



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 Kniesburges, 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 Kniesburges, 6257120 Marcus Köthenbürger, 6258550 Sebastian Huschenbeth, 6230411
2.0	28.11.2004	Bernhard Dietrich, 6256800 Lars Fernhomberg, 6256030 Sebastian Kniesburges, 6257120 Marcus Köthenbürger, 6258550 Sebastian Huschenbeth, 6230411
3.0	22.12.2004	Bernhard Dietrich, 6256800 Lars Fernhomberg, 6256030 Sebastian Kniesburges, 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

2.1. Beschreibung des Problembereichs

Der Problembereich des Spieles „Mississippi Queen“ wird in der Spielanleitung des Brettspiels ausführlich erläutert. Zu beachten ist, dass in der geplanten Umsetzung für PDAs auf das in der Anleitung als fortgeschrittene Spielversion erläuterte Spielelement der „Passagierbeförderung“ verzichtet wird.

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





Flussteile/Verbindungsnase	Dampfer auf ein, bis dahin noch leeres, Flusssteil kommt. Der Pfeil des Richtungswürfels gibt an, an welcher Verbindungsnase das nächste Flusssteil angelegt wird.
	Das Spielfeld setzt sich im Laufe des Spiels zusammen. Die einzelnen Spielfeldteile heißen Flusssteile. Sie werden an so genannten Verbindungsnasen angelegt. Das Prinzip der Verbindungsnase ist Puzzel-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.

2.3. Modell des Problembereichs

Im folgenden findet sich ein Modell des Problembereichs, welches mit Hilfe eines UML Klassendiagramms dargestellt wird. Im Anschluss finden Sie erläuternde Worte zu den einzelnen Designentscheidungen.

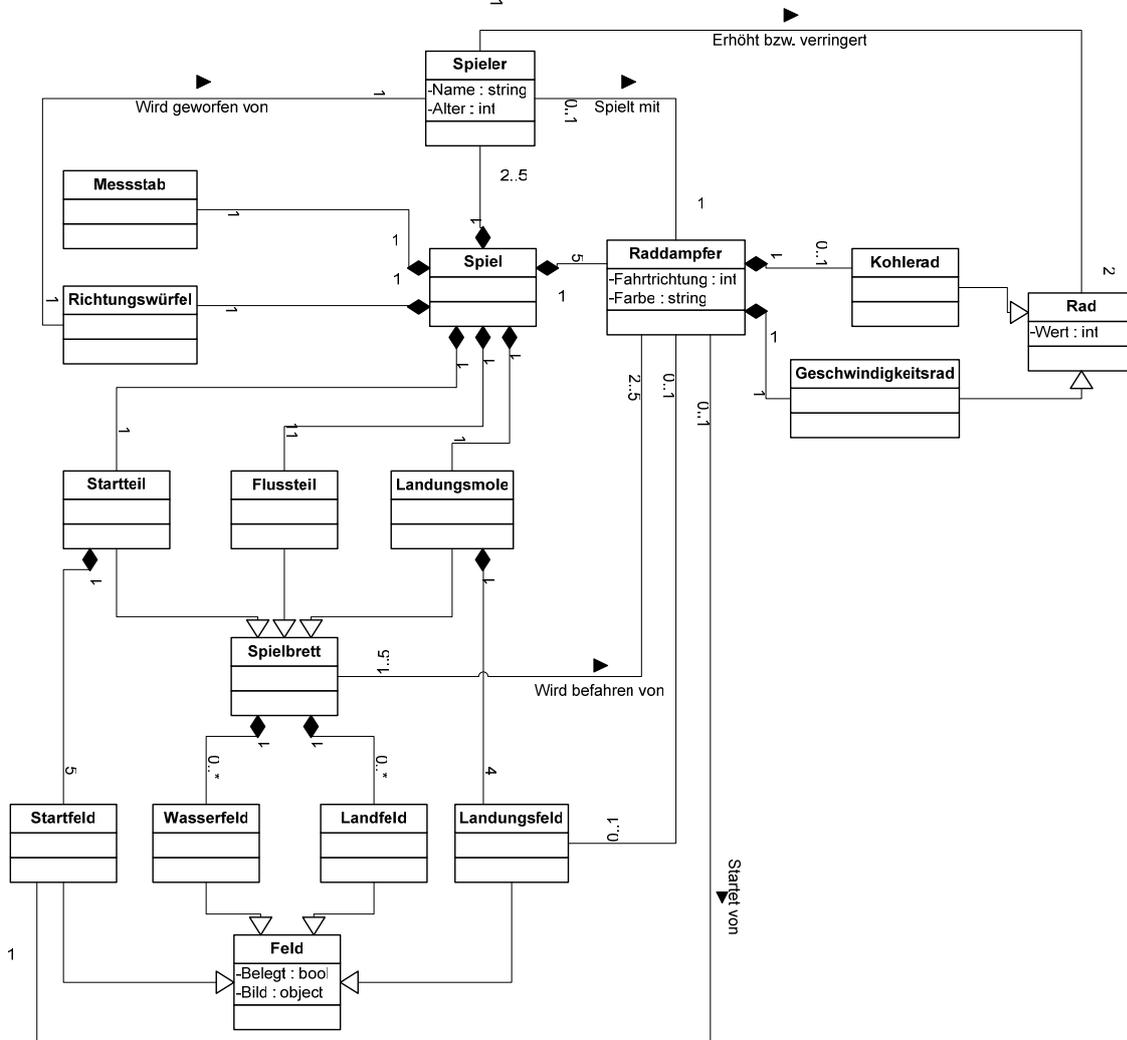


Abbildung 1 Klassendiagramm für den Problembereich

Zentraler Dreh- und Angelpunkt des UML Diagramms ist die Klasse „Spiel“, welche genau einen Messstab, genau einen Richtungswürfel, genau fünf Raddampfer sowie genau ein Startteil, genau elf „normale“ Flussteile sowie genau eine Landungsmole besitzt. Des weiteren nehmen zwei bis fünf Spieler teil, so dass das Spiel ebenfalls zwei bis fünf Spieler „besitzt“. Die einzelnen Kardinalitäten erheben sich aus der Spielanleitung und dem Packungsinhalt des originalen Brettspiels.

Der Spieler besitzt zur Identifizierung einen „Name[n]“ (vom Typ String) sowie ein „Alter“ (vom Typ Integer), welches zur Bestimmung der Startreihenfolge benötigt wird, da der jüngste Spieler beginnt. Jeder Spieler hat genau einen Raddampfer als Spielfigur, wobei allerdings nicht jeder Raddampfer zwangsläufig einen Spieler besitzt, da sich das Spiel auch mit weniger Teilnehmern als Raddampfern spielen lässt und somit einige Raddampfer nicht am Spielverlauf teilnehmen. Es ergibt sich somit eine 0..1 Kardinalität.



Der Raddampfer besitzt die Attribute „Fahrtrichtung“ vom Typ Integer und „Farbe“ vom Typ String. Die Fahrtrichtung gibt den Winkel an, mit dem der Raddampfer auf einem Feld steht.

Ein Raddampfer hat ein bzw. je nach Spielverlauf auch kein Kohlerad sowie genau ein Geschwindigkeitsrad, so dass sich die Kardinalitäten automatisch ergeben. Beide Räder erben von der Klasse „Rad“, welches das Attribut „Wert“ hat, in welchem der aktuelle Stand des Rades gespeichert wird. Da der Spieler durch seine Spielweise die Werte des jeweiligen Rades verringert bzw. erhöht liegt zwischen dem Spieler und dem Rad eine 1 zu 2 Kardinalität vor (**@Seb: Richtig so????**).

Die Raddampfer fahren auf Spielbrettern, so dass die Spielbretter von den Raddampfern befahren werden. Es gibt 2 bis fünf Raddampfer (je nach Teilnehmeranzahl), die auf mindestens einem maximal aber fünf Spielbrettern fahren. Wenn alle Dampfer auf dem gleichen Spielbrett sind, ergibt sich die untere Grenze der Kardinalität, wenn alle Dampfer auf unterschiedlichen Brettern sind und fünf Teilnehmer spielen, ergibt sich die obere Grenze der Kardinalität.

Ein Raddampfer startet von genau einem der fünf Startfelder, welche auf dem Startteil vorhanden sind, während nicht zwangsläufig jedes Startfeld ein Raddampfer beherbergen muss, da es vorkommen kann, dass weniger als fünf Leute mitspielen. Da es nur vier Landungsmolen gibt (Plätze 1-4; der Letzte ist Nr. 5) kann ein Raddampfer entweder auf einem Feld landen oder nicht. Da es ferner aber auch sein kann, dass es weniger als vier Mitspieler und somit Raddampfer gibt, muss nicht zwangsläufig auf jedem Landungsfeld ein Raddampfer landen, so dass sich die entsprechenden Kardinalitäten automatisch ergeben.

Das Spiel besteht, wie bereits oben erwähnt, aus genau einem Startteil, genau elf „normalen“ Flussteilen und einer Landungsmole. Da diese Spielelemente sehr ähnlich strukturiert sind, erben diese von der Klasse „Spielbrett“. Jedes Spielbrett hat 0 bis „unendlich“ Wasser- und Landfelder, während jedes Wasser- und jedes Landfeld genau ein zugehöriges Spielbrett hat.

Zusätzlich gibt es noch die besonderen Spielbretter „Startteil“ und „Landungsmole“, welche genau fünf Startfelder bzw. genau vier Landungsfelder haben.

Start-, Wasser, Land- und Landungsfelder erben ferner von der Klasse „Feld“, da die einzelnen Feldtypen grundlegende Ähnlichkeiten, z.B. ein Attribut welches verrät, ob das Feld belegt ist, bzw. ein Attribut für das anzuzeigende Bild, haben.

Der Richtungswürfel wird bei jedem Wurf von genau einem Spieler geworfen, während der Spieler bei jedem Wurf genau einen Richtungswürfel wirft, so dass die Kardinalität offensichtlich ist.

2.4. Beschreibung der Geschäftsprozesse

Dieser Abschnitt ergänzt den vorigen und entfällt somit auch, wenn der vorige entfallen ist. Seine Aufgabe ist es, die dort identifizierten Geschäftsprozesse näher zu beschreiben. Jeder von ihnen erhält einen eigenen Unterabschnitt gemäß dem Template. In diesem Abschnitt wird der Ablauf der Geschäftsprozesse des vorigen Abschnittes genauer beschrieben. Diese Abläufe sind es, die das zu entwickelnde System ausschnittsweise unterstützen soll.

Beschreibung zu <Prozess-ID>: <Name des Geschäftsprozesses>

Auslösendes Ereignis	<i><Handlung oder Zeitpunkt, die Geschäftspro-</i>
-----------------------------	---



nis:	<i>zess auslöst bzw. zu dem er beginnt</i>
Ergebnis:	<i><Was im Falle einer erfolgreichen Ausführung des Geschäftsprozesses erreicht werden soll></i>
Mitwirkende:	<i><Rollenname derjenigen, die an der Durchführung des Geschäftsprozesses beteiligt sind. Das können auch existierende Systeme sein.></i>

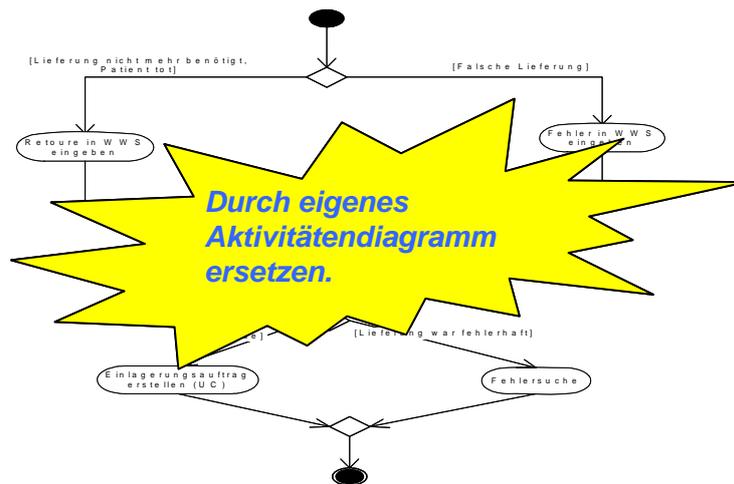


Abbildung2: Illustration von <Prozess-ID> durch Aktivitätendiagramm

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

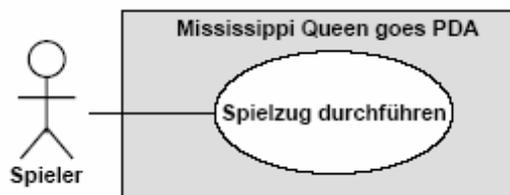


Abbildung 2 Use Case Diagramm (Ausschnitt)

3.2. Beschreibung zu UC1: *Spielzug durchführen*

3.2.1. Charakterisierende Informationen

Name:	Spielzug durchführen
--------------	----------------------

Primärer Nutzer:	Aktiver Spieler
Ziel (des Nutzers):	Der Spieler möchte mit seinem Spielzug möglichst effektiv auf dem aktuellen Spielbrett navigieren, so dass seine Ausgangsposition für den nächsten Schritt möglichst gut und seine Gewinnaussicht für das gesamte Spiel ebenfalls gut ist.
Vorbedingung:	Das Spielbrett ist ordnungsgemäß „initialisiert“.
Nachbedingung bei erfolgreicher Ausführung:	Spielzug ist innerhalb der Spielregeln durchgeführt worden.
Auslösendes Ereignis:	Ermittlung des nächsten Spielers hat ergeben, dass der entsprechende Spieler an der Reihe ist.
Umgebende Systemgrenze:	???
Beteiligte Nutzer:	Aktiver Spieler, Raddampfer, Spielbretter ???

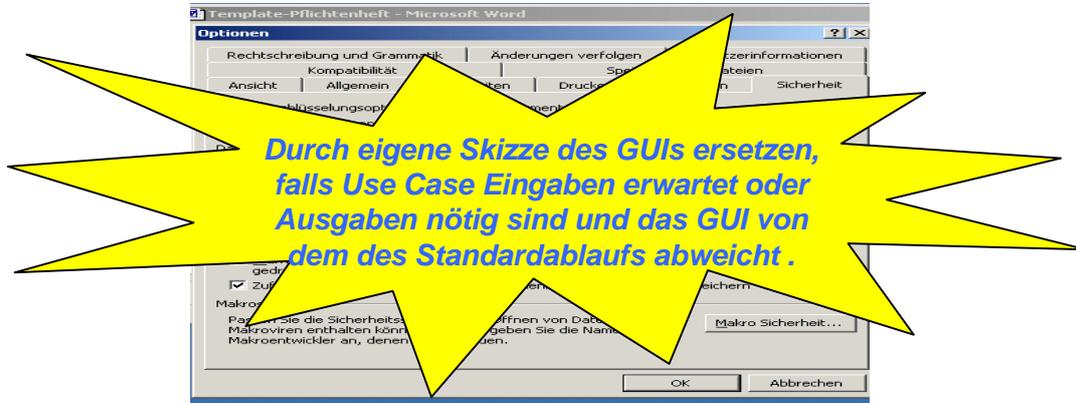
3.2.2. Hauptszenario (Erfolg)

Schritt	Nutzer	Beschreibung der Aktivität
1	Aktiver Spieler	Geschwindigkeitsrad des Kohledampfers um eins erhöhen
2	Aktiver Spieler	In gerader Richtung entsprechend der Geschwindigkeit des Geschwindigkeitsrades ziehen

3.2.3. Alternativszenarien (Misserfolg oder Umwege zum Erfolg)

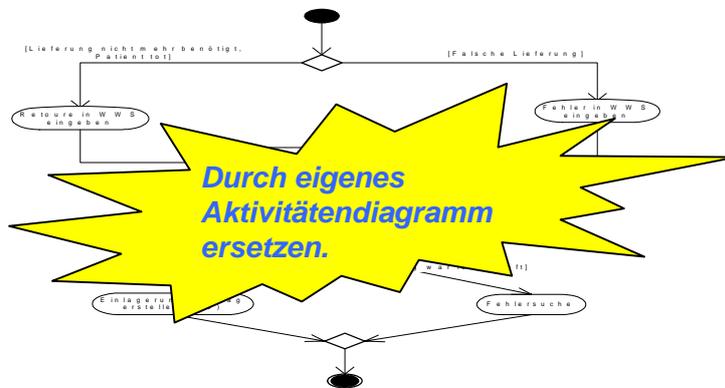
Schritt	Bedingung für Alternative	Beschreibung der Aktivität
1	Geschwindigkeit wird um zwei Punkte erhöht	Kohlerad um eins verringern, Wenn Kohlerad < 0, scheidet der Spieler aus
2	Drehung des Dampfens um 120° am Ende der Bewegung	

GUIs für die Alternativszenarien 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>

3.2.4. 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



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) Case(s)	Use <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) Case(s)	Use 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) Case(s)	Use Möglichst detailgetreue Nachbildung der Spielelemente

Name:	Erweiterbarkeit (Multiplayer, Addons, Updates)
Typ:	Pflege
Beschreibung:	Das Produkt wird erweitert und/oder verbessert und der Benutzer kann sein Produkt aktualisieren.





Zugeordnete(r) Case(s)	Use	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) Case(s)	Use	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: Effizienzforderung

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.



Aufgabe 4: Beschreiben Sie den Architekturstil „Pipes and Filters“.

Das Pipes und Filter Architekturmuster bietet eine Struktur für Systeme, die einen Datenstrom verarbeiten. Jeder verarbeitende Schritt ist in einer Filterkomponente abgebildet. Zwischen benachbarten Filtern werden Daten mittels Pipes transportiert. Filter lassen sich neu kombinieren um verwandte Anwendungen zu erzeugen.

Das Pipes und Filter Architekturmuster unterteilt die Aufgabe des Gesamtsystems in sequentielle verarbeitende Schritte. Diese Schritte werden durch einen Datenfluß durch das Gesamtsystem verbunden wobei die Ausgabedaten des einen Filters die Eingabedaten des nächsten sind.

Jeder verarbeitende Schritt wird implementiert durch eine Filterkomponente. Eine Filterkomponente verarbeitet seine Eingabedaten kontinuierlich und stellt Ausgabedaten sofort bereit.