



Duale Hochschule Baden-Württemberg  
Mannheim

## **Seminararbeit**

# **Dokumentation zum Unternehmensplanspiel "Avalon"**

Studiengang Wirtschaftsinformatik Software Engineering

Bearbeitungszeitraum: 18.11.2013 bis 14.02.2014

### **Verfasser**

Martin Achenbach  
Jonas Braun  
Johannes Haaß  
Frederik Pahde

### **Kurs Studiengangsleiter**

WISE12B  
Prof. Dr. Thomas Holey

### **Ausbildungsfirma**

SAP AG  
Dietmar-Hopp-Allee 16  
Walldorf

### **Dozent**

Gregor Tielsch

## **Abstract**

# Inhaltsverzeichnis

	Seite
<b>Abkürzungsverzeichnis . . . . .</b>	<b>III</b>
<b>Abbildungsverzeichnis . . . . .</b>	<b>IV</b>
<b>1 Einleitung . . . . .</b>	<b>1</b>
1.1 Problemstellung und -abgrenzung . . . . .	1
1.2 Ziel der Arbeit und Vorgehensweise . . . . .	1
<b>2 Planung und Vorgehen . . . . .</b>	<b>2</b>
2.1 Projektmanagement . . . . .	2
2.1.1 Anforderungen . . . . .	2
2.1.2 Einrichtung von Trello und GitHub . . . . .	3
2.1.3 Praktische Erfahrungen mit Trello und GitHub . . . . .	4
2.2 Fachkonzept und Iterationen . . . . .	5
2.2.1 Erstellung des ersten Fachkonzepts . . . . .	5
2.2.2 1. Iteration . . . . .	6
2.2.3 2. Iteration . . . . .	6
2.2.4 3. Iteration . . . . .	7
2.2.5 Reflexion der Iterationsschritte . . . . .	7
2.3 Events . . . . .	8
<b>3 Anforderungsspezifikation . . . . .</b>	<b>10</b>
3.1 Zielbestimmung . . . . .	10
3.1.1 Muss-Kriterien . . . . .	10
3.1.2 Kann-Kriterien . . . . .	10
3.1.3 Abgrenzungskriterien . . . . .	11
3.2 Einsatz . . . . .	11
3.3 Umgebung . . . . .	11
3.4 Funktionalität . . . . .	11
3.5 Daten . . . . .	12
3.6 Leistungen . . . . .	12
3.7 Benutzeroberfläche . . . . .	12
3.8 Qualitätsziele . . . . .	13

<b>4</b>	<b>Vorgehen und Iterationen . . . . .</b>	<b>14</b>
<b>5</b>	<b>Zusammenfassung und Ausblick . . . . .</b>	<b>15</b>
<b>6</b>	<b>Planung und Vorgehen . . . . .</b>	<b>16</b>
<b>7</b>	<b>Einleitung . . . . .</b>	<b>17</b>
<b>8</b>	<b>Markt . . . . .</b>	<b>18</b>
8.1	Definition . . . . .	18
8.2	Geschichte und Entwicklung . . . . .	18
8.3	Aktuelle Situation . . . . .	19
8.3.1	Besonderheiten . . . . .	22
<b>9</b>	<b>Spielprinzip . . . . .</b>	<b>24</b>
9.1	Abteilungen . . . . .	24
<b>10</b>	<b>User Interface . . . . .</b>	<b>27</b>
<b>11</b>	<b>Planung und Vorgehen . . . . .</b>	<b>29</b>
	<b>Literaturverzeichnis . . . . .</b>	<b>i</b>
	<b>Anhang . . . . .</b>	<b>i</b>
	<b>Ehrenwörtliche Erklärung . . . . .</b>	<b>ii</b>

# **Abkürzungsverzeichnis**

# Abbildungsverzeichnis

8.1 Statistik Absatzprognose . . . . .	19
8.2 Kaufkriterien: 2012 (hellblau)   2013 (dunkelblau) . . . . .	20
10.1 erstes Mockup . . . . .	27
10.2 zweites Mockup . . . . .	28
10.3 Screenshot . . . . .	28

# **1 Einleitung**

## **1.1 Problemstellung und -abgrenzung**

## **1.2 Ziel der Arbeit und Vorgehensweise**

## **2 Planung und Vorgehen**

Bei der Planung und Entwicklung von Avalon soll Effektivität und Zielstrebigkeit eine wichtige Rolle spielen. Deshalb war es uns von Beginn an wichtig, genügend Zeit in die Planung zu investieren. Während der ersten Planungsphase haben wir entschieden, das Projektmanagement softwareseitig zu unterstützen um die Organisation und die Zusammenarbeit untereinander zu vereinfachen sowie den zeitlichen Aufwand für die Organisation zu minimieren.

Außerdem haben wir uns entschieden die Entwicklung nach den Grundsätzen der agilen Softwareentwicklung durchzuführen, die wir in Semester 2 in einer Vorlesung kennengelernt haben. Ein Grundbaustein der agilen Softwareentwicklung ist es zu Beginn zunächst ein Fachkonzept zu erstellen und im weiteren Verlauf der Entwicklung mehrere Iterationen durchzuführen. Mit diesen Iterationen sollen Fehler leichter aufgedeckt werden und Prioritäten für zentrale Aspekte von Avalon gesetzt werden können.

### **2.1 Projektmanagement**

#### **2.1.1 Anforderungen**

Für die softwareseitige Unterstützung des Projektmanagements haben wir zunächst Anforderungen herausgearbeitet. Für uns ergaben sich drei zentrale Anforderungen an diese Software.

Die erste Anforderung ist es einen Gesamtüberblick über den aktuellen Stand des Projektes darzustellen. Dazu soll eine Aufgabenliste angezeigt werden, bei der sich Aufgaben auch an einzelne Personen zuordnen lassen. Diese Aufgabenliste soll einem Backlog entsprechen, wie man es aus dem Scrum-Prozess kennt.

Die zweite Anforderung ist eine zentrale Codeverwaltung. Damit soll erreicht werden das die Teammitglieder immer Zugriff auf den aktuellen Code haben und Erkennbar ist wer welche Veränderungen vorgenommen hat. Zudem soll die Codeverwaltung auch eine Versionsverwaltung bieten um gegen Datenverluste jeglicher Art vorzubeugen.



Die dritte Anforderung bezieht sich auf die Anforderungen 1 und 2. Da wir oft von unterschiedlichen Orten und Rechnern arbeiten, sollen die Daten überall und jederzeit verfügbar sein. Um diese Anforderung zu erfüllen lag es nahe, eine Lösung zu nutzen, bei der die Daten, online, in einer Cloud gespeichert werden. Aufgrund von persönlichen Erfahrungen aus vergangenen Projekten und Internet Recherchen haben wir entschieden, dass wir zwei verschiedene Dienste einsetzen um diese Anforderungen abzudecken. Für den Aufgabenübersicht werden wir den Online-Dienst *Trello* einsetzen und für die Codeverwaltung *GitHub* benutzen. Beide Tools erfüllen die oben genannten Anforderungen vollständig.

### 2.1.2 Einrichtung von Trello und GitHub

Die Einrichtung von Trello ist sehr einfach, da keine Software installiert werden muss und alles im Browser läuft. Zudem gibt es auch eine App für Smartphones, mit der man auch Unterwegs immer einen Überblick bekommen kann. Zur Einrichtung von Trello ist nur eine Registrierung auf der Webseite und die Erstellung eines gemeinsamen *Boards* notwendig. Ein solches Board besteht aus beliebig vielen Spalten, denen die konkreten Aufgaben untergeordnet sind. Die Spalten dienen dazu das ganze Board zu gliedern. Am besten lässt sich dieses Prinzip anhand der folgenden Tabelle verstehen, welche den Aufbau unseres Boards beispielhaft darstellt.

To-Do	Doing	Done	Idea's
Kaufalg.	Use-Cases	Marktanalyse	Historie
...	...	...	...

Auch die Einrichtung von GitHub ist recht einfach. Auch hier müssen sich die Mitglieder auf der GitHub Homepage registrieren. Danach erstellt ein Mitglied ein neues Repository und fügt die anderen Mitarbeiter als Collaborators hinzu. Zur Nutzung ist es noch notwendig *Git* oder die *GitHub-GUI* auf den einzelnen Rechnern zu installieren. Dieses Tool benötigt man um das aktuelle Repository herunterzuladen und Änderungen als Commit hochzuladen. Da in unserem Team

nur sehr geringe Erfahrungen mit Git vorhanden waren, entschieden wir uns nur die GitHub-GUI zu nutzen.

### **2.1.3 Praktische Erfahrungen mit Trello und GitHub**

Während Entwicklung von Avalon stellte sich besonders Trello als sehr hilfreiches, einfaches und fehlerfreies Tool dar. Die Funktionsweise wurde jedem sofort verständlich und war selbst erklärend. Mit diesen Eigenschaften konnte Trello immer einen guten Überblick über den aktuellen Stand liefern und erfüllte die von uns gestellten Anforderungen sehr gut.

GitHub dagegen lief nicht ganz so unproblematisch. Die Funktionsweise über die GUI war zwar sehr logisch und einleuchtend, allerdings kam es regelmäßig zu teils merkwürdigen Fehlern. Das größte Problem entstand, wenn zwei Mitglieder die gleiche Datei bearbeiteten. Das Zusammenführen der beiden Änderungen (engl. mergen) über die GUI war nie möglich und manuelles Mergen über die Konsole funktionierte nur selten. Um dieses Problem im praktischen Einsatz schnell und einfach zu lösen, hat einer der beiden Betreffenden seine Änderungen separat gespeichert und das Repository neu heruntergeladen. Diese Fehler führte oft zu Frustrationen innerhalb des Teams. Der Grund für diese Fehler lag sowohl an unsere Unerfahrenheit mit Codeverwaltung und an der GitHub GUI, da diese anscheinend bestimmte Befehle nicht korrekt ausführt.

Ein weiterer frustrierender Fehler war das Überschreiben von einigen Dateien mit älterem Inhalt. Der Grund hierfür lag an der falschen Konfiguration unseres Repositories. Allerdings konnten wir aus Zeitgründen diesen Fehler nicht beheben. All diese Probleme führten immer wieder zu Überlegungen GitHub durch ein anderes Tool, wie Dropbox, zu ersetzen. Diese Überlegungen wurden jedoch wieder verworfen, da damit vermutlich noch weitere Probleme entstanden wären. Im Nachhinein wäre es besser gewesen die Git-Bash zu verwenden und vor Projektbeginn eine ausführliches Tutorial zu Git zu machen.

## **2.2 Fachkonzept und Iterationen**

Aufgrund der Zeitspanne von drei Monaten über welche dieses Projekt verlief, haben wir uns entschieden zunächst ein Fachkonzept zu erstellen und darauf aufbauend drei Iterationen durchzuführen. Nach den Prinzipien der agilen Softwariemethodik sind die Übergänge zwischen diesen Iterationen in beide Richtungen offen. Dies war für uns sehr essentiell, um Ergänzungen vorzunehmen und um bestimmte Aspekte im Nachhinein noch konkretisieren zu können.

### **2.2.1 Erstellung des ersten Fachkonzepts**

Bevor wir mit den Iterationen beginnen konnten, war es zunächst notwendig das Fachkonzept zu planen und zu erstellen. Als Grundlage für diese wurde zunächst eine Marktanalyse durchgeführt, um die zentralen Merkmale des Smartphone-Marktes herauszuarbeiten. Hierbei spielten auch unsere eigene Erfahrungen mit Smartphones im Unternehmen und im privaten Umfeld eine entscheidende Rolle. Aus diesen Erkenntnissen konnte ein erstes Modell unseres Spiel grob skizziert werden und die grundlegende Spielidee festgemacht werden. Diese Erkenntnisse wurde in Form eines Pflichtenheft festgehalten und führten zur Erstellung der ersten Use-Cases.

Aus dem Pflichtenheft und den Use-Cases ließen sich die ersten Klassen identifizieren. Dabei wurden nach den Regeln der Objekt Orientieren Analyse (kurz: OOA) nur Klassen identifiziert, die für die Logik des späteren Spiels notwendig sind. Während dieser Phase ließen sich auch schon einige Assoziationen und Vererbungsstrukturen, wie zum Beispiel bei den Department Klassen, erkennen. Mit diesem grundlegenden Fachkonzept war es möglich den ersten Iterationsschritt einzuleiten. Während den Iterationen wurde das Fachkonzept an einigen Stellen noch um einige Klassen, Assoziationen und Vererbungen ergänzt und präzisiert. Zeitlich wurde das erste Fachkonzept innerhalb der ersten zwei Wochen des Projektes erstellt.

### **2.2.2 1. Iteration**

Die erste Iteration hatte das Ziel die zentralen Merkmale des Fachkonzept zu implementieren und somit ein Grundgerüst für die weitere Entwicklung aufbauen. Dazu wurden die wichtigsten Klassen aus dem Fachkonzept, wie zum Beispiel die Departments in Java programmiert. Komplexere Klassen, wie der Markt, wurden dabei bewusst ohne Inhalt implementiert. Während der Durchführung der ersten Iteration sind uns immer wieder einige Lücken und Fehler im Fachkonzept aufgefallen, die ein funktionierendes Spiel verhindert haben. Um die Vollständigkeit und die Richtigkeit zu gewährleisten haben wir in diesen Fällen das Fachkonzept um die fehlenden Merkmale ergänzt beziehungsweise die Fehler behoben.

Um nicht den Überblick zu verlieren wurde Parallel zur Implementierung der Klassen des Fachkonzepts bereits, soweit wie möglich, die ersten Unit-Tests geschrieben. Dieser Schritte hatte die Ambition, zu gewährleisten, dass der Quellcode fehlerfrei ist. Nachdem die wichtigsten Klassen implementiert waren, wurde der zweite Iterationsschritt eingeleitet.

### **2.2.3 2. Iteration**

Im zweiten Iterationsschritt wurden weitere Klassen aus dem Fachkonzept implementiert. Besonders wurde die Implementierung des Market und der Consumer Groups priorisiert, da diese ein sehr wichtiges Element von Avalon sind. Während dieser Phase wurde auch eine Excel-Tabelle erstellt, welche den Kaufalgorithmus in seinem vollen Umfang simuliert. Damit konnte die Funktionsweise des Algorithmus bereits vor der Implementierung sichergestellt und mit unterschiedlichen Parametern getestet werden.

Ein zweiter wichtiger Punkt in dieser Iteration war die Implementierung des Client/Server Modells und der Config-Datei. Somit ging diese Iteration über das Fachkonzept hinaus und fügte weitere Mechanismen hinzu, die für das fertige Spiel zwingend notwendig sind. Auch während dieser Iteration wurden, soweit wie möglich, Unit-Tests geschrieben um mögliche Fehler zu entdecken und zu beheben.

### **2.2.4 3. Iteration**

Die dritte und letzten Iteration hatte das Ziel die noch fehlenden Klassen des Fachkonzepts und die Benutzeroberfläche zu implementiert, sowie Fehler zu beheben. Zu den fehlenden Klassen des Fachkonzepts handelte es sich konkret, um die Zufallseignisse für Einzelunternehmen und alle Unternehmen. Die Benutzeroberfläche wurde in kleinen „Unter- Iterationen“ von einem ersten Prototyp bis hin zur vollständigen Oberfläche umgesetzt. Dabei war immer das Design der Mockup's das Ziel. Damit die Benutzeroberfläche funktionieren konnte, muss auch noch eine Schnittstelle implementiert werden, die den Datentransfer zwischen Fachkonzept, Kommunikation und Benutzeroberfläche sicherstellt.

Schließlich stand in der letzten Iteration noch die Fehlerbehebung an. Dazu wurden die noch fehlenden Unit-Tests implementiert und ausgeführt. Zudem wurde das Spiel unter realistischen Umständen gespielt. Durch diesen praktischen Test konnte größtenteils Fehler gefunden werden, die außerhalb des Fachkonzepts waren.

Durch diese zwei unabhängigen Testmethoden konnte gewährleistet werden, dass das Spiel den Qualitätskriterien entspricht und damit verbunden auch spielbar ist.

### **2.2.5 Reflexion der Iterationsschritte**

Die Einführung der Iterationsschritte war sehr Sinnvoll, da das Projekt so in kleine Teiletappen eingeteilt werden konnte. Die Übergänge in den Iterationsschritten waren bei uns sehr fließend. So kam es vor, dass ein Teammitglied bereits den nächsten Iterationsschritt bearbeitete. Mit diesen flexiblen Übergängen und den Iterationen konnten einzelne Aufgaben auch gut an die einzelnen Teammitglieder und deren Fähigkeiten zugeordnet werden. Zudem konnte durch die genaue definierten Ziele der Iterationsschritte immer ein Gesamtstatus des Projektes gewährleistet werden. Dies war eine sehr wichtige Entscheidungsbasis für langfristige Entscheidungen.

Zusammenfassend lässt sich sagen, dass die Effektivität der Entwicklung durch die Iterationsschritte gesteigert wurde und ohne diese Schritte wohl kein zufriedenstellendes Ergebnis entstanden wäre.

## 2.3 Events

Die Zufallsereignisse, die Einzelunternehmen, alle Unternehmen oder den Markt betreffen werden durch drei Klassen im Fachkonzept abgebildet.

Die Klasse *EventManager* verwaltet die gesamten Events. Dazu hat die Klasse folgenden Attribute:

- *groupEvents*: Diese *ArrayList* vom Typ *Event* speichert die Events, welche alle Spieler und den Markt betreffen (z.B. Wirtschaftskrise).
- *singleEvents*: Diese *ArrayList* vom Typ *Event* speichert die Events, welche einzelne Spieler betreffen. (z.B. Fabrik ist abgebrannt)
- *triggeredGroupEvents*: Diese *ArrayList* vom Typ *EventTrigger* speichert die ausgelösten Gruppen- und Marktevents.
- *triggeredSingleEvents*: Diese *ArrayList* vom Typ *EventTrigger* speichert die ausgelösten Events die einzelne Unternehmen betreffen.
- *groupChance*: Diese Integer Variable enthält die Wahrscheinlichkeit das ein Gruppenevent eintritt.
- *singleChance*: Diese Integer Variable enthält die Wahrscheinlichkeit das ein Einzelevent eintritt.

Der Typ *Event* besteht aus einem *Text*, einem *Type* und einem *Value*. Der Inhalt dieser Attribute lässt sich an folgendem Beispiel erkennen:

Ein Unternehmen muss Steuerabgaben zahlen, dies entspricht dem *Text*. Da ein Geldabfluss vorliegt ist der *Type* *cost*. Und schließlich noch die Höhe der Abgabe, welche als Integer in *value* gespeichert wird.

Der Typ *EventTrigger* ist eine Erweiterung der Klasse *Event* und enthält noch einen *counter*. Dieser hat den Zweck Events abzubilden, die erst in späteren Runden ausgelöst werden. Ein Beispiel für ein solches Event wäre ein Event, dass die Kaufkraft, nach einer Wirtschaftskrise, wieder auf den vorherigen Wert zurücksetzt. Zudem enthält die Klasse noch entsprechende *simulate* Methoden, welche den jeweiligen *Type* des Events analysieren und darauf aufbauend die entsprechenden Veränderung in den betreffenden Klassen der Company bzw. Marktes vornehmen.

Die oben genannten Attribute *groupEvents*, *singleEvents*, *groupChance* und *singleChance* werden in der Methode *createConfig()* mit vorgegebenen Daten gefüllt. Dazu werden die entsprechenden Werte aus der Config-Datei ausgelesen und in die Attribute geschrieben.

Die Simulation der Events findet in der Methode *simEvents()* statt. Diese Methode wird vom *GameManager* ab Runde 3 aufgerufen. Diese Methode überprüft ob eine Zufallszahl kleiner gleich der *groupChance* ist. Trifft dies zu, wird ein Zufälliges Event aus dem Attribut *groupEvents* ausgewählt und dem Attribut *triggeredGroupEvents* hinzugefügt. Über diese ArrayList läuft im nächsten Schritt eine Schleife, in welcher die entsprechende *simulate* Methode der Klasse *EventTrigger* aufgerufen wird.

Das gleiche geschieht danach noch für die Einzelevents mit den entsprechenden Variablen *singleChance*, *singleEvents* und *triggeredSingleEvents*.

## 3 Anforderungsspezifikation

### 3.1 Zielbestimmung

Mit diesem Unternehmensplanspiel soll ermöglicht werden, dass ein Spieler ein Unternehmen steuert, dass Smartphones produziert und diese Endkunden verkauft.

#### 3.1.1 Muss-Kriterien

- Der Spieler kann die Abteilungen Einkauf, Produktion, Verkauf, Forschung, Marketing und Rechtsabteilung steuern und verwalten.
- Die Spieler spielen gegen andere menschliche Spieler.
- Die Nachfrage nach Smartphones soll nach realistischen Werten simuliert werden.
- Das Spiel läuft rundenbasiert ab. Eine Runde entspricht einer Periode.
- Spieler können über ein Client-Server Modell gemeinsam spielen.
- Abteilungen und Produkte können durch ein "Level up"verbessert werden.
- Es gibt einen zentralen Markt auf dem die Smartphones abgesetzt werden können.
- Zufallsereignisse sollen den Spielverlauf für einzelne Unternehmen und für den ganzen Markt beeinflussen.

#### 3.1.2 Kann-Kriterien

- Die Spieler können über einen Chat miteinander kommunizieren.
- Downgrade von Abteilungen um Fixkosten zu senken.



### **3.1.3 Abgrenzungskriterien**

Aufgabe des Spieles ist es nicht sämtliche Aspekte eines Unternehmens zu simulieren. Es genügt wenn wesentliche Merkmale eines Smartphones-Herstellers abgebildet werden. Es wird unter anderem keinen Wert darauf gelegt Mitarbeiter zu simulieren und das Lager zu vergrößern. Eine Künstliche Intelligenz um Einzelspiele zu spielen ist nicht vorgesehen.

## **3.2 Einsatz**

Für dieses Unternehmensplanspiel gibt es keine konkrete Zielgruppe. Es soll aber von allen Interessenten verwendet werden können, die sich für ein wirtschaftliches Planspiel begeistern können.

Der Anwendungsbereich spielt keine Rolle. Da die Software regelmäßig Eingaben benötigt, ist ein unbeaufsichtigter Betrieb nicht vorgesehen.

## **3.3 Umgebung**

Die Software soll auf Windows-Rechnern laufen, auf denen eine aktuelle Java Version installiert ist. Die Hardware muss den Java Mindestanforderungen entsprechen und es muss eine Netzwerkkarte vorhanden sein. Um das Spiel zu spielen sollen mindestens zwei Spieler erforderlich sein.

## **3.4 Funktionalität**

In diesem Unternehmensplanspiel sind die folgenden Abläufe vorgesehen:

- Das Unternehmen kauft Rohstoffen ein.
- Das Produzieren von Smartphones.
- Das Verkaufen von Smartphones auf einem gemeinsamen Markt.
- Das Festlegen der Verkaufspreise für Smartphones.

- Das Upgraden von Abteilungen um die Kapazität bzw. Erfolgchance zu verbessern.
- Das Starten von unterschiedlichen Forschungskampagnen mit unterschiedlichen Auswirkungen.
- Das Ausspionieren anderer Spieler um bessere Smartphones zu produzieren.
- Das Starten von Marketingkampagnen um das Image zu verbessern.
- Das Überprüfen und Verklagen von anderen Spieler um sich gegen Spionagen zu wehren.
- Das Erstellen von Patenten um eine besser Position bei Rechtsstreiten zu haben.
- Das Anfechten eines Rechtsstreites.

### **3.5 Daten**

In diesem Planspiel müssen die Benutzereingaben nicht dauerhaft gespeichert werden. Die Startparameter für das Spiel sollen in einer Datei dauerhaft gespeichert werden.

### **3.6 Leistungen**

Es gibt keine konkreten Vorgaben bezüglich der Leistung. Der Nutzer soll aber eine positive Nutzererlebnis haben. Dazu soll das Spiel zügig auf Eingaben reagieren. Insgesamt sollten Wartezeiten unter 10 Sekunden liegen.

### **3.7 Benutzeroberfläche**

Die Benutzeroberfläche dieses Planspiels soll alle veränderbaren und nicht veränderbaren Parameter des Unternehmens darstellen. Zudem soll die Oberfläche

minimalistisch und übersichtlich sein. Ein ansprechendes Design ist kein entscheidendes Kriterium.

## **3.8 Qualitätsziele**

Das Unternehmensplanspiel soll am Ende fehlerfrei laufen und soll den hier genannten Anforderungen entsprechen. Der Code soll übersichtlich und kommentiert sein.

## **4 Vorgehen und Iterationen**

## **5 Zusammenfassung und Ausblick**

## **6 Planung und Vorgehen**

## 7 Einleitung

Es gibt zahlreiche Simulationen, die versuchen dem Spieler die Realität der Wirtschaft abzubilden. Avalon ist eine dieser Ideen, die dem Spieler den Smartphone-markt virtuell kreiert und ihm alle Möglichkeiten lässt, seine betriebswirtschaftlichen Ideen und Handlungen auszuführen. Nach dem Motto *grow fast* muss der Spieler sich versuchen auf dem aggressiven und schnelllebigen Markt zu etablieren und die Produkte abzusetzen. Dabei sollte er stets die anfallenden Kosten unter Kontrolle halten. Sein Ziel ist es vor den anderen Teilnehmern einen festgelegten Betrag auf dem Konto stehen zu haben. Neben diesem Lerneffekt gab es noch einen weiteren, die Programmierer von Avalon konnten wichtige Konzepte und Vorgehensweisen in der Entwicklungsphase mitnehmen. In der folgende Arbeit wird auf die Analyse des Marktes, verwendete Techniken, die Umsetzung und Programmierung eingegangen. Ein Fokus liegt auf den Besonderheiten des Marktes, deren Abbildung im Spiel und letztendlich der Implementierung.

## 8 Markt

### 8.1 Definition

Die Bezeichnung eines Smartphone trifft diese Definition aus dem Gabler Wirtschaftslexikon:

*Mobiltelefon mit erweitertem Funktionsumfang. Dazu zählen neben der Telefonie und Short Message Service (SMS) üblicherweise Zusatzdienste wie Electronic Mail (E-Mail), World Wide Web (WWW), Terminkalender, Navigation sowie Aufnahme und Wiedergabe audiovisueller Inhalte. Auf Smartphones laufen gegenüber herkömmlichen Mobiltelefonen komplexere Betriebssysteme wie etwa Symbian OS, Blackberry OS oder das iPhone OS. Die hierdurch geschaffene Möglichkeit zur Installation weiterer Applikationen durch den Endnutzer verleiht Smartphones einen erweiterbaren und individualisierbaren Funktionsumfang.*

Nach der Abgrenzung des Begriffes, wird nun im Folgenden auf die Geschichte und Entwicklung des Markts eingegangen.

### 8.2 Geschichte und Entwicklung

Laut dieser Definition gibt es das eigentliche Smartphone schon deutlich länger als viele Leute vermuten. Bereits 1994 entwickelte IBM ein damals „Personal Communicator“ genanntes tragbares Gerät. Weiter Anbieter kamen auf den Markt dazu, sodass Anfang des 21. Jahrhunderts Windows Mobile, Symbian, Blackberry OS und Palm OS zu den bekannten Systemen zählten. Allerdings waren diese Geräte nur wenigen Teilen der Bevölkerung vorenthalten und wurden hauptsächlich im beruflichen Umfeld genutzt. Einen regelrechten Boom hat der Smartphonemarkt durch die Einführung des iPhone's im Jahr 2007 erfahren. Von diesem Zeitpunkt an kamen weitere Anbieter hinzu, unter anderem Samsung und HTC. Diese Hersteller setzen davor zwar auch schon auf vorhandene Systeme, doch Apple verstand es am besten die Technik und Vorteile an den Privatanwender zu bringen und dies



mit einer Einfachheit, dass auch Menschen mit wenig technischem Verständnis die Geräte bedienen konnten. Von diesen Ideen geprägt veränderten auch die anderen Hersteller ihr Produktportfolio und damit ihre Smartphones. Durch die Veränderungen wurde auf einmal ein viel größerer Markt möglich. Angefangen im Jahre 2007 mit 122 Millionen verkauften Einheiten, hat sich der Markt rasant entwickelt. In der folgenden Statistik erkennt man, wie viel Potenzial noch in dem Markt steckt:

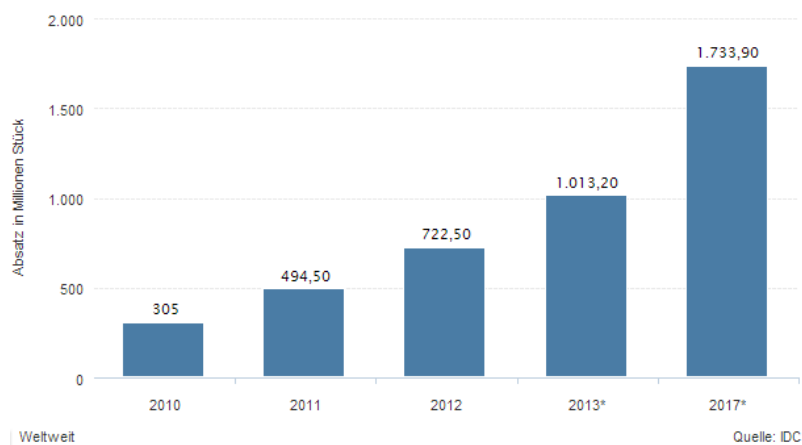


Abbildung 8.1: Statistik Absatzprognose

### 8.3 Aktuelle Situation

Der Smartphonemarkt ist ein besonderer Markt. Anders als bei den Computern, auf dem Windows mit fast 90 Prozent stark dominiert, mischen auf dem Smartphonemarkt verschiedene Betriebssysteme mit. Auf die drei größten wird kurz eingegangen:

**Android:** wird von der Open Handset Alliance entwickelt, deren Gründer Google ist. Auf Grund keiner Gebühren findet man es auf sehr vielen Geräten wieder. Android basiert auf dem Linuxkernel und soll ein freies Betriebssystem sein. Die Hersteller der Handys können es nach Belieben modifizieren und eigene Oberflächen für die Nutzer bereitstellen.

**iOS:** ist das Betriebssystem von Apple. Im Gegensatz zu Android läuft es nur auf den Geräten von Apple, dazu zählt das iPhone, iPad und iTV. Es ist keine Modi-

fikation des Betriebssystems möglich und Applikationen können ausschließlich über den eigenen Appstore installiert werden.

**Windows Phone:** ist das ehemalige Windows Mobile und der Versuch von Microsoft auch auf Smartphonemarkt mitzuspielen. Es wurde erst 2010 vorgestellt und ist damit im Vergleich zu den anderen Systemen das jüngste. Die Hersteller können entgegen einer Lizenzgebühr das Betriebssystem auf ihren Geräten installieren.

Im letzten Quartal 2013 wurden folgende Prozentsätze von den verschiedenen Systemen in Deutschland abgesetzt. An der Spitze steht Android mit fast 75 Prozent, gefolgt von iOS mit ca. 17 Prozent und dahinter WindowsPhone mit ca. 6 Prozent. Viele Käufer nehmen eines der oben genannten Betriebssysteme als eine Grundvoraussetzung für ihre Kaufentscheidung an. Weitere wichtige Entscheidungen werden in der folgenden Statistik deutlich:

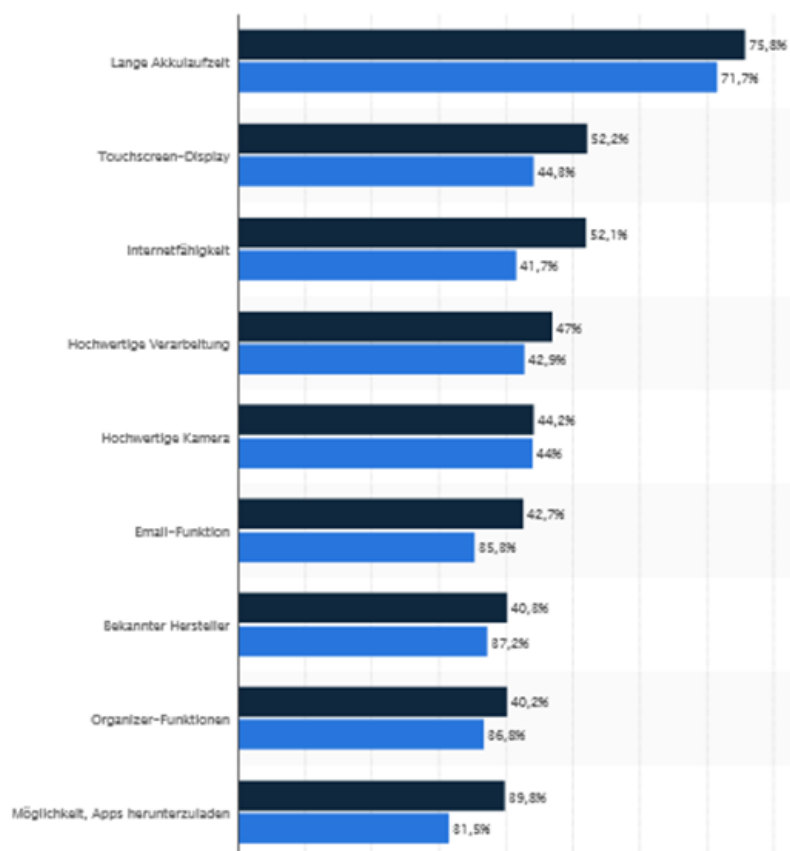


Abbildung 8.2: Kaufkriterien: 2012 (hellblau) | 2013 (dunkelblau)

Ein entscheidender Punkt für den Kauf eines Gerätes stellen technische Features da. Auf Platz eins steht in dieser Umfrage die Akkulaufzeit, d.h. wenn ein Hersteller in diesen Bereich stark investiert und eine wichtige Entdeckung machen würde, hätte der Kunde einen guten Grund für den Kauf seines Gerätes. Ähnlich verhält es sich mit einer Vielzahl von anderen Punkten auf der Statistik, wie die hochwertige Kamera, ein großes Display oder der Touchscreen. Aus dieser Darstellung lässt sich erkennen dass die Forschung in dem Smartphonemarkt eine sehr wichtige Rolle spielt. Ohne einen Touchscreen würde sich heute eine Vielzahl an Modellen überhaupt nicht mehr verkaufen. Ein gutes Beispiel hierfür ist der Hersteller Blackberry. Früher einer der Vorreiter im Enterprisebereich, hat heute sehr geringe Absatzzahlen und eine wirtschaftlich schwierige Lage. Einer der Gründe für die Situation war sicherlich, dass Blackberry zu lange an seinen klassischen Tastaturtelefonen festgehalten hat und nicht dem Trend nach neuen Entwicklungen gefolgt ist. Neben der Technik spielt allerdings für 40 Prozent der Befragten auch ein bekannter Hersteller eine wichtige Rolle, dieser Punkt wird in der folgenden Käufergruppenanalyse nochmal genauer dargestellt.

Auch im Smartphonemarkt gibt es unterschiedliche Preissegmente und somit auch Käufergruppen. Zu dem hochpreisigem Sortiment zählen die iPhones und Top-Modelle von Samsung, Sony, Nokia und HTC. Diese Modelle kosten bei ihrer Markteinführung meist über 500 Euro und deren Preis steigt mit zusätzlichen internen Speicher noch weiter. Typisch für die Modelle dieser Preisklasse sind technische Neuheiten, wie eine sehr hohe Displayauflösung oder eine besondereameratechnik. Die Nutzer sind bereit viel Geld auszugeben um auf dem neusten Stand der Technik zu sein. Außerdem finden sich in diesem Bereich zusätzlich Markenkäufer, d.h. Konsumenten die einem Hersteller treu bleiben. Typisch ist Markentreue für die Applegerätenutzern.

Nach diesem hochpreisigen Bereich, gibt es auch ein mittelpreisiges Segment. Die Modelle kosten zwischen 200 und 500 Euro und sind damit für eine größere Masse erschwinglich. Durch den niedrigeren Preis ist allerdings auch weniger Technik verbaut. Im Allgemeinen sind die Prozessoren nicht mehr so leistungsstark, das Display oder dessen Auflösung kleiner und keine absoluten neuen Features in das Gerät integriert. Die Nutzer dieser Smartphones, entsprechen dem Durchschnitt und wollen einige technische Möglichkeiten, jedoch nicht alle haben. Man könnte sie als Preis-Leistungs-Käufer bezeichnen.

Die letzte Käuferschicht, entspricht dem billig Segment. Die Smartphones kosten unter 200 Euro und sinken mit Preis bis zu hohen zweistelligen Beträgen. Die Hersteller, die in diesem Segment ihre Produkte verkaufen, wollen vor allen Dingen einen hohen Marktanteil in der Bevölkerung erreichen und auch weniger technisch versierte Menschen damit ansprechen, sich ein Smartphone zu kaufen. Von den Features werden wenig eingebaut um den Preis maximal drücken zu können. Meist besitzen die Geräte eine schlechtere Kameras und kleinere Displays.

Die oben beschriebenen drei Segmente schwanken natürlich, es gibt keine feste Abgrenzungsmöglichkeit. So existieren auch Geräte die wenig Kosten aber trotzdem gut ausgestattet sind und aus Gründen der Marktdurchdringung verkauft werden.

### 8.3.1 Besonderheiten

Der letzte wichtige Punkt in dem Smartphonemarkt sind Patentstreitigkeiten. Patente sind gewerbliche Schutzrechte, die eine Firma auf technische Neuheiten anmelden kann. Da sich auf dem Smartphonemarkt in den letzten Jahren sehr viele neue Techniken entwickelt haben, wurde auch eine große Anzahl von Patenten angemeldet. Das hat zum Ziel das wichtige Entdeckungen nur in den eigenen Produkten verbaut werden dürfen und so einen Wettbewerbsvorteil vor dem Konkurrenten entsteht oder das Unternehmen zumindest Lizenzgebühren für die Verwendung ihrer Erfindung einnehmen kann. Durch den hohen Druck auf dem Markt, verwenden viele Firmen jedoch einfach Technologien von Konkurrenten für die sie keine Genehmigung besitzen. Einer der größten Fälle, war die Klage von Apple gegen Samsung. Die zeitweise sogar ein Verkaufsverbot für ein Tablet von Samsung zur Folge hatte. Apple forderte ursprünglich eine Milliarde Dollar Schadensersatz von Samsung(<http://www.heise.de/mac-and-i/meldung/Samsung-muss-Apple-weitere-290-Millionen-Dollar-Schadensersatz-zahlen-2052251.html>)

Im Folgenden wird nun vereinfacht der Ablauf eines solchen Patentstreites dargestellt. Falls der Verdacht einer Verletzung der eigenen Patente vorliegt, hat das patentbesitzende Unternehmen grundsätzlich folgende Rechte:

- Unterlassung des Verkaufs des geschützten Gegenstandes
- Zahlung von Schadensersatz

- Produkte des Konkurrenten aus dem Markt entfernen
- Erhalt genauer Zahlen über den Verkauf der geschützten Erzeugnisse

Der Schadensersatz kann aus drei möglichen Faktoren berechnet werden. Zum ersten dem entgangenen Gewinn des patentbesitzenden Unternehmens, zum zweiten die Herausgabe des Gewinns die das gegnerische Unternehmen durch die Verletzung der Patente erzeugt hat und dritten die Zahlung von Lizenzgebühren. Verhärtet sich der Verdacht, dass der Konkurrent keine Rechte an der eingesetzten Technologie hat, kann das patentbesitzende Unternehmen versuchen die oben genannten Rechte durchzusetzen. Im allgemeinen Fall wird zu erste eine Abmahnung verschickt mit einer geforderten Schadenssumme. Selten jedoch geht das gegnerische Unternehmen im Smartphonemarkt darauf ein, da es sich meist um große Konzerne handelt. Um sich nun auf die Klage vorzubereiten ist eine sorgfältige Planung und Analyse des Sachverhaltes nötig, denn das klagende Unternehmen trägt die Beweislast. D.h. das Gericht stellt keine Nachforschungen an sondern bewertet nur die ihm vorgelegten Erkenntnisse. Oft dauert der Prozess über mehre Verhandlungstage hin bis zu Jahren wie in dem Fall Apple vs. Samsung, der schon seit April 2011 läuft. Entweder verhängt das Gericht als Urteil ein Verkaufsverbot oder fordert eine Schadensersatzzahlung von dem verurteilten Unternehmen.

Da diese Patentstreitigkeiten sich, wie beschrieben, oft länger hinziehen können, muss ein Unternehmen auch ausreichend Geld besitzen um seine eigene Rechtsabteilung mit den Anwälten zu finanzieren. Verliert das Unternehmen den Rechtstreit trägt es im Normalfall auch die Gerichtskosten des Gewinners, zusätzlich zu dem Schadensersatz oder andere Ansprüche, die es als Strafe bezahlen muss. Ein Gerichtsverfahren kann deshalb sehr teuer werden. Aus diesem Grund heraus haben viele Smartphonehersteller Patente und Lizenzen von anderen Unternehmen erworben um gegen solche Streitigkeiten gewappnet zu sein.

## 9 Spielprinzip

Das im Folgenden dargestellte Spiel „Avalon“ soll rundenbasiert verlaufen. Darunter versteht man, dass nach jeder abgeschlossenen Runde eine Simulationsphase startet, die für die nächste Runde neue Ausgangsbedingungen erzeugt. Jeder Spieler legt in der aktuellen Runde seine Tätigkeiten für die nächste Periode fest und trifft Entscheidungen. Danach wartet das System bis jeder Spieler seine Runde abgeschlossen hat. Darauf folgt die Simulationsphase, die alle Tätigkeiten der Spieler erfasst und umsetzt. In der nächsten Runde beginnt das Ganze von vorne. Jeder Teilnehmer des Wirtschaftssimulationsspiels soll dabei von seinem eigenen Rechner aus spielen können und nicht wie bei dem „hot seat“ Verfahren, alle Mitspieler an einem Computer abwechselnd. Die Hauptaufgabe besteht darin die verschiedenen Abteilungen zu koordinieren. In dem Spiel werden die Produktion, der Verkauf, der Einkauf, das Marketing, die Forschung und die Rechtsabteilung abgebildet. Jede dieser Abteilungen hat fixe Kosten die immer berechnet werden und variable Kosten, die nur berechnet werden wenn etwas in der Abteilung gemacht wird. Der Spieler soll die Möglichkeit bekommen, strategische Schwerpunkte zu setzen, in dem er einzelne Abteilungen z.B. stärker finanziert als andere. Außerdem soll durch Zufallseignisse das Spiel spannend gehalten und eine Monotonie vermieden werden.

### 9.1 Abteilungen

In diesem Kapitel werden die Abteilungen näher erläutert.

**Produktion:** In dieser Abteilung kann das aktuelle Smartphone produziert werden. Die Herstellung älterer Modelle ist nicht möglich. Für die Produktion werden Ressourcen benötigt und zwar pro Smartphone immer eine. Ein Upgrade dieser Abteilung hat geringere Ausschussquoten und somit auch weniger unnötig gekaufte Ressourcen zur Folge, verursacht jedoch höhere Fixkosten. Für die Produktion eines Smartphones fallen variable Kosten an.

**Verkauf:** In der Verkaufsabteilung werden alle Produkte, die sich im Lager des Unternehmens befinden angezeigt. Ältere Modelle können solange verkauft werden, wie sie noch vorhanden sind, jedoch wie oben beschrieben nicht mehr produziert. Um ein Modell zum Verkauf auf dem Markt freizugeben, genügt es einen Preis für dieses festzulegen.

**Einkauf:** Im Einkauf steht die Ressourcenbeschaffung im Vordergrund. In diesem Spiel hat der Nutzer die Möglichkeit sich zwischen drei Lieferanten zu entscheiden. Jeder dieser Lieferanten hat unterschiedliche Preise, Qualität und Zuverlässigkeit. Eine erhöhte Qualität hat einen geringeren Ausschuss bei der Produktion zur Folge und die Zuverlässigkeit erhöht die Planbarkeit des Unternehmens, denn falls dieser Wert gering ist, wird eine Lieferverzögerung häufiger auftreten.

**Forschung:** Die Forschungsabteilung ist ein sehr wichtiger Bestandteil der Firma. Grundsätzlich kann der Spieler Forschungskampagnen für Display, Prozessor und Akku starten. Jede der Kampagnen hat neben ihrem Preis drei festgelegte Werte, die Erfolgswahrscheinlichkeit, die Dauer und die mögliche Levelsteigerung. Die Erfolgswahrscheinlichkeit gibt an, mit welchem Prozentsatz eine neue Entwicklung gemacht wird. In der Dauer ist festgelegt, wie viel Zeit vergeht bis zu dem Zeitpunkt wann die Resultate der Kampagne gegeben werden. Falls nun die Kampagne geglückt ist wird der Wert der Levelsteigerung auf das aktuelle Forschungslevel addiert. Um sich nun gegen kommende Klagen zu schützen, kann der Spieler sein aktuellen Level gegen einen gewissen Geldbetrag patentieren lassen. Die Erfolgswahrscheinlichkeit der Kampagnen steigt durch ein Upgrade der Abteilung an, dies hat jedoch höhere Fixkosten zur Folge. Falls der Spieler wenig Geld investieren will, gibt es zusätzlich noch die Möglichkeit einen Konkurrenten auszuspionieren. Durch eine erfolgreiche Spionage steigt das eigene Forschungslevel an. Allerdings kann die Spionage auch von dem Gegner entdeckt werden und rechtliche Konsequenzen darauf folgen. Von der Forschungsabteilung kann der Spieler nun ein neues Produkt in die Produktion geben und somit das alte Modell ersetzen.

**Marketing:** Ähnlich wie in der Forschungsabteilung kann der Spieler im Marketing auch Kampagnen starten. Es kann zwischen einer Printmedien-, einer Internet- und einer Fernsehkampagne ausgewählt werden. Neben verschiedenen Preisen unterscheiden sich die Marketingmaßnahmen in der Laufzeit, dem Imageupgrade und der Erfolgswahrscheinlichkeit. Das Imageupgrade erhöht bei

einer erfolgreichen Kampagne die Absatzchancen auf dem Markt. Wie bei der Forschungsabteilung auch, kann die Erfolgswahrscheinlichkeit durch ein Upgrade, welches höhere Fixkosten verursacht, erhöht werden.

**Rechtsabteilung:** In der Rechtsabteilung kann der Spieler seine Konkurrenten auf Spionageaktivitäten in seinem eigenen Unternehmen untersuchen. Wird eine Spionagekampagne gefunden kann er den Gegenspieler verklagen. Ein Upgrade der Abteilung bringt höhere Siegchancen im Rechtsstreit mit sich.



## 10 User Interface

Zu Beginn der User Interface Entwicklung wurden ein Brainstorming innerhalb des Teams veranstaltet. Nach einigen Überlegungen und Diskussionen, einigte man sich auf das Kachel Design, wie es auch in Windows 8 zu finden ist. Ziel war es grundlegende Informationen direkt anzuzeigen und weitere Einstellungen und Details nach einem Klick auf die Kachel.

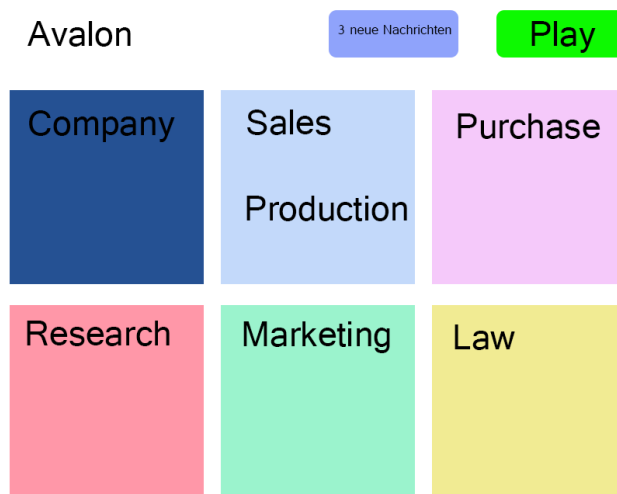


Abbildung 10.1: erstes Mockup

In der oberen Zeile, hat der Spieler die Möglichkeit seine Nachrichten zu lesen und durch das Drücken des Playbuttons die Runde zu beenden. In diesem Entwurf wurde davon ausgegangen, dass sich durch einen Klick auf die Panels ein Pop-up-Fenster öffnet, in dem der Nutzer seine Eingaben und Entscheidungen für die jeweilige Abteilung festlegt. Das zweite Mockup setzte jedoch schon auf eine andere Möglichkeit der Bedienung:

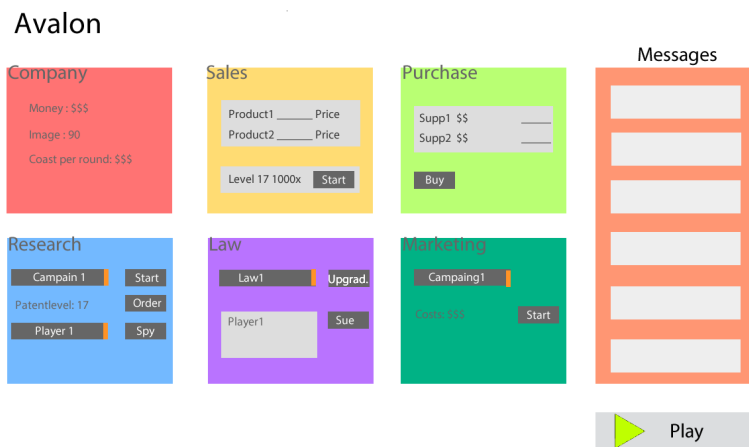


Abbildung 10.2: zweites Mockup

Das neuen Mockup ist darauf ausgelegt worden, dass sich für die Bedienung des Spiels keine weiteren Fenster mehr öffnen müssen. Die einzige Ausnahme ist das Lesen einer Nachricht. Ansonsten kann der Spieler alle Einstellungen direkt vornehmen. In dem Companypanel werden grundsätzliche Informationen wie das Geld und die Fixkosten angezeigt, die restlichen Abteilungen funktionieren wie im Spielprinzip beschrieben. Dieses Mockup hat sich zum letztendlichen Design nur noch wenig verändert. Hauptsächlich wurden Dropdown-Menüs durch Selectboxen ersetzt. Dies ist ein Screenshot des aktuellen Designs:

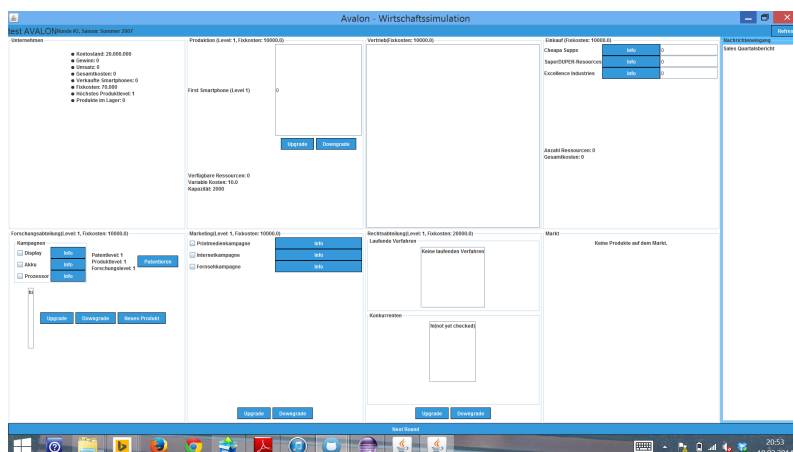
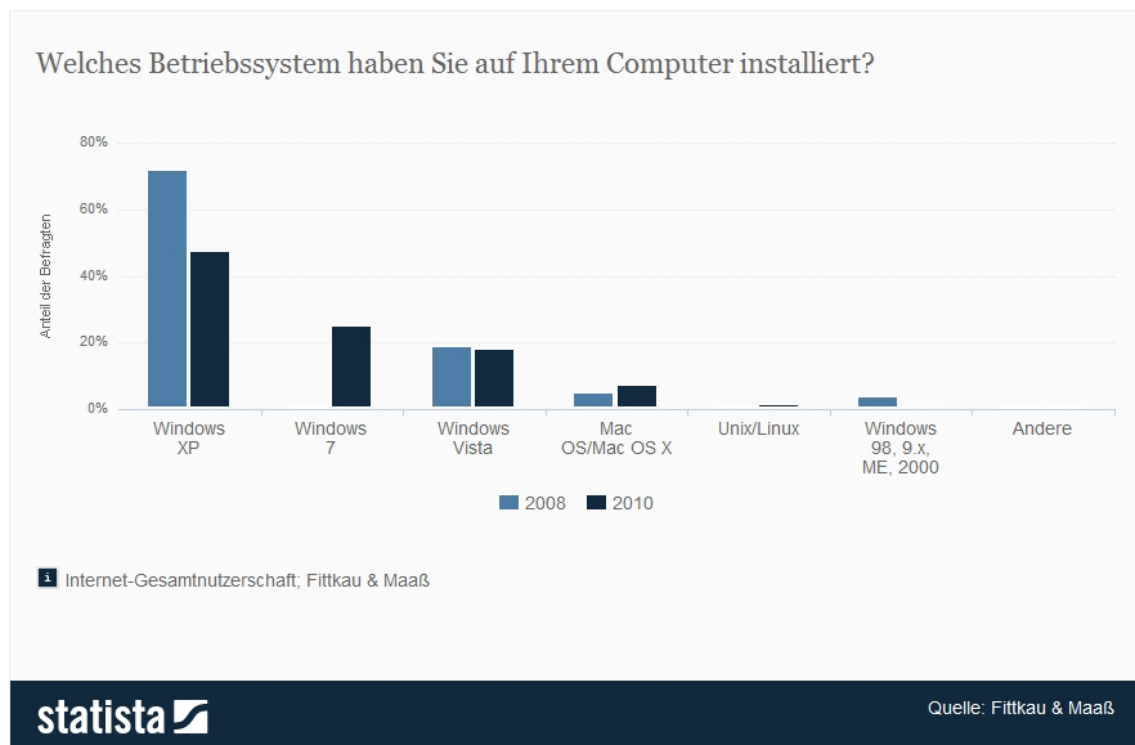


Abbildung 10.3: Screenshot

# **11 Planung und Vorgehen**

# Anhang

## Abschnitt A Umfrage:



# Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich:

1. dass ich meine Seminararbeit mit dem Thema *Dokumentation zum Unternehmensplanspiel "Avalon"* ohne fremde Hilfe angefertigt habe;
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe;
3. dass ich meine Seminararbeit bei keiner anderen Prüfung vorgelegt habe;
4. dass die eingereichte elektronische Fassung exakt mit der eingereichten schriftlichen Fassung übereinstimmt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Mannheim, 14. Februar 2014

Johannes Haaß