

UP

Dokumentation nach dem Unified Process

Prof. Dr. Karl Friedrich Gebhardt

©1996 – 2014 Karl Friedrich Gebhardt

Auflage vom 3. Dezember 2014

Prof. Dr. K. F. Gebhardt
Duale Hochschule Baden-Württemberg Stuttgart
Angewandte Informatik

Tel: 0711-667345-11(16)(15)(12)
Fax: 0711-667345-10
email: kfg@dhbw-stuttgart.de

Inhaltsverzeichnis

1	Management	1
1.1	Technisches Management	1
1.1.1	Konzeption	1
1.1.2	Entwurf	1
1.1.3	Konstruktion	1
1.1.4	Auslieferung	1
1.2	Kundenmanagement	1
1.2.1	Auftraggeber	1
1.2.2	Anwender	1
1.3	Teammanagement	2
1.4	Organisationsmanagement	2
1.5	Besprechungsprotokolle	2
2	Architektur	2
2.1	Entwicklungsumgebung	2
2.2	Klassenbibliotheken	2
2.3	Komponenten	2
2.4	Frameworks	2
2.5	System/Subsystem-Struktur	2
2.6	Fehlerbehandlung	3
2.7	Persistenz	3
2.8	Testen	3
2.9	Verteilung von Objekten – Networking	3
2.10	Sicherheit	3
2.11	Benutzeroberfläche	3
2.12	Dokumentations	3
2.13	Systemstart und -ende	3
3	Anforderungsanalyse	3

4	Analyse	4
4.1	Brainstorming	4
4.2	Systembeschreibung	4
4.3	Anwendungsfälle	4
4.3.1	Anwendungsfall Name	4
4.4	Fachwissen	5
4.5	Analyse von Entitäten	5
4.5.1	Datenstruktur	5
4.5.2	Verhalten	5
4.6	Datenflüsse	5
4.7	Ideensammlung	5
5	Design	5
5.1	Substantivlisten	5
5.2	CRC-Karten	5
5.3	Klassendiagramme	6
5.4	Beschreibung der Klassen	6
5.5	Design-Patterns	6
5.6	Verhaltens-Diagramme	6
6	Implementierung	6
7	Testen	6
7.1	Anwendungsfälle	6
8	Systemintegration	6
8.1	Teilsysteme	6
8.2	Komponenten	7
8.3	Bibliotheken	7
9	Einsatz	7
9.1	Benutzerdokumentation	7
9.2	Auslieferung, Installation	7
9.3	Einweisung und Schulung	7
9.4	Produktionseinsatz, Operation des Systems	7

9.5	Wartung	7
9.6	Erweiterung	7
10	Literatur	7
11	Management	8
11.1	Technical Management	8
11.1.1	Inception	8
11.1.2	Elaboration	8
11.1.3	Construction	8
11.1.4	Transition	8
11.2	Client Management	8
11.2.1	Customer (Decisionmaker)	8
11.2.2	User	9
11.3	Team Management	9
11.4	Organization Management	9
11.5	Protocols	9
12	Architecture	9
12.1	Development Environment	9
12.2	Class Libraries	9
12.3	Components	9
12.4	Frameworks	9
12.5	System/Subsystem Structure	9
12.6	Error Handling	10
12.7	Persistence	10
12.8	Testing	10
12.9	Distribution of Objects – Networking	10
12.10	Security	10
12.11	User Interface	10
12.12	Systemstart and -end	10
13	Requirements-Analysis	10

14 Analysis	10
14.1 Brainstorming	11
14.2 System Description	11
14.3 Use Cases	11
14.4 Specialist Knowledge	11
14.4.1 Literature	11
14.5 Analysis of Entities	11
14.5.1 Datastructures	11
14.5.2 Behaviour	11
14.6 Dataflows	11
14.7 Collection of Ideas	11
15 Design	12
15.1 Lists of Nouns	12
15.2 CRC-Cards	12
15.3 Class Diagrams	12
15.4 Description of Classes	12
15.5 Design Patterns	12
15.6 Behavioural Diagrams	12
16 Implementation	12
17 Testing	12
17.1 Use Cases	13
18 System Integration	13
18.1 Subsystems	13
18.2 Components	13
18.3 Libraries	13
19 Deployment	13
19.1 User Documentation	13
19.2 Delivery, Installation	13
19.3 Introduction and Training	13
19.4 Production System, Operation of System	13

19.5 Maintenance	14
19.6 Extensions	14
20 Literature	14

1 Management

(Diese Prozesskomponente enthält eventuell alles, was zum typischen Projektmanagement (Vorgehensweisen, Zeitpläne usw) gehört.)

Arbeitstitel: **Projekttitel**

1.1 Technisches Management

(Wird eventuell ganz im Abschnitt "Architektur" abgehandelt.

Allerdings kann hier die Definition und Verwaltung der Prozessphasen untergebracht werden. Was wurde in den Phasen getan? Wie lange haben sie gedauert? Welche Probleme gab es? Die Darstellung eines Designs wäre hier fehl am Platz.)

1.1.1 Konzeption

1.1.2 Entwurf

1.1.3 Konstruktion

1.1.4 Auslieferung

1.2 Kundenmanagement

1.2.1 Auftraggeber

1.2.2 Anwender

1.3 Teammanagement

(Wer sind die Teammitglieder? Gibt es eine Hierarchie oder Aufgabenverteilung?)

1.4 Organisationsmanagement

(Wie ist das Projekt in das Unternehmen eingebettet? Gibt es Unternehmenspolitische Randbedingungen? Gibt es Projekte, die konkurrieren oder mit denen zusammengearbeitet werden kann?)

1.5 Besprechungsprotokolle

2 Architektur

2.1 Entwicklungsumgebung

(Welche Entwicklungsumgebung wird verwendet? Programmiersprachen? Kodierkonventionen? Editor? Hardware?)

2.2 Klassenbibliotheken

2.3 Komponenten

(Werden Komponenten gekauft? Werden Komponenten im Zuge des Projekts entwickelt?)

2.4 Frameworks

2.5 System/Subsystem-Struktur

2.6 Fehlerbehandlung

(Programmierfehler? Fehler des Benutzers? Systemabstürze?)

2.7 Persistenz

(Wird ein Datenbanksystem verwendet? Wenn ja, welches? Wann und wo wird auf das DBS zugegriffen? Behandlung von Transaktionen?)

2.8 Testen

(Wie wird getestet? Hier keine Testresultate.)

2.9 Verteilung von Objekten – Networking

(Resultiert das Projekt in einem verteilten System? Welche Software soll auf welchen Hosts laufen?)

2.10 Sicherheit

(Wie sollen eventuelle Sicherheitsprobleme behandelt werden?)

2.11 Benutzeroberfläche

2.12 Dokumentations

(Wo wird dokumentiert? Welche Dokumente werden erzeugt? Wo findet man was?)

2.13 Systemstart und -ende

(Wahrscheinlich gehört das in die Prozesskomponente "Einsatz". Aber eventuell gibt es allgemeinere, projektübergreifende Vorgehensweisen.)

3 Anforderungsanalyse

*(Spezifische Anforderungen des Kunden werden hier gelistet.
Eventuell werden eigene Anforderungen an das System getellt.)*

4 Analyse

*(Wie sieht die Welt des Kunden aus? Erarbeitung des notwendigen Fachwissens.
Was soll das System tun?)*

4.1 Brainstorming

(Wird ganz früh im Projekt durchgeführt.)

4.2 Systembeschreibung

(Textuelle Beschreibung: Wie sieht die Welt des Kunden aus? Wie soll das zu entwickelnde System aussehen?)

4.3 Anwendungsfälle

(Liste der dargestellten Anwendungsfälle. Erwähnung trivialer Anwendungsfälle, die nicht explizit dargestellt werden.)

4.3.1 Anwendungsfall Name

Name:

Kurzbeschreibung:

Ablaufbeschreibung:

1. x

2. x

Akteure:

- Primäre Akteure:
- Sekundäre Akteure:

Vorbedingungen:

Nachbedingungen:

Invarianten:

Regeln:

Nicht-funktionale Anforderungen:

Erweiterungspunkte:

Ausnahmen, Fehlersituationen:

Variationen:

Dienste:

Anmerkungen:

4.4 Fachwissen

(Erarbeitung und Darstellung des Fachwissens über die Welt des Kunden. Das sollte unbedingt vom Kunden geprüft werden.)

4.5 Analyse von Entitäten

(Eine Liste von Substantiven der Systembeschreibung, der Anwendungsfälle und sonstiger Texte wird iterativ bearbeitet, bis eine Liste von möglichen Klassen und eventuell auch Attributen resultiert. Eventuell CRCs.)

4.5.1 Datenstruktur

(Hier werden die Attribute der Klassen entwickelt.)

4.5.2 Verhalten

(Hier werden die Methoden der Klassen entwickelt. Welche Dienste soll eine Klasse zur Verfügung stellen?)

4.6 Datenflüsse

4.7 Ideensammlung

5 Design

5.1 Substantivlisten

5.2 CRC-Karten

5.3 Klassendiagramme

5.4 Beschreibung der Klassen

(Oft sind die Namen von Klassen so gewählt, dass nicht unmittelbar klar ist, was diese Klassen wozu und mit welchen anderen Klassen tun. Diese Informationen sollen hier gegeben werden. Es müssen daher nicht alle Klassen beschrieben werden. Ferner genügen nur die wichtigsten Methoden.)

5.5 Design-Patterns

(Wurden Design-Patterns verwendet? Wenn ja, wo?)

5.6 Verhaltens-Diagramme

6 Implementierung

(Interessante, kommentierte Code-Snipets wären hier ideal.)

7 Testen

7.1 Anwendungsfälle

8 Systemintegration

8.1 Teilsysteme

8.2 Komponenten

8.3 Bibliotheken

9 Einsatz

9.1 Benutzerdokumentation

9.2 Auslieferung, Installation

9.3 Einweisung und Schulung

9.4 Produktionseinsatz, Operation des Systems

9.5 Wartung

9.6 Erweiterung

10 Literatur

(Enthält insbesondere auch Literatur zum Fachwissen und zur Welt des Kunden.)

Doku Englisch, teilweise mit Erklärungen

11 Management

(Diese Prozesskomponente enthält eventuell alles, was zum typischen Projektmanagement (Vorgehensweisen, Zeitpläne usw) gehört.)

Working Title:

11.1 Technical Management

(Wird eventuell ganz im Abschnitt "Architektur" abgehandelt.

Allerdings kann hier die Definition und Verwaltung der Prozessphasen untergebracht werden. Was wurde in den Phasen getan? Wie lange haben sie gedauert? Welche Probleme gab es? Die Darstellung eines Designs wäre hier fehl am Platze.)

See architecture.

11.1.1 Inception

Start Oct-31,2012 15 minutes

Project definition

Brainstorming 30 minutes

11.1.2 Elaboration

11.1.3 Construction

11.1.4 Transition

11.2 Client Management

11.2.1 Customer (Decisionmaker)

Gebhardt

11.2.2 User

11.3 Team Management

Team name: –

Team members: Gebhardt

Team leader: –

Team minute taker: –

11.4 Organization Management

The project is an assignment within the OSWE lecture. It will be done mostly within the time of the lecture. Therefore, the allocated time is very limited, maybe around 12 hours.

11.5 Protocols

12 Architecture

12.1 Development Environment

- Programming language: Java
- Operating system: Any OS where Java runs.
- IDE: Ovid

12.2 Class Libraries

- Java API
- kj

12.3 Components

No components will be bought.

12.4 Frameworks

12.5 System/Subsystem Structure

There is not yet a subsystem structure. Instead we consider the database system as another system, which will not be developed by this project.

12.6 Error Handling

- Error in the code: Program should crash with
`e.printStackTrace ()`
- User errors should be handled.

12.7 Persistence

Client requirement: An object-oriented database should be used. Probably db4o.

12.8 Testing

(Wie wird getestet? Hier keine Testresultate.)

The use cases are played, eventually with extreme scenarios.

12.9 Distribution of Objects – Networking

- Application server
- Database server
- Just only XHTML Web clients

12.10 Security

All users of the system are required to log in.

12.11 User Interface

Web interface constructed with HWT will be used.

12.12 Systemstart and -end

13 Requirements-Analysis

(Spezifische Anforderungen des Kunden werden hier gelistet.

Eventuell werden eigene Anforderungen an das System getellt.)

14 Analysis

(Was soll das System tun?)

14.1 Brainstorming

14.2 System Description

14.3 Use Cases

14.4 Specialist Knowledge

14.4.1 Literature

14.5 Analysis of Entities

14.5.1 Datastructures

14.5.2 Behaviour

14.6 Dataflows

14.7 Collection of Ideas

15 Design

15.1 Lists of Nouns

15.2 CRC-Cards

15.3 Class Diagrams

15.4 Description of Classes

(Oft sind die Namen von Klassen so gewählt, dass nicht unmittelbar klar ist, was diese Klassen wozu und mit welchen anderen Klassen tun. Diese Informationen sollen hier gegeben werden. Es müssen daher nicht alle Klassen beschrieben werden. Ferner genügen nur die wichtigsten Methoden.)

15.5 Design Patterns

15.6 Behavioural Diagrams

16 Implementation

17 Testing

17.1 Use Cases

18 System Integration

18.1 Subsystems

18.2 Components

18.3 Libraries

19 Deployment

19.1 User Documentation

19.2 Delivery, Installation

19.3 Introduction and Training

19.4 Production System, Operation of System

19.5 Maintenance

19.6 Extensions

20 Literature