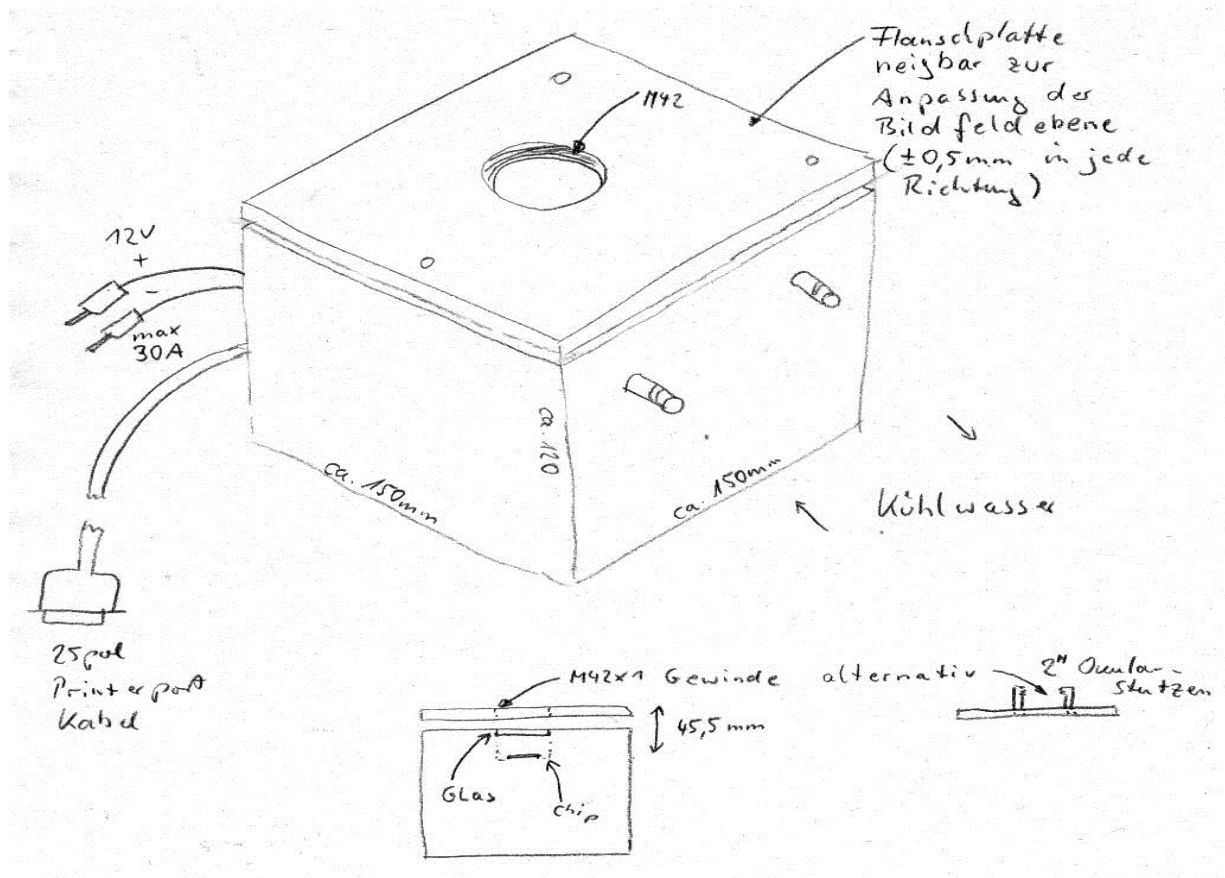
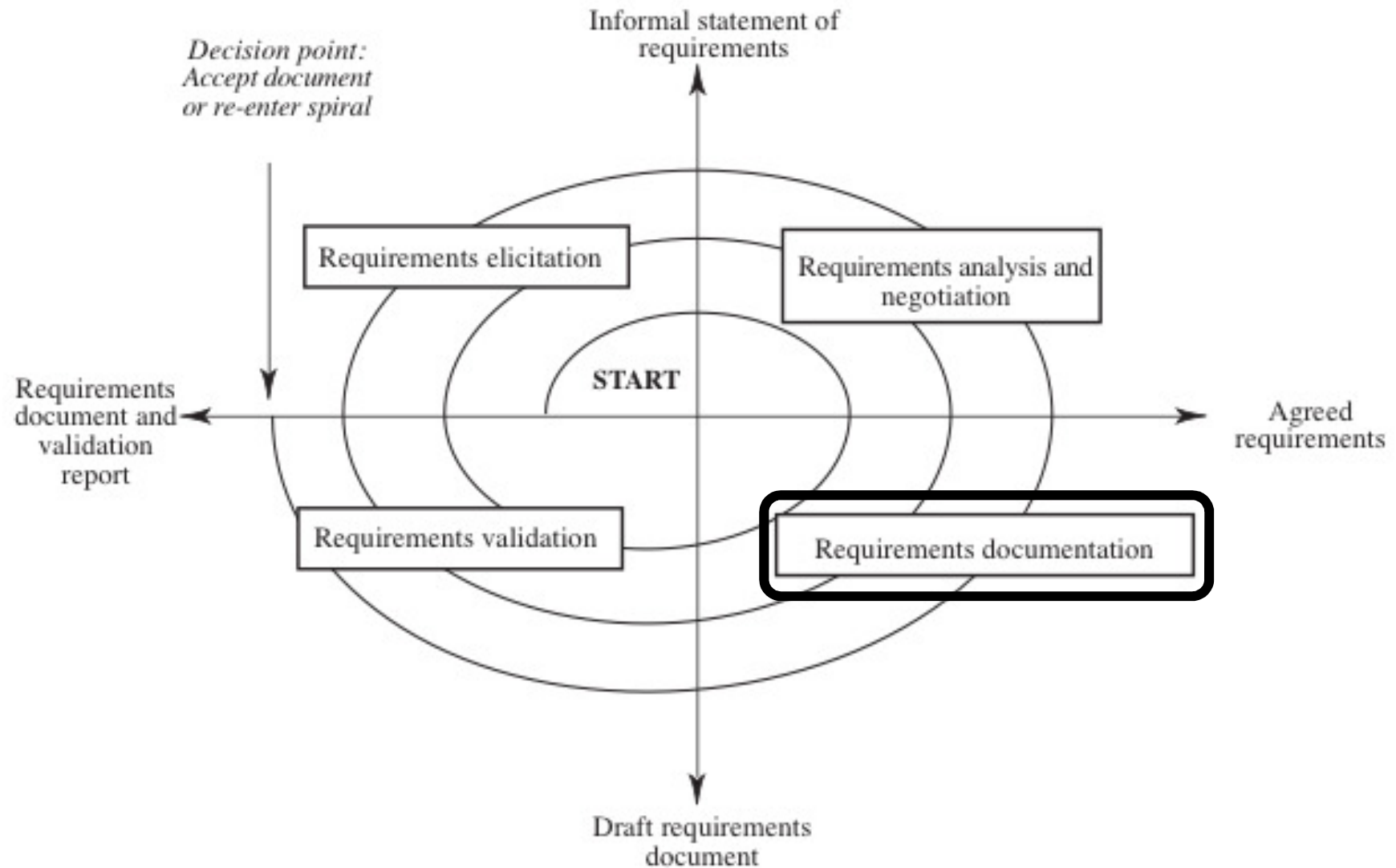


# Requirements Specification Overview



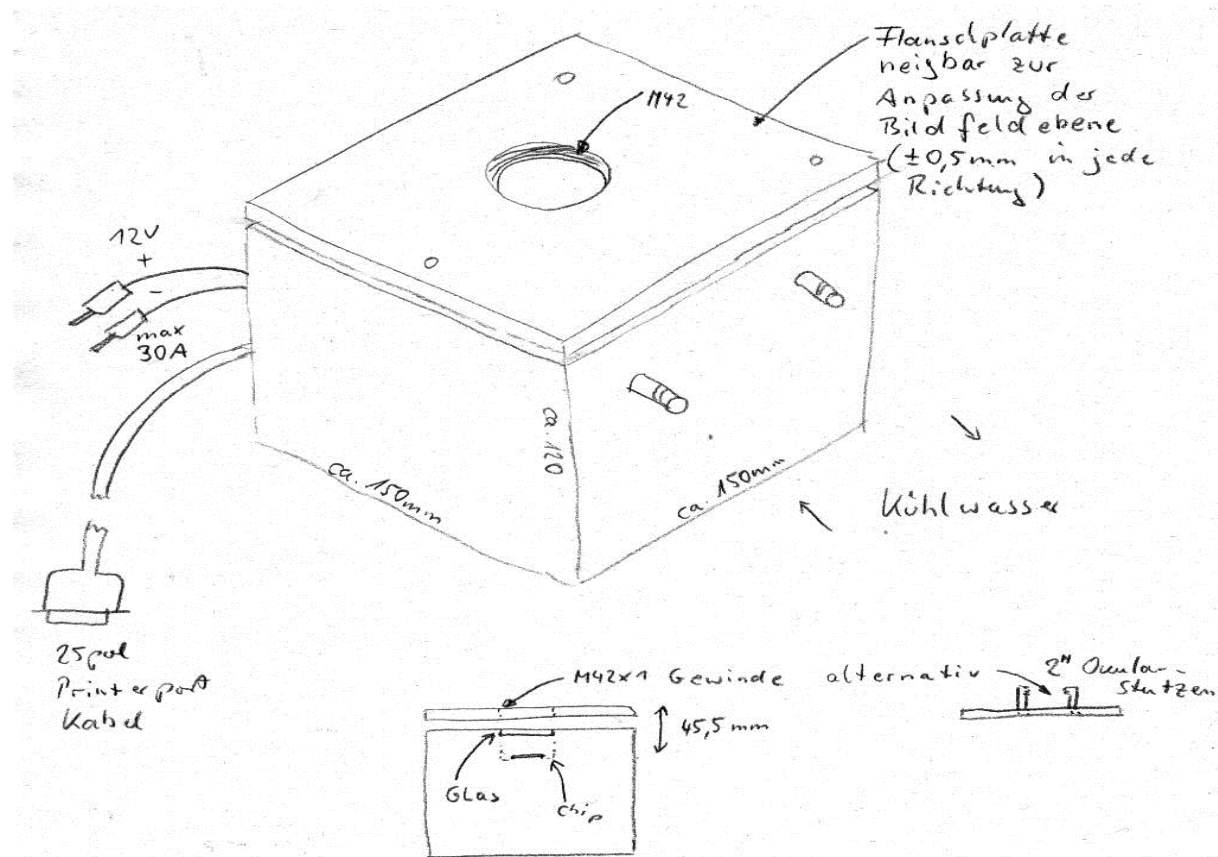
## Elicitation and the RE Process



*The Spiral RE Process (G. Kotonya and I. Sommerville, 1998)*

## Discussion

What is a and what do we expect from a Software Requirements Specification (SRS)?



## **Definition: “Software Requirements Specification”**

*“Requirements specification:* The activity of translating the information gathered during the analysis activity into a document that defines a set of requirements. Two types of requirements may be included in this document.”

- *User requirements* are abstract statements of the system requirements for the customer and the end-user of the system (e.g., Use Cases);
- *System requirements* are more detailed descriptions of the functionality to be provided (e.g., Functional, NFR).

**Definition: “Software Requirements Specification”**

*A requirements specification is a systematically represented collection of requirements, typically for a system or component, that satisfies given criteria.*

## Why do we need a good SRS?

Requirements are the basis of the system development

- Establish the basis for agreement
- The SRS helps to determine if the software specified meets the needs of customer and end-user
- The SRS often is the basis of contracts

Reduce the development effort

- The preparation of the SRS forces the concerned people to think of all of the requirements before design begins (reduces later redesign, recoding, and retesting)

Provide a basis for estimating costs and schedules

- The SRS is a realistic basis for estimating project costs and can be used to obtain approval for bids or price estimates

*Source: Source: ANSI/IEEE Std 830-1998;*

<http://www.microtoolsinc.com/Howsrs.php>

## Why do we need a good SRS?

Provide a baseline for validation and verification

- A good SRS is a basis to develop validation and verification plans much more productively.
- The SRS can be also used to create the test plans

Facilitate transfer

- The SRS makes it easier to transfer the software product to new users or new machines

Serve as a basis for enhancement

- The SRS discusses the product but not the project that developed it - it serves as a basis for later enhancement of the finished product

*Source: Source: ANSI/IEEE Std 830-1998;  
<http://www.microtoolsinc.com/Howrsrs.php>*

## Typical Specification Problems

Noise - information not relevant to the problem, or a repetition of existing information

Silence - Omitted requirements; important aspects of the problem are simply not mentioned (29% of errors)

Over-specification - aspects of the solution are mentioned as part of the problem description

Wishful thinking - a realistic solution would be difficult or impossible

Ambiguity - more than one meaning for a given word or phrase

Reader subjectivity - cultural background, language, personal experience, etc.

Forward references - aspects of a problem not yet defined

Oversized documents - difficult to understand, use and maintain

Customer uncertainty - inability to express specific needs results in vague descriptions

Incorrect assumptions - (49 % of requirements problems)

Multiple representations - creation of the same requirements; maintenance, translation and synchronization problems

Little to no user involvement - instead by requirements solicitors, business analysts, domain experts or even developers

*Source: Melnik G., Read K., Maurer F.: Suitability of FIT User Acceptance Tests for Specifying Functional Requirements: Developer Perspective", Proc. XP/Agile Universe 2004, LNCS 3134, Springer 2004*



## Characteristics of a good SRS

Correct: Correct and Ever Correcting

Unambiguous: ...every requirement stated should have only one interpretation

Complete: Do we have all the necessary things to build the software system?

Consistent: The stated requirement does not contradict other requirements; the same term is used for the same item in all requirements.

Prioritization: Some requirements might be more important than others

Verifiable: How to measure these requirements: “The system should be stable”; “The system should provide good performance”; “The system should have an adequate User Interface”, ...?

Modifiable: Make the SRS changeable

Traceable: Why do we need this requirement?

*Source: Source: IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications – Description, <http://www.microtoolsinc.com/HowSRS.php>*

## **Standardisierte Dokumentenstrukturen**

### **Die Verwendung von Standardgliederungen hat folgende Vorteile:**

- Erleichtern die Einarbeitung neuer Mitarbeiter
- Ermöglichen eine schnellere Erfassung (Verständnis) ausgewählter Bereiche & Inhalte
- Ermöglichen selektives Lesen / Überprüfen
- Ermöglichen eine automatische Überprüfung (zumindest auf Vollständigkeit)
- Ermöglichen eine einfachere Wiederverwendbarkeit

## Häufig angewendete Standardisierte Dokumentenstrukturen

	Customer	Supplier
Englisch	IEEE 1362	IEEE 830
Deutschland (V-Modell XT)	Lastenheft	Pflichtenheft
Schweiz (HERMES)	Pflichtenheft	(Offerte)
Rational Unified Process (RUP)	Stakeholder Request	Vision, Use Cases, Supplementary Requirements
Scrum	Product Backlog	Sprint Backlog

**Wichtig: Templates an die Umgebung und die Bedürfnisse des Projekts anpassen!**

## **Muster aller Anforderungsdokumente**

**Die von Standards und Normen vorgeschlagenen Templates ähneln sich in drei Grundelementen:**

### **Übersicht, Kontext, Scope**

- Für alle (d.h. Programmierer ebenso wie Management) lesbare kurze Übersicht über das Projekt.
- Keine Details, sondern Abgrenzung, Ziele, angestrebte Lösung
- RUP: Vision Document

### **Funktionale Anforderungen**

- Spezifikation der Funktionalität des Systems
- RUP: Use Case Model

### **Nichtfunktionale Anforderungen**

- RUP: Supplementary Specification

## **Grobstruktur nach IEEE-Standard 830-1998**

**Einleitung**

**Ziele**

**Überblick / Einsatzbereich / Kontext**

**Globale Anforderungen (Randbedingungen,  
generelle Anforderungen, Normen, Standards)**

**Einzelanforderungen (z.B. pro Geschäftsprozess,... )**

- Funktionen (Use Cases)
- Geschäftsobjekte (Klasse)
- Zustände (Regeln)
- Input/Output-Definitionen
- Schnittstellen
- Nicht-Funktionale Anforderungen

**Glossar**

**Spezielle Anhänge (z.B. bestehende Konzepte, Spezifikationen; GUI-Style-Guides, ...)**

## **Beispiel RUP-Artefakte**

### **Vision**

- Liefert Überblick und Kontext
- Dokumentiert eine einheitliche Sicht aller Stakeholder

### **Use Case Model**

- Beinhaltet Grossteil der funktionalen Anforderungen

### **Supplementary Specification**

- Beschreibt Rahmenbedingungen, nichtfunktionale Anforderungen und (restliche) funktionale Anforderungen

## Beispiel Volere: Checkliste für Anforderungen

### Project Drivers

1. The Purpose of the Product
2. Client, Customer, Stakeholders
3. Users of the Product

### Project Constraints

4. Mandated Constraints
5. Naming Conventions and Definitions
6. Relevant Facts and Assumptions

### Functional Requirements

7. The Scope of the Work
8. The Scope of the Product
9. Functional and Data Requirements

### Non-Functional Requirements

10. Look and Feel
11. Usability and Humanity

- ☐ 12. Performance
- ☐ 13. Operational
- ☐ 14. Maintainability and Support
- ☐ 15. Security
- ☐ 16. Cultural and Political
- ☐ 17. Legal
- ☐ **Project Issues**
- ☐ 18. Open Issues
- ☐ 19. Off-the-shelf Solutions
- ☐ 20. New Problems
- ☐ 21. Tasks
- ☐ 22. Migration to the New Product
- ☐ 23. Risks
- ☐ 24. Costs
- ☐ 25. User Documentation
- ☐ 26. Waiting Room
- ☐ 27. Ideas for Solutions

## **Verwendung von Anforderungsdokumenten**

**für die Planung**

**für den Architekturentwurf**

**für die Implementierung**

**für den Test**

**beim Änderungsmanagement**

**bei der Systemnutzung und Systemwartung**

**im Vertragsmanagement**



## **Spezifikation: Unterschiedliche Begriffe und Inhalte zwischen DIN und Schweiz (z.B. HERMES (Die Schweizerische Projektführungsmethode))**

Das *Lastenheft* beschreibt die vom Auftraggeber festgelegte Gesamtheit der Forderungen an die Lieferungen und Leistungen eines Auftragnehmers innerhalb eines Auftrages. Das Lastenheft beschreibt in der Regel also, was und wofür etwas gemacht werden soll.

Das *Pflichtenheft* hingegen beschreibt, wie und womit etwas realisiert werden soll. Nach enthält das Pflichtenheft die vom Auftragnehmer erarbeiteten Realisierungsvorgaben aufgrund der Umsetzung des vom Auftraggeber vorgegebenen Lastenheftes.

*DIN 69905*

### **Achtung Vorsicht:**

Schweizer HERMES Pflichtenheft  $\approx$  Lastenheft

Lastenheft  $\approx$  Spezifikation

## Spezifikation: Ergebnisfluss im HERMES

