



Pflichtenheft

(Universität Paderborn, Softwareentwurf – WS 2004/2005)

Projekt: Entwicklung von „Mississippi-Queen“ für PDAs

Auftraggeber: Nils Bandener
Gruppe 5, Dienstags 9:00-11:00 Uhr, N3.206

Auftragnehmer: Universität Paderborn
Institut für Informatik
Bernhard Dietrich, btd@upb.de 6256800
Lars Fernhomberg, lafe@upb.de, 6256030
Sebastian Kniesburgs, seb.k.@gmx.de, 6257120
Marcus Köthenbürger, captain@upb.de, 6258550
Sebastian Huschenbeth, sebus@upb.de, 6230411
(Studentische Arbeitsgruppe)

Version	Datum	Autor(en)
1.0	02.11.2004	Bernhard Dietrich, 6256800 Lars Fernhomberg, 6256030 Sebastian Kniesburgs, 6257120 Marcus Köthenbürger, 6258550 Sebastian Huschenbeth, 6230411





1. Zielbestimmung

Der Trend geht immer mehr in die Richtung, Brettspiele für den PC oder auch Konsolen zu implementieren, wie zum Beispiel Monopoly. Das Spiel Mississippi Queen ist aufgrund seiner Erfolge (Spiel des Jahres, Verkaufszahlen) hervorragend für eine Digitale Umsetzung geeignet.

Die Nachteile von Brettspielen greifen bei diesem Spiel noch stärker als beispielsweise bei Monopoly. Die Nachteile wären:

- Die Unhandlichkeit (Große Spielfeldteile)
- Abnutzungserscheinungen (ständiges Anlegen von Teilen)
- Verlust von Spielfiguren (kleine Teile)

Die Vorteile der PDA-Version wären dementsprechend das genaue Gegenteil der Brettspiel-Nachteile und somit Handlichkeit, Robustheit und kein Risiko, Spielfiguren zu verlieren.

Die daraus resultierende Zielgruppe sind Studenten und junge Büro-Angestellte, sowie auch junge Manager. Dies ist auch die Zielgruppe für PDAs im Allgemeinen. Obwohl das Spiel mehr ein Familienspiel ist, ist der PDA für gemeinsame Spiele-Abende ungeeignet und wendet sich eher an die Pausen-Unterhaltung in Arbeitspausen.

2. Produkteinsatz

Dieser Abschnitt hat die Aufgabe den Einsatzbereich des zu entwickelnden Systems klarzustellen. Dazu gehören Erläuterungen der notwendigen Fachbegriffe und deren Zusammenhänge ebenso wie die Darstellung der systemrelevanten Abläufe im Einsatzbereich.

Unter dem Produkteinsatz versteht man sowohl den direkten Problembereich, wo das zu entwickelnde System eingesetzt werden soll, als auch die umgebenden Geschäftsprozesse.

Hier also den Problembereich des Projektes benennen und erläutern, ob es zu unterstützende Abläufe im Einsatzbereich (Geschäftsprozesse) gibt und wo sie zu finden sind.

2.1. Beschreibung des Problembereichs

Aufgabe dieses Abschnittes ist es, den Laien mit der Terminologie und den Zusammenhängen im Problembereich vertraut zu machen. Daher muss die Beschreibung möglichst allgemein sein.

Außerdem sollte der Text gut strukturiert sein. Auch der Einsatz von erläuternden Graphiken ist manchmal sinnvoll.





Abbildung 2.1: Graphik zur Illustration des Problembereichs

Wichtig ist es auch noch, gemachte Annahmen sauber von den oben beschriebenen Fakten getrennt aufzulisten. Dies erleichtert eine spätere Fehlersuche, wenn das System nicht die Erwartungen erfüllt.

2.2. Glossar

Kohlerad	Das Kohlerad ist der aktuelle Kohlebestand. Er wird vermindert bei speziellen Bewegungen. Bei zwei oder mehr Drehungen und/oder bei Geschwindigkeitsverminderung oder -erhöhung um mehr als einen Punkt werden Kohlepunkte abgezogen. Der Kohlebestand endet bei 0 und darf natürlich nicht negativ werden.
Geschwindigkeitsrad	Das Geschwindigkeitsrad gibt an, wie viele Felder die Spielfigur pro Spielzug bewegt werden darf. Die Geschwindigkeit kann pro Runde um eins erhöht oder um eins vermindert werden. Jede weitere Änderung der Geschwindigkeit kostet Kohlepunkte.
Richtungswürfel	Der Richtungswürfel wird geworfen, wenn ein Dampfer auf ein, bis dahin noch leeres, Flussteil kommt. Der Pfeil des Richtungswürfels gibt an, an welcher Verbindungsnase das nächste Flussteil angelegt wird.
Flussteile/Verbindungsnase	Das Spielfeld setzt sich im Laufe des Spiels zusammen. Die einzelnen Spielfeldteile heißen Flussteile. Sie werden an so genannten Verbindungsnasen angelegt. Das Prinzip der Verbindungsnase ist Puzzle-Ähnlich.
Landungsmole	Als Landungsmole wird das Endstück des Spielfelds bezeichnet. Es wird als letztes angelegt und die Dampfer müssen mit Geschwindigkeit 1 einlaufen. Der Dampfer, der die Landungsmole als erster erreicht (mit Geschwindigkeit 1), gewinnt das Spiel.



Abbildung 2.4: Illustration von <Prozess-ID> durch Aktivitätendiagramm

3. Produktfunktionen

Dieser Abschnitt hat die Aufgabe, die Funktionalität des zu entwickelnden Systems sowohl überblicksartig als auch detaillierter zu beschreiben. In diesem Abschnitt werden die vom Produkt erwarteten Funktionalitäten beschrieben. Jede dieser Funktionalitäten lässt sich einem elementaren Geschäftsprozess zuordnen, der im vorigen Abschnitt beschrieben wurde.

3.1. Use Case Diagramm

Aufgabe dieses Abschnittes ist es, einen Überblick über die Produktfunktionen zu geben. Dazu wird ein Use Case Diagramm eingesetzt, das eine abstrakte Sicht auf die Produktfunktionen und die externen Beteiligten an diesen Funktionen gibt.

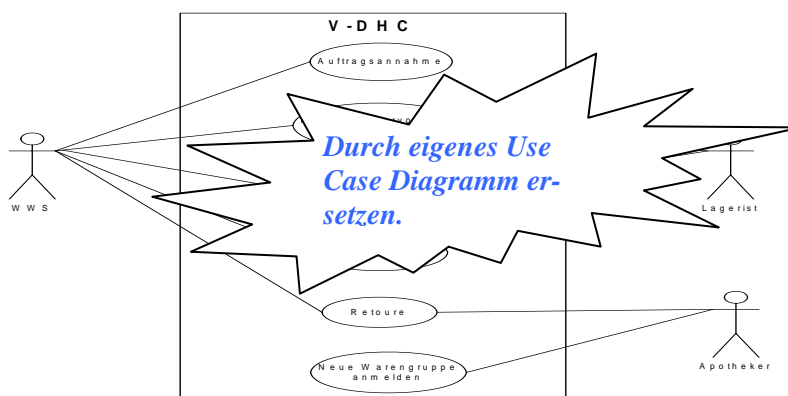


Abbildung 2.3: Use Case Diagramm mit den Produktfunktionen

3.2. Beschreibung zu Use Case-ID : Use Case-Name

Dieser Abschnitt muss als Template für jeden Use Case aus dem vorigen Abschnitt wiederholt werden. Es ist sein Aufgabe die Beschreibung der Funktionalität mit Details anzureichern. Einige der Punkte sind optional (z.B. GUI-Abbildungen) und müssen nur angegeben werden, wenn sie zutreffen.

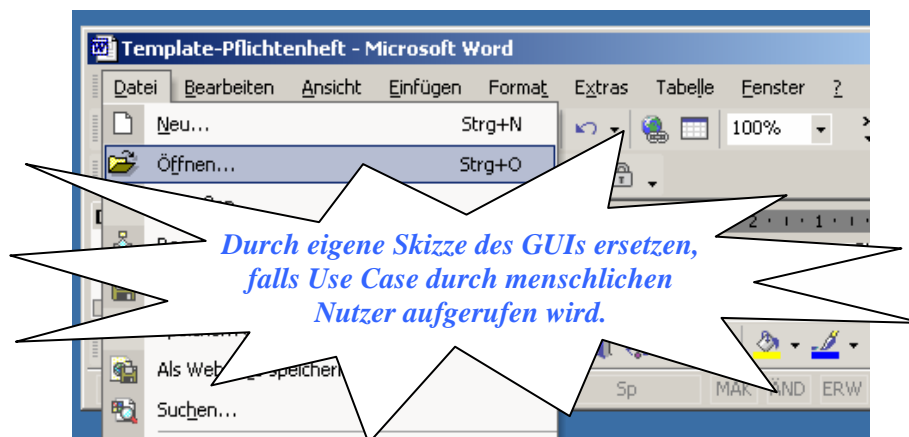
Charakterisierende Informationen

Aufgabe dieses Abschnittes ist die Erfassung der Hintergründe der Existenz des Use Cases.

Übergeordneter elementarer Geschäftsprozess:	<i>Prozess-ID: <elementarer Geschäftsprozess (verweist auf Abschnitt 2.5)></i>
Ziel des Use Cases:	<i>Ausführliche Beschreibung des Zieles des Use Cases</i>
Umgebende Systemgrenze:	<i>System, das betrachtet wird (Systemgrenze im Diagramm des vorigen Abschnittes)</i>
Vorbedingung:	<i>Was muss garantiert werden, damit der Use Case durchgeführt werden kann?</i>
Nachbedingung bei erfolgreicher Ausführung:	<i>Was muss sichergestellt werden für eine erfolgreiche Ausführung des Use Case</i>
Beteiligte Nutzer:	<i>Rollenname: Beschreibung des Nutzers, der mit dem System interagiert. Nutzer können auch andere Systeme sein.</i>
Auslösendes Ereignis:	<i>Handlung oder Zeitpunkt, die Use Case auslöst bzw. zu dem er beginnt</i>

GUI für den Aufruf des Use Cases:

Für menschliche Benutzer kann eine Skizze/Prototyp der Benutzungsschnittstelle dem Use Case beigelegt werden. Hier soll (typischerweise) dargestellt werden, mit welchem Button oder aus welchem Menü diese Produktfunktion aufgerufen wird.



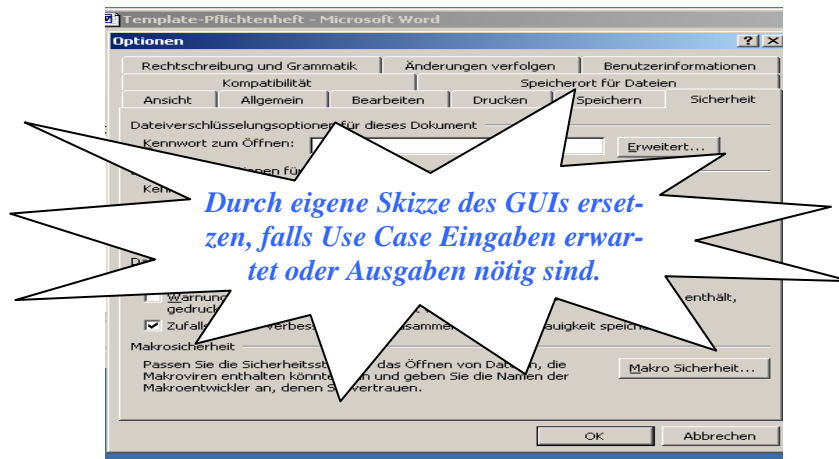
Szenario für den Standardablauf (Erfolg)

Dieser Abschnitt beschreibt die einzelnen Schritte, die vom auslösenden Ereignis bis zur erfolgreichen Beendigung des Use Cases aus der Sicht der beteiligten Nutzer notwendig sind.

Schritt	Nutzer	Beschreibung der Aktivität
<i>Schrittnr.</i>	<i>Name des beteiligten Nutzers</i>	<i>Beschreibung dessen, was der Nutzer tut</i>

GUIs für den Standardablauf des Use Cases:

Für Use Cases, die eine Interaktion mit einem menschlichen Benutzer erfordern, kann man eine Skizze/einen Prototypen der Benutzungsschnittstelle abbilden. Insbesondere sollten die Eingabemöglichkeiten des Benutzers detailliert dargestellt werden.



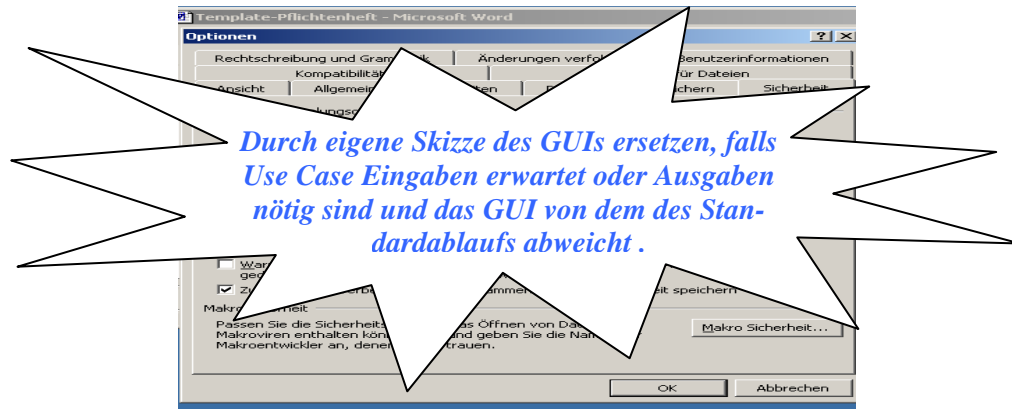
Eingabefeld	Erlaubte Eingabewerte
<Name eines Eingabefeldes aus dem GUI>	<Beschreibung, was ein erlaubter Eingabewert ist und was nicht>

Szenarien für alternative Abläufe (Misserfolg oder Umwege zum Erfolg)

Aufgabe dieses Abschnittes ist es, Fehlerfälle sowie Variationsmöglichkeiten im Ablauf des Use Cases zu beschreiben.

Schritt	Bedingung für Alternative	Beschreibung der Aktivität
Referenz auf Schritt nr. aus Standardablauf	Was verursacht den alternativen Ablauf?	Beschreibung der entsprechenden Aktivität bzw. Use Case-ID des Unter-Use Cases

GUIs für die alternativen Abläufe des Use Cases:



Eingabefeld	Erlaubte Eingabewerte
<i>Name eines Eingabefeldes aus dem GUI</i>	<i>Beschreibung, was ein erlaubter Eingabewert ist und was nicht</i>

Beschreibung des allgemeinen Ablaufes

In diesem Abschnitt werden die beiden vorherigen in einem Aktivitätendiagramm zusammengefasst. Außerdem wird die unvollständige Nutzersicht um interne Abläufe ergänzt, so dass eine vollständige Beschreibung entsteht. Existiert nur ein einziger Ablauf kann auf die Angabe des Aktivitätendiagramms verzichtet werden.



3.3. Offene Punkte

In diesem Abschnitt sollen alle Probleme und offenen Fragen gesammelt werden. Bei einem fertigen Pflichtenheft sollte er hoffentlich leer sein, aber bei Zwischenversionen kommt diesem Abschnitt besondere Bedeutung zu!

4. Produktcharakteristiken

In diesem Abschnitt werden Eigenschaften des zu entwickelnden Produktes beschrieben, die nicht direkt die zu leistende Funktionalität betreffen. Dies sind insbesondere die Systemumgebung in der das Produkt eingesetzt werden soll sowie die nicht-funktionalen Anforderungen. Je präziser diese Angaben sind, desto besser kann das realistische Verhalten des Produktes in Testumgebungen bestimmt werden.

4.1. Systemumgebung

Im Folgenden werden die Hard- und Softwaregegebenheiten spezifiziert, die das Spiel zum einwandfreien Betrieb benötigt.

4.1.1. Hardwareumgebung

PocketPC (NICHT Palm)

- mindestens 100 Mhz
- mindestens 64k Farbtiefe
- Optional: Audioausgang für Soundausgabe
- Optional: WLAN- oder Bluetooth-Verbindungen für Erweiterungen und/oder Netzwerkspiele

4.1.2. Softwareumgebung

Microsoft Windows Mobile 2003

Microsoft Windows CE

Microsoft .NET Compact Framework

4.2. Nicht-funktionale Anforderungen

Name:	Wenig Speicherbedarf
Typ:	Effizienz
Beschreibung:	Möglichst wenig Festplattenspeicher soll belegt werden, da auf PDAs traditionell wenig vorhanden ist.
Zugeordnete(r) Use Case(s)	<500 Kilobyte Speicherplatzbedarf

Name:	Schnelle Ladezeiten
Typ:	Effizienz
Beschreibung:	Da das Spiel kurzfristig in Pausen gespielt werden soll, muss es kurze Lade-/Speicherzeiten haben.
Zugeordnete(r) Use Case(s)	Ladezeiten des Spiels (komplett bis erster Spielzug getätigt werden kann) <10s

Name:	Ähnliches Aussehen wie Spielbrett
Typ:	Use
Beschreibung:	Um Nutzern des Brettspiels den Einstieg zu erleichtern, sollte die Programmoberfläche ähnlich der Brettspieloberfläche sein.
Zugeordnete(r) Use Case(s)	Möglichst detailgetreue Nachbildung der Spielele-



	mente
--	-------

Name:	Erweiterbarkeit (Multiplayer, Addons, Updates)
Typ:	Pflege
Beschreibung:	Das Produkt wird erweitert und/oder verbessert und der Benutzer kann sein Produkt aktualisieren.
Zugeordnete(r) Use Case(s)	Leichte Erweiterbarkeit via Internet

Name:	Cheat-Sicherheit
Typ:	Sicherheit
Beschreibung:	Um den Spielspaß auf Dauer zu gewährleisten, muss das Programm gegen Cheats gesichert sein.
Zugeordnete(r) Use Case(s)	Möglichst große Fehlerfreiheit

Typen von Produktcharakteristiken

Typ USE: Benutzbarkeitsanforderung

Die in Abschnitt 1 beschriebene Zielgruppe liegt diesen Anforderungen zugrunde. Wie muss die Software beschaffen sein, damit diese Zielgruppe gerne damit arbeitet?

Beispiel: Die Software soll flexibel für unterschiedliche Arbeitsweisen einsetzbar sein.

ODER

Die Software soll dem Erscheinungsbild anderer Produkte des Herstellers entsprechen.

Typ EFFIZIENZ: Effizienzanforderung

Hier geht es sowohl um Laufzeit- als auch um Speichereffizienz. Was wird unter dem sparsamen Einsatz dieser Ressourcen verstanden?

Beispiel: Die Berechnung darf nicht länger als 0,25 Sekunden dauern.

Typ PFLEGE: Wartbarkeits- und Portierbarkeitsanforderung

Welcher Grad an Änderbarkeit wird gefordert? Hier werden, soweit wie möglich, kommende Anpassungen und Erweiterungen vorhergesehen.

Beispiel: Das Produkt soll später auch in englischer Sprache verfügbar sein.

Typ SICHER: Sicherheitsanforderung

Zu den Sicherheitsanforderungen gehören die Aspekte Vertraulichkeit, Datenintegrität und Verfügbarkeit. Wie sehr müssen die Daten vor dem Zugriff durch Dritte geschützt werden? Ist es entscheidend, die Korrektheit der erfassten Daten und ihre Konsistenz zu gewährleisten? Dürfen Systemausfälle vorkommen?

Beispiel: Das System muss gewährleisten, dass Daten nie verändert werden können.

Typ LEGAL: Gesetzliche Anforderung

Welche Standards und Gesetze müssen beachtet werden?

Beispiel: Das Produkt muss die ISO 9000 Norm erfüllen.

