildung 4.1: Nutzerklassifizierung abgeleitet aus Nutzungskriterien

### 4.2.1 Pownutzer

Diese Nutzergruppe verwendet die Software meist jeden Tag und dies mehrere Stunden täglich. Dies bedeutet es müssen sehr viele Inhalte erlernt werden. Wenn möglich in möglichst kurzer Zeit. Dies erfordert ein Lernen von mehreren Wegen zur Lösung des Problems. Für diese Art von Nutzern zählt eine effektive Nutzung der Software. Die Lerninhalte müssen somit möglichst umfassend sein und entsprechend aufbereitet werden.

### 4.2.2 Gelegenheitsnutzer

Der lösungsorientierte Nutzer möchte schnell die Lösung zu seinem aktuellen Problem haben. Er möchte sich nicht lange mit Online-Hilfesystemen oder langen Suchen beschäftigen.

Zu dieser Gruppe gehören vor allem die Gelegenheitsnutzer. Diese verwenden die Software selten und sind deshalb wenig angetan, wenn sie lange in einem bestimmten Problem stecken. Ihnen muss schnell geholfen werden, so dass kein Frust beim Verwenden der Software aufkommt. Hier läuft die Lösung nach dem Prinzip ab, dass ein Problem beim Verwenden der Software auftritt und schnellstmöglich eine Lösung her muss.

### 4.2.3 Spezielle Nutzer

Die Studenten erhalten bei uns eine extra Klasse. Sie werden als sogenannte "spezielle Nutzer" eingestuft. Sie setzen die Software sehr sporadisch ein und möchten nicht viele neue Inhalte lernen, da sie schon mit anderen Themen ihres Studiums sehr beschäftigt sind. Wenn neue Inhalte für die Verwendung der Software gelernt werden müssen, dann mit Spaß verbunden. Sie benötigen somit eine extra Motivation um neue Inhalte der Software zu lernen. Weiterhin schwierig bei dieser Nutzergruppe ist die Diversität der Nutzertypen. Dies bedeutet, dass viele verschiedene Fachbereiche, von sehr technisch versierten Leuten wie Ingenieursstudiengänge bis stark intellektuelle Studiengänge wie Germanistik oder Jura bedient werden müssen. Diese starke Diversität veranlasst uns die Gruppe der Studenten in die Klasse "spezielle Nutzer" einzuordnen.

## 5. Lösungsansätze

Dieses Kapitel widmet sich den Lösungsansätzen, die als Grundlage für unsere abschließende Empfehlung dienen. Es handelt sich dabei um verschiedene Ansätze, die aufgrund unserer Nutzerklassifizierung und deren Ziele entstanden sind. Bevor wir die Lösungen vorstellen wollen wir auf die Bewertungskriterien eingehen, die für eine abschließende Bewertung der Lösungen von Bedeutung sind.

### 5.1 Bewertungskriterien

Abbildung 5.1 zeigt unsere Bewertungskriterien, die im Folgenden erläutert werden sollen.

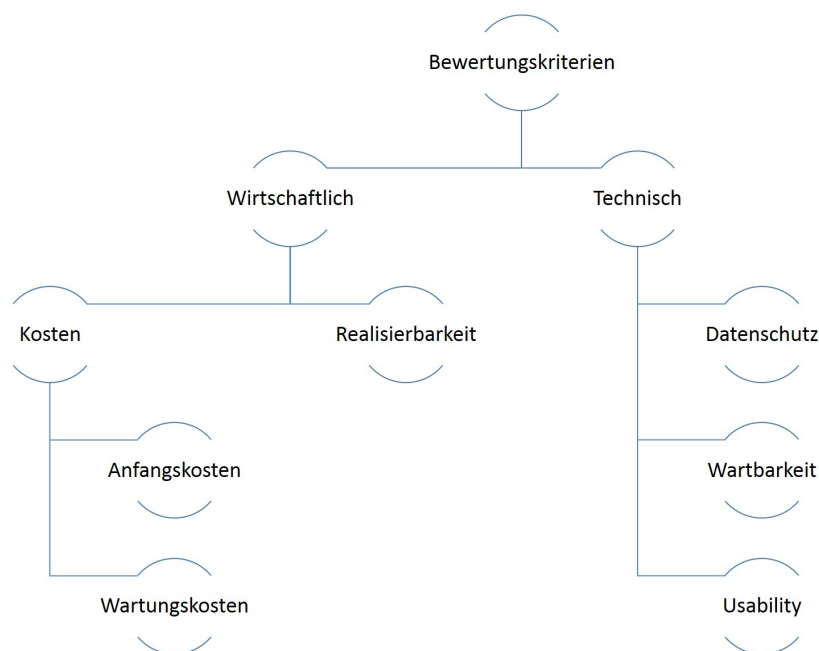


Abbildung 5.1: Bewertungskriterien für die Lösungsansatzbewertung



### 5.1.1 Begründung der Kriterien

Die Kriterien lassen sich grob in 2 Kategorien einteilen: wirtschaftliche und technische Kriterien. Zu den wirtschaftlichen Kriterien zählen die Kosten, die noch einmal in Anfangskosten und Wartungskosten unterteilt werden, und die Realisierbarkeit der einzelnen Lösungsansätze. Zu den technischen Kriterien haben wir die Wartbarkeit, den Datenschutz und die Usability gezählt.

### 5.1.2 Wirtschaftlich

#### 5.1.2.1 Anfangskosten

Unter den Anfangskosten verstehen wir die alle Kosten die anfallen, bis der Lösungsansatz in CAS Campus verfügbar und einsetzbar gemacht werden kann. Diese setzen sich aus Sachkosten und Personalkosten zusammen. Zu den Sachkosten zählen zum Beispiel die benötigte Hardware, Lizenzen für bestimmte Software, usw. Da bei der Softwareentwicklung aber der kosten treibende Faktor die Personalkosten sind, werden die Sachkosten im Folgenden nicht näher betrachtet. Hinzuzufügen ist, dass die Personalkosten sich am Aufwand des Projekts orientieren. Man spricht also von einem Personalaufwand für ein bestimmtes Projekt. Um diesen Personalaufwand zu abzuschätzen gibt es bereits verschiedene etablierte Methoden, die jedoch von verschiedenen Faktoren wie dem Vorgehensmodell (z.B. Wasserfallmodell vs. SCRUM) (Bunse, o. J.) oder der Art der Schätzung (Einfache Schätztechniken wie zu erwartende Lines of Code (LOC) anhand von bekannten LOCs berechnen vs. Algorithmische Aufwandsschätzungen wie COCOMO) (Hummel, 2011). Generell kann man dazu aber sagen, dass eine Schätzung ungenauer wird, sofern sich der abzuschätzende Zeitraum vergrößert. Die lässt sich mit dem, in der Softwaretechnik anerkannten, sogenannten Kegel der Unsicherheit (engl. Cone of Uncertainty, siehe Abbildung 5.2) einfach darstellen. Dabei bildet die x-Achse die Phase eines Projekts und die y-Achse den relativen Aufwand eines Projekts ab.

#### 5.1.2.2 Wartungskosten

#### 5.1.2.3 Nutzen

### 5.1.3 Technisch

Da Software ein sehr technisches Gut ist, müssen technische Bewertungskriterien zur Bewertung der Lösungsansätze verwendet werden. Im Folgenden sollen unsere Kriterien für die Arbeit aufgezeigt werden.

#### 5.1.3.1 Wartbarkeit

Die Wartbarkeit einer Software kann mithilfe der sogenannten internen Softwarequalität beschreiben (?). Die interne Software-Qualität bewertet die Software anhand von der gesamten Wartbarkeit des Systems. Hier geht es insbesondere darum, wie schnell es geht den vorhandenen Code umzustrukturieren und wie das Design anpassbar ist, so dass es auf zukünftige Anforderungen der Software reagieren kann. In unserem Umfeld ist es wichtig,

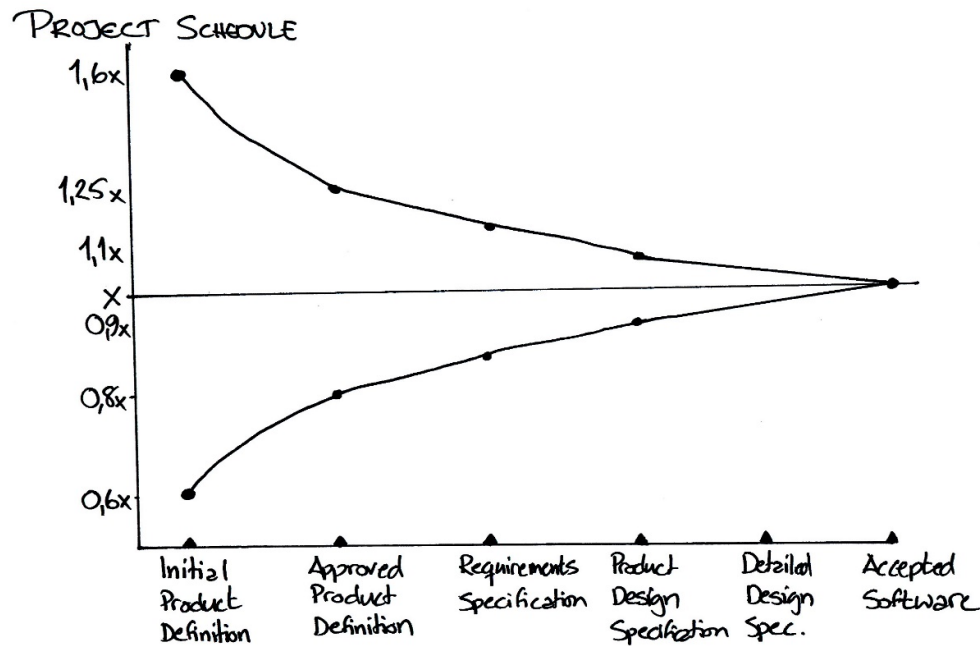


Abbildung 5.2: Kegel der Unsicherheit

dass die Lösung dem Zielsystem Software passend gemacht wird. Im Zuge der Wartbarkeit ist es wichtig, das für neue Hilfethemen, die aufgrund von neuen Anforderungen hinzukommen das vorhandene Hilfesystem schnell angepasst werden kann, so dass die Kosten im Rahmen gehalten werden kann.

### 5.1.3.2 Datenschutz

Aufgrund der aktuellen politischen Lage mit der NSA-Affäre müssen heutige Softwaresysteme sehr sorgfältig mit dem Thema Datenschutz umgehen. Jedes System, was Informationen vom Nutzer sammelt, wird hier kritisch bewertet, da die Daten ein wichtiges Gut sind. In unserer Arbeit ist die Informationsbereitstellung des Nutzers sehr wichtig, da anhand der bereitgestellten Informationen die Hilfeinhalte aufbereitet werden um eine Adaption zu gewährleisten. Die entwickelten Lösungsansätze müssen einer kritischen Prüfung zum Thema Datenschutz standhalten.

### 5.1.3.3 Usability

Das Thema Usability ist ein wichtiges Qualitätsmerkmal von Software (?). Ohne eine vernünftige Usability ist die Akzeptanz der Software sehr gering und der Nutzer möchte die Software nur ungern verwenden. Weiterhin möchte der Nutzer bei der Benutzung schnell zu seinem Ziel kommen und die Aufgabe möglichst effizient bearbeiten. Heutige Software kann komplexe Prozesse einfach abbilden. Dies ist das Hauptziel der Usability. Schwierige Sachverhalte einfach aufzubereiten, so dass sie jeder verstehen kann. In unserer Arbeit müssen aus diesem Grund die Lösungsansätze mit der Usability bewertet werden.

## 5.2 Klassische Hilfe

Das klassische Hilfesystem soll unserer Klasse der Problemorientierten Nutzer weiterhelfen. Durch eine schnelle Lösungsfindung und Aufbereitung soll dem Nutzer möglichst schnell und effektiv geholfen werden.

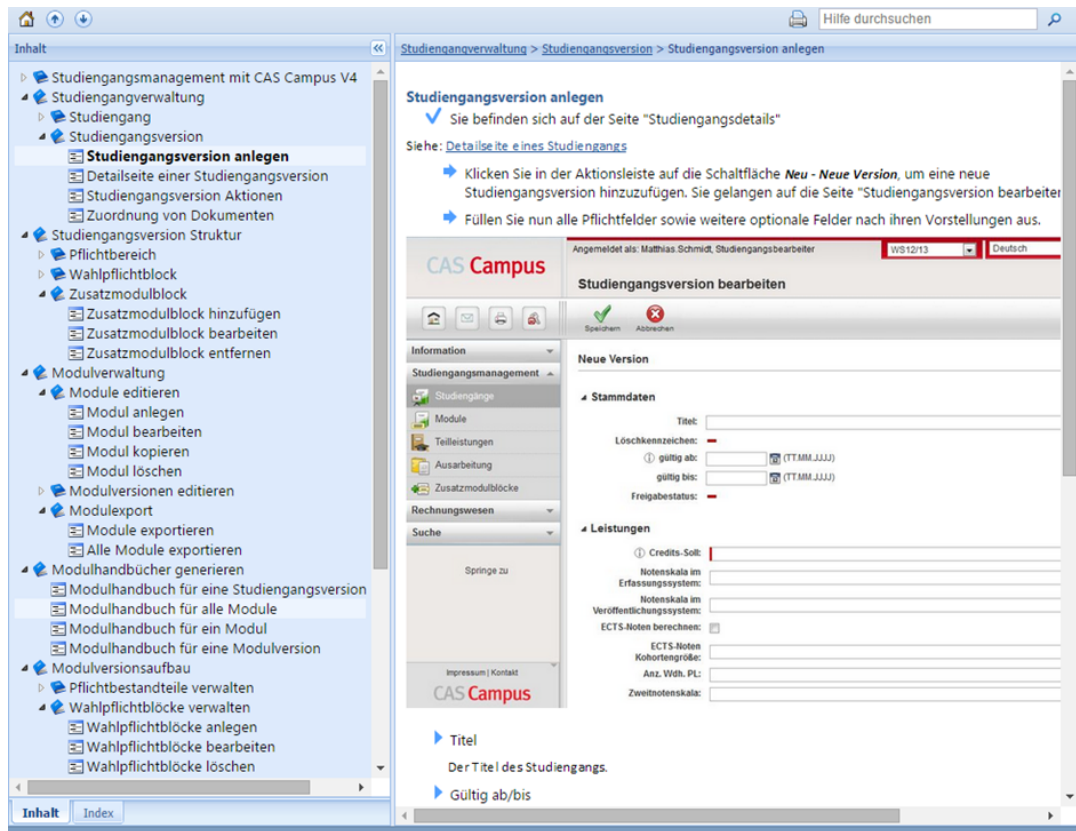


Abbildung 5.3: Klassischer Aufbau eines Hilfesystems

### 5.2.1 Erläuterung

Die klassische Hilfe entspricht, wie der Name bereits vermuten lässt, einem Hilfesystem wie man es bereits aus mehreren Programmen kennt. Ein Hilfesystem ist die „*Gesamtheit an Informationen, die gegeben werden, um dem Benutzer beim Umgang mit [Anwendungs]programmen zu helfen*“ (Duden, 2015). Es handelt sich im Falle der Abbildung um ein Handbuch welches online abgerufen werden kann. Der Inhalt der Hilfe wird als grafisch aufbereiteter Text dargestellt und mit Bildausschnitten aus dem Programm ergänzt. Die aufgerufene Hilfeseite wird wie in Abbildung 5.3 auf der rechten Seite des Fensters dargestellt. Die linke Seite des Fensters enthält die einzelnen Hilfeseiten, nach Inhalt in diverse Kategorien geordnet. Eine Indexierung der Hilfeseiten ist ebenfalls vorhanden und lässt sich über einen weitere Registerkarte im linken Bereich einblenden. Die Suchfunktion nach Inhalt oder Indexeinträgen befindet sich in der Kopfzeile des Fensters. Diese Form der Hilfe bietet die Möglichkeit zu einem konkreten Problem die genaue Lösung zu erhalten. Dieser Ansatz funktioniert am Besten, wenn der Nutzer sein konkretes Problem kennt und

auch entsprechend beschreiben kann. Dies ist bei Software sehr oft gegeben, was bedeutet ein Nutzer hängt gerade in einer Anwendung und weiß wonach er genau suchen muss.

## 5.3 Lernspiel-App

Der zweite Lösungsansatz soll eine Lösung in Form einer sogenannten Lernspiel-App aufzeigen. Zuerst wird eine Begriffsdefinition von App durchgeführt bevor die konkrete Lösung vorgestellt wird.

### 5.3.1 App

Der App-Begriff bezieht sich in unserer Arbeit auf einen kleinen Prozess, der mit einer Software durchgeführt werden kann. Dieses Prinzip kann man am Besten anhand von Unix-Systemen erklären. Hier gibt es für jedes kleine Teilproblem, welches bei der Verwendung eines Computers eintritt ein extra Programm. Dieses Programm macht genau die Aufgabe, die es soll und keine zusätzlichen Extra-Funktionen. Ein gutes Beispiel ist ein Texteditor. Bei einer grafischen Oberfläche beherrscht dieser beispielsweise die Funktion suchen und ersetzen. Weiterhin kann der Editor beispielsweise bestimmte Textfelder rot markieren und eine Datei von dem geschriebenen anlegen. In einer Unix-Konsole gäbe es für alle diese Funktionen einen extra Befehl, der genau diese eine Funktion erfüllt. Dies hat den Vorteil, dass sehr effizient gearbeitet werden kann. Dieses Prinzip verwenden die modernen Apps auf dem Smartphone ebenfalls. Für jeden Anwendungsfall gibt es genau eine App. Das Smartphone besitzt somit eine Ansammlung mehrerer Apps zur Lösung von mehreren Teilproblemen. Eine solche Plattform ist in Abbildung 5.4 abgebildet. Auf der linken Seite befinden sich die Icons von mehreren Apps. Auf der rechten Seite ist der konkrete Prozessworkflow, in diesem Beispiel eine Kontaktliste sichtbar.

### 5.3.2 Lernspiel

Unter einem Lernspiel verstehen wir die Verknüpfung von Gamification-Ansätzen mit dem Erlernen von neuen Inhalten. Hier sind Konzepte für ein umfangreiches Belohnungssystem von Interesse. Ein Nutzer kann eine Belohnung erhalten wenn er eine bestimmte Aufgabenstellung lösen kann. Diese Aufgabe bezieht sich immer auf das Software-Feld. Hierzu erhält der Nutzer ein gezieltes Problem, welches er eventuell auf verschiedenen Wegen lösen kann. Dieser Prozess des Lernens soll durch eine Belohnung am Ende motiviert werden. Erreicht der Nutzer ein bestimmtes Lernziel, so erhält er beispielsweise einen sogenannten Badge (kleine Auszeichnung) oder erreicht das nächste Level, wo ihm ein komplexeres Problem erwartet.

### 5.3.3 Lösungsansatz

Die Verbindung vom App-Prinzip mit dem Lernspiel soll unser zweiter Lösungsansatz repräsentieren. Das daraus folgende Konzept soll eine App designen, die komplexe Inhalte unserer Software in einem Spiel aufbereitet. Dieses Spiel besteht aus mehreren Levels, die unterschiedliche Komplexitätsgrade beherbergen. Der Schwierigkeitsgrad der Lernspiel-App hängt von den Erfahrungen des Nutzers ab, der das Spiel durchführt. Wenn keine

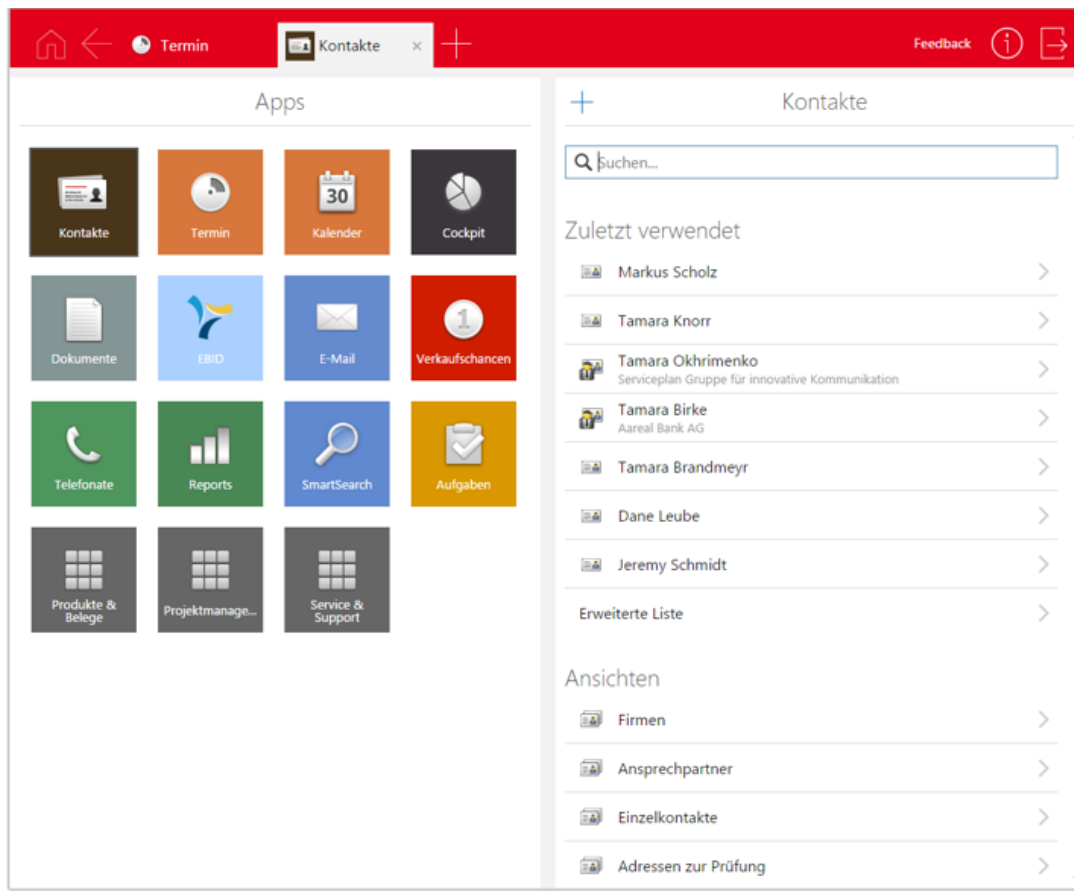


Abbildung 5.4: App-Plattform und Prozessworkflow einer App

Erfahrungen von seitens des Nutzers bestehen, so fängt er bei der einfachsten Problemstellung an und arbeitet sich durch mehrere Levels. Hierbei muss aufgepasst werden, dass der ganze Ansatz nicht zu verspielt ist, da die problembasierten Nutzer sich nicht lange mit so einem simplen Spiel aufhalten wollen. Für die Studentengruppe hingegen soll der Spaß im Vordergrund stehen weshalb eine spannende Aufbereitung der Inhalte wichtig ist. Hier spielen besonders Inhalte aus dem Game-Based Learning eine wichtige Rolle. Die Lerninhalte müssen in eine spannende Form gepackt werden, damit dem Anwender nicht langweilig wird. Für die effektiven Nutzer, bzw. Powernutzer muss die Lernspiel-App viele Problemstellungen innerhalb der Software behandeln und möglichst viele Levels anbieten. Weiterhin muss für die Nutzer ebenfalls ein großer Grad an Diversität erreicht werden. Die Diversität besteht in der Aufbereitung von mehreren Bereichen innerhalb der Software. Also von einem Bereich zum Erfassen neuer Adressdatensätze bis hin zum Modellieren von komplexen Studiengängen.

## 5.4 Expertensystem

Unser sogenanntes Expertensystem beruht auf einem Recommendersystem. Es soll, wie in einem Recommender, Empfehlungen von Nutzern generieren. Diese Empfehlungen werden direkt von einem Nutzer zum Nächsten weitergegeben. Wichtig hierbei ist die Identifikation

des Experten der die Empfehlungen abgibt. Diese Identifikation soll anhand der Lernspiel-App erfolgen. Wer hier gut abschneidet, also eine schnelle und effektive Lösung für das gestellte Problem findet, der kann sich durch Badges einen Expertenstatus erarbeiten. Dieser Status gilt immer für einen bestimmten Bereich der Software, also einer App und einem dazugehörigen Workflow. Der Lösungsansatz muss die Information, wer ein solcher Experte für einen bestimmten Bereich ist, bereitstellen. Unsere Recommender-Empfehlung besteht aus der Anzeige eines Experten für einen bestimmten Bereich innerhalb der Software. Aufgrund dieser Anzeige kann ein Nutzer, der gerade Hilfe benötigt den Experten identifizieren und ihn um Rat bitten. Diese Frage kann er dann entweder per E-Mail oder direkt per Telefon durchführen. Eine weitere Möglichkeit zum Erlangen des Expertenstatus kann durch das Manuelle Einstellen durch einen Administrator sein, der im Backend dies einstellt.

## 6. Empfehlung

Unsere abschließende Empfehlung soll die Lösungsansätze anhand der Nutzerklassen und unserer Kriterien bewerten. Hierzu erfolgt jeweils eine Gegenüberstellung.

### 6.1 Bewertung der Lösungsansätze anhand der aufgestellten Nutzerklassen

### 6.2 Bewertung der Lösungsansätze anhand der aufgestellten Kriterien

### 6.3 Konkrete Empfehlung

Unsere Empfehlung sieht eine Kombination der beiden Lösungsansätze Lernspiel-App und Expertensystem vor. Mithilfe dieser Kombination sehen wir die Ziele der einzelnen Nutzergruppen am besten zufriedengestellt. Eine klassische Online-Lernhilfe besteht im Software-Zielsystem bereits. Aus diesem Grund bietet der Lösungsansatz nicht neues. Die Lernspiel-App hingegen eignet sich hervorragend um komplexe Sachverhalte in der Software auf eine spielerische Art zu erlernen. Wer sich hierbei als besonders gut hervortut, der kann im Anschluss dann als Experte für ein Expertensystem genommen werden. Das Expertensystem bietet den Hauptvorteil, dass dem Nutzer bei einem konkreten Problem schnell geholfen werden kann. Beim Einsatz des Expertensystem ist wie gesagt die Zeitkomponente wichtig, da ein Nutzer der Software nicht ständig mit Anfragen von anderen Nutzern gestört werden möchte, muss diese Lösung auch von der Universität bei der die Lösung zum Einsatz kommt getragen werden, damit der Ansatz funktioniert. Somit die konkrete Empfehlung wäre eine oder mehrere Lernspiel-Apps, die komplexe Inhalte der Software für den Nutzer einfach aufbereiteten, so dass er schnell viel neues erlernen kann. Wer besonders kreative Lösungen für die Problemstellung in der App findet, der eignet sich als späterer Experte, der bei konkreten Problemen in der Verwendung der Software unterstützen kann.

## 7. Zusammenfassung

Die Aufgabe dieser Seminararbeit war es, ein adaptives Online-Hilfesystem zu entwickeln. Gestartet haben wir mit der Frage nach der Adaption. Hier haben wir uns, durch die Vorgabe unserer Betreuer, auf die Adaption an den Nutzer konzentriert. Nachdem wir alle Nutzerrollen aufgezeigt haben, konnten wir eine Klassifizierung der Nutzer durchführen. Diese Klassifizierung hat uns geholfen auf den Nutzer passende Lösungsansätze zu entwickeln. Diese sind durch verschiedene Grundlagen des problembasierten und differenziellen Lernens entstanden. Weiterhin konnten wichtige Ansätze der Gamification in einer App-Lösung erarbeitet werden. Eine weitere Lösung ist durch eine Recommendersystem-Grundlage entstanden. Damit ein besserer Vergleich stattfinden konnte, wurde die bisherige Lösung eines klassischen Hilfesystems vorgestellt. In der folgenden Gegenüberstellung der Lösungsansätze wurden die Lösungen zuerst mit den Nutzerklassen und anschließend nach unseren selbst auferlegten Kriterien bewertet. Die daraus folgende Bewertung hat gezeigt, dass eine Kombination von allen drei Lösungsansätzen sinnvoll ist und durchgeführt werden kann. Eine klassische Lernhilfe besteht bereits und wird durch die beiden zusätzlichen Lösungsalternativen Lernspiel-App und Expertensystem ergänzt.



## 8. Ausblick

Für eine konkrete Ausarbeitung unserer Empfehlung ist eine Ausarbeitung eines Konzeptes basierend auf unseren Lösungsansätzen der nächste Schritt. Die aufgezeigten Konzepte sollen hierbei in einen konkreten Software-Workflow implementiert werden, so dass eine Lernspiel-App entstehen kann. Das Ziel dieser App soll es sein, die Nutzer zu schulen und ihnen neue Lerninhalte der Software zu vermitteln. Weiterhin sollen, darauf aufbauend, führende Experten identifiziert werden, die anderen Nutzern bei konkreten Problemen bei der Verwendung der Software schnell weiterhelfen können.



## Literatur

(o. J.).

Bendel, P. D. O. (2015). Gamification. Zugriff auf <http://wirtschaftslexikon.gabler.de/Archiv/688938796/gamification-v4.html>

Bunse, v. K. A., C. (o. J.). *Vorgehensmodelle kompakt*. Spektrum Akademischer Verlag. Zugriff auf <https://books.google.de/books?id=X47DrRaHJNkC>

Duden. (Zugriff am 20.07.2015a). adaptieren. Zugriff auf <http://www.duden.de/node/687662/revisions/1384857/view>

Duden. (Zugriff am 20.07.2015b). Lernhilfe. Zugriff auf <http://www.duden.de/node/733105/revisions/1232018/view>

Duden. (Zugriff am 20.07.2015c). Onlinehilfe. Zugriff auf <http://www.duden.de/node/769414/revisions/1344323/view>

Hummel, O. (2011). *Aufwandsschätzungen in der software- und systementwicklung kompakt* (Bd. 1.Hrsg.). Spektrum Akademischer Verlag.

onlinemarketing praxis.de. (2015). onlinemarketing praxis - personas. Zugriff auf <http://www.onlinemarketing-praxis.de/glossar/personas>Zugriffam20.07.2015

Wolfgang I. Schöllhorn, D. J. M. M., Hendrik Beckmann. (2009, Februar). Differenzielles lehren und lernen im sport. , Zugriff am 16.07.2015. Zugriff auf <http://www.hofmann-verlag.de/project/zsarchiv/archiv/sportunterricht/2009/sportunterricht-Ausgabe-Februar-2009.pdf#page=6>

Zumbach, J. (o. J.). *Problembasiertes lernen*. Waxmann Verlag. Zugriff auf <https://books.google.de/books?id=X47DrRaHJNkC>