Systemanalse & -Entwurf

- Course Outline
- Introduction & Definitions

Contents

- 9 Lectures
- Theory & practical experience
- Voluntary Exercises & Bonus for the exam
- Document repository & literature online
  - Use Foxit reader for annotation

Lecture Introduction

- Terms & Definitions
  - Its all about communication
- Functional Analysis
  - Game Design Document
- Rapid Prototyping
  - Paper Prototype

Course outline

- The job of the system analyst
- How to extract requirements
- Documentation of designing a game

Paper prototyping

- Prototyping
- Fail fast fail often
- Try a software before programming it

Lecture System Analysis

- How to extract system descriptions
  - Monitoring techniques
  - Interview techniques
  - Feedback techniques
- Requirements Engineering
- A general process to gather requirements

Lecture System Modeling

- Specification documents
- Software life cycle
- Software Requirements

Course outline

- Analytical Specification
- Documentation Forms of Requirements
- Properties of Requirements
- User Story
- Templates

Lecture Structural Modeling

- UML class diagram
  - States & Behavior
  - Encapsulation
  - Inheritance
  - Modularity
  - Aggregation / Komposition
  - Interfaces & Abstract class

Lecture Object-Oriented Programming

- Revisit concepts of object-orientation
- Encapsulation concept
- Inheritance
- Visibility of behavior methods and state variables

Lecture Behavioral Modeling

- UML Activity Diagram
  - States
  - Transitions
  - Conditions
- UML Sequence Diagram
- UML State Diagram
- User Stories
- Use Cases

Lecture Project Management

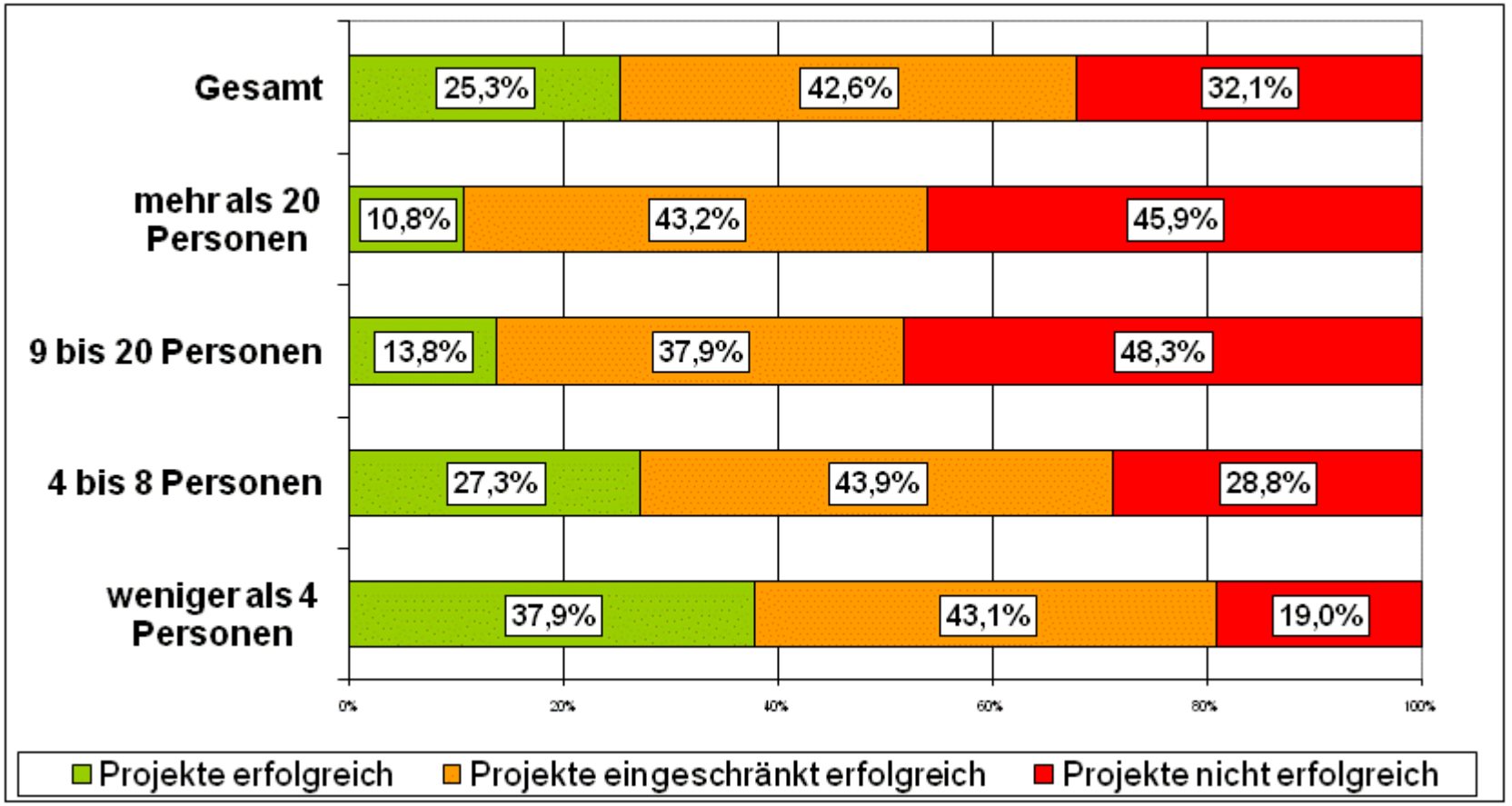
- Lifecycle of requirements
- System development models
  - Scrum Versus Waterfall Versus V-Model
- Testing
- Prototyping
  - Wireframes / Mockup

Motivation

- 40% -80% of Software Projects fail or exceed cost or time spans (for reference see list of literature on failing projects)
- Main Reasons are
  - Misunderstanding in the requirements
  - Requirements changes dynamically during the project
  - Some stakeholders are not involved
  - Large team sizes (see next slide)

Nr. 1 reason for failure

- Large team sizes & long-term projects



http://www.pentaeder.de/projekte/2009/07/02/erganzungen-zum-chaos-report/

Definitions

- System
  - A collection of components organized to accomplish a specific function or set of functions
- Software Engineering (Systemanalyse)
  - (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
  - (2) The study of approaches as in (1).

IEEE Standard Glossary of Software Engineering Terminology, 1990.

Definition Requirement

- (1) A condition or capability needed by a user to solve a problem or achieve an objective.
- (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.
- (3) A documented representation of a condition or capability as in (1)or (2).

IEEE Standard Glossary of Software Engineering Terminology, 1990.

Definition Requirement Analysis

- (1) The process of studying *user needs* to arrive at a definition of system, hardware, or software requirements.
- (2) The process of studying and refining system, hardware, or software requirements.

IEEE Standard Glossary of Software Engineering Terminology, 1990.

Stakeholder (Rollen)

Rollenträger	Primäre Interessen
Auftraggeber (Bauherr)	Optimale Erreichung seiner Ziel- und Wunschvorstellungen, Nutzen- und Kostenfragen
Systemdesigner (Architekt)	Konzeptions- und Entwurfsfragen, um das Konglomerat an gewünschten, erhofften Softwareleistungen (≡ Benutzermaschine) auf die konkrete (das heißt vorhandene oder zubeschaffene) Computer (≡ Basismaschine) abbilden zu können.
Programmierer (Bauunternehmer)	Verständlichkeits- und Interpretationsfragen, um durchschaubare und zuverlässige Programme erstellen zu können.
Betreiber & Operateure (Hausmeister)	Leichte Bedienbarkeit und Zuverlässigkeit
Wartungsdienst (Handwerker)	Änderungs- und Anpassungsfreundlichkeit, insbesondere Nachvollziehbarkeit der (Kontroll-)Strukturen
Weitläufig *Betroffene* (Nachbarn)	Fragen der Schadensbegrenzung, das heißt, Einflußnahme auf die Ausschaltung von negativen Wirkungen; Schnittstellen, Auflagen
Spätere *Benutzer* (Bewohner)	Akzeptanz, Softwareergonomie; Erfüllung der fachlichen Anforderungen zuverlässig und in hinreichender Qualität
Zu Beteiligende, zum Beispiel Betriebsrat oder Datenschutzbeauftragte (Baubehörden)	Erfüllung von Gesetzen, Vorschriften, Standards und Normen.

Systemanalyse für Softwaresysteme, Bonin, 2006.

Steps in Software Development

Aktivitäten der System-Entwicklung					
Ermitteln der Anforderungen (*Requirements-Engineering*)		*Konzeptionieren* des Systems		*Realisieren* des Systems	
Probleme analysieren*	Anforderungen definieren*	Bausteine spezifizieren*	Bausteine konstruieren*	Bausteine implementieren (programmieren)*	Bausteine integrieren (montieren)*

Systemanalyse für Softwaresysteme, Bonin, 2006.

Documents Overview & Phasis

- Current State Analysis (Ist-Analyse)
- To-Be Analysis (Soll-Analyse)
  - Analytical Specification (Lastenheft)
  - Technical Specification (Pflichtenheft)
- Design document (Entwurf)

Kriterium	Lastenheft	Pflichtenheft
Erstellungszeitpunkt:	Definitionsphase	Planungsphase
Intention:	*Stop-or-go*-Frage	Vertragsgrundlage
Detailierungsgrad:	Grobe Übersicht	Umfassende Beschreibung
Hauptakteure:	Auftraggeber Projektleiter (Fachexperte)	Auftraggeber Projektleiter Fachexperte Systemanalytiker

Charts in Requirement Analysis

	Structure	Behavior
Analytical Specification (Lastenheft)	Analytical Class- Object- Packet-	User Story Use-Case- Activity-
Technical Specification (Pflichtenheft)	Technical Class- Components- Distribution-	State- Sequence- Communication- Interaction-

Above assignment are not that strict, take what best suits your needs