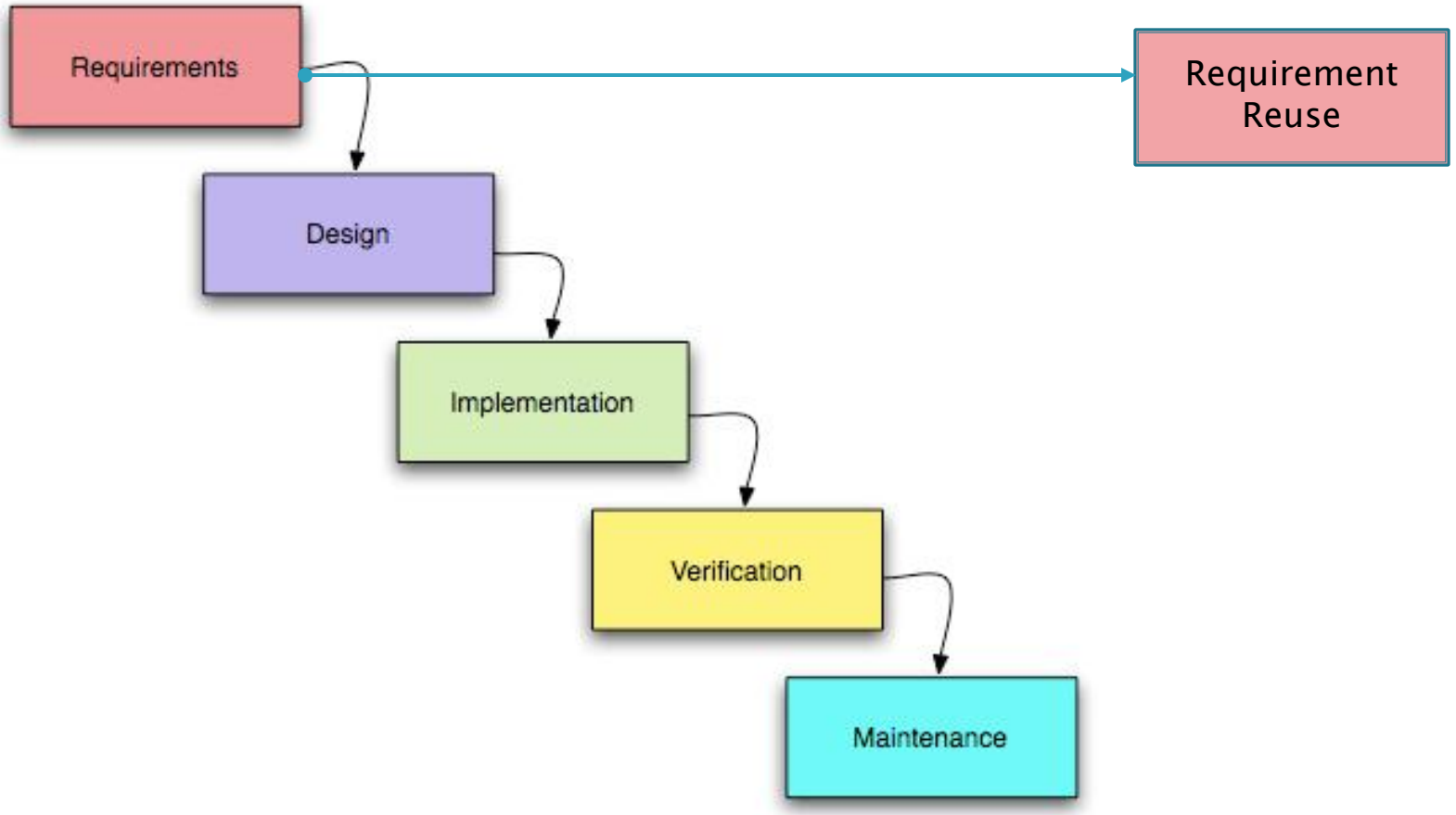
Module 2. Requirement Reuse

Dr. Shiping Chen
shipingchen2005@gmail.com

Where are we?



Part of this figure is from: http://en.wikipedia.org/wiki/File:Waterfall_model.png

Outline

- Overview of Software Requirement Specification (SRS)
- Challenges in developing a SRC
- How can Reuse help?
- Task 1 of Assignment

Software Requirement Analysis

- Traditionally, regarding as 1st stage of SDLC
- Including the following tasks/activities:
  - *Collecting Requirements*: to get the users' needs of the software systems, also called requirements gathering
  - *Analysing Requirements*: to check the requirements and resolve the conflicts requirements with the stakeholders if any
  - *Recording Requirements:* to document the requirements in a requirements document, i.e. **Software Requirement Specification** (SRS)

Software Requirement Specification (SRS)

- A requirement specification for a software system, which should capture:
  - The Key actors and components in the system
  - The Key behaviours/functionalities of the system, (usually via use cases)
  - The Key non-functionalities, such as performance, security, standards and compliance, etc.
  - The Key constrains/limitations in terms of (technical and human) resources and timing
- Very important for the success of the system!

Some Facts

- 71% of software project failures because of *poor requirements management*
  - From *CIO Magazine*
- Surveyed 9,236 IT projects, found that the top 3 causes of project failure were lack of user input, *incomplete requirements* or *changing requirements*
  - Standish CHAOS Report
- More than 50% software projects perceived that they had major problems in the area of requirements specification
  - From Survey of European Software Organizations

Challenges of SRS

- No enough time to fully understand customers' requirements
  - Today SDLC of many software projects become shorter and shorter
- Even **no customers** at beginning for some new companies/new software products
  - So sometimes you must assume you are the customers!
- Ever-changing requirements
- Not a complete technical issue
- Heavily depend on experiences and domain knowledge

Reuse can help…

- Reuse (follow) a suitable SR analysis process;
- Reuse (refer to) a standard SRS as template
- Reuse (select) a good requirement management tool

Software Requirement Analysis

- Different SD mythologies has different SR analysis processes and cycles:
  - Classical Waterfall
  - RAD (Rapid Application Development)
  - DSDM (Dynamic systems development method)
  - Agile
- …but they all must address the following issues at different levels:
  - Who?
  - What?
  - Why?
  - How?

At Business Level

Issues	Details
Who	are the stakeholders of the software?
What	business goals to achieve with the software
Why	the software can help achieve the goals
When	the software should be delivered

At User Level

Issues	Details
Who	will use the software system/product
What	User Interface (UI) looks like?
Why	design in that way?
How	to interact with the users?

At System/Product/Operational Level

Issues	Details
Who	deploy, operate, maintain/support the software system/product?
What	• key functional requirements; and • non-functional requirements (QoS)
Why	needs these functionalities and non-functionalities?
How	to use these functionalities (use cases)?

Levels & Types of Requirements

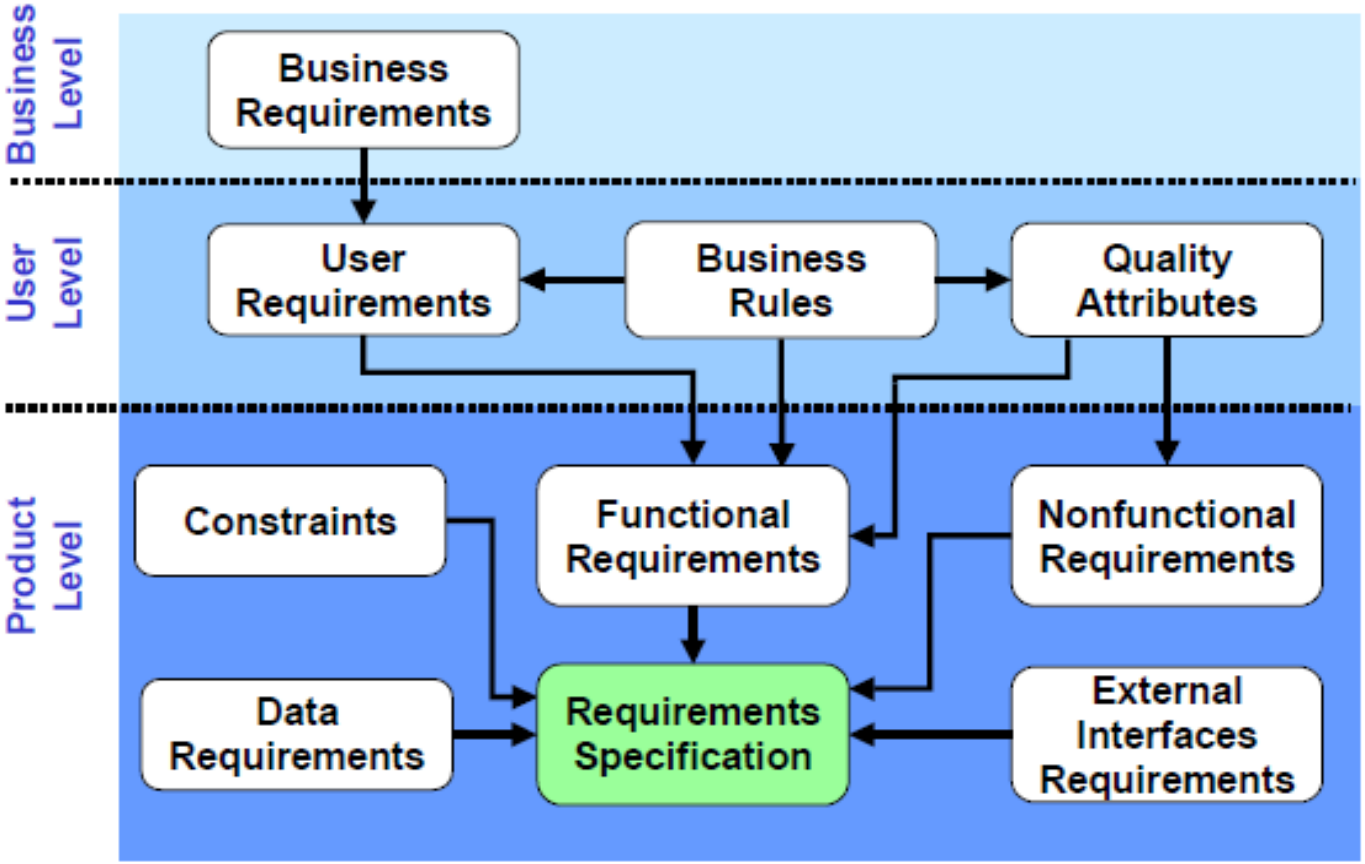


Figure 1 Levels and types of requirements

From: http://www.westfallteam.com/Papers/The_Why_What_Who_When_and_How_Of_Software_Requirements.pdf

Software Requirement Specification (SRS)

- A specification to document the result of software requirement analysis

- A General Organization of an SRS includes:
  - Introduction
  - Overall Description
  - Specific Requirements
  - Appendix

Introduction to SRS

- **Purpose**
  - Delineate the purpose of the SRS;
  - Specify the intended audience for the SRS
- **Scope**
  - Identify the software product(s) to be produced by name
  - Explain what the software product(s) will, and will not do
  - Describe the application of the software being specifed
- **Definitions**
  - provide the definitions of all terms, acronyms, and abbreviations
- **Overview/Organization**
  - Describe how the rest of the SRS is organized.
- **References**
  - Provide a complete list of all documents referenced in the SRS

Overall Description of a SRS

- **Product Perspective**
  - Give an overview of the software system/product to
  - A picture is worth 1024 words
- **Product Functions**
  - Key functions that the software system/product will offer
- **User Characteristics**
  - Who is likely to use the software
  - What background knowledge and skills are expected
- **Constraints, Assumptions and Dependencies**
  - What assumptions are made
  - What may affect on the development of this software

Specific Requirements (Major Part)

- External interfaces
  - Describe all inputs into and outputs from the software system
- Functions (Business Rules)
  - List what must be done from an input to an output
- Performance requirements
  - Specify both the static and the dynamic numerical requirements placed on the software
- Data Model
  - Specify the logical requirements for any information that is to be placed into a database
- Design constraints & Key Features
  - Specify design constraints that can be imposed by other standards, hardware limitations, etc.
- Software system attributes
  - E.g. Reliability, Availability, Security, Portability, Maintainability ...

Summary for SRS

- The above SRS is based on IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998)
- It is a generic SRS template
- Should be reused in a justice way for a specific software system/product, e.g.
  - Add/Remove some sections; and/or
  - Focus on some particular sections more than others
- Should use standard software model languages as much as possible, such as UML.

Software Requirement Management (RM)

- A process of documenting, analysing, tracing, prioritizing and agreeing on requirements and then controlling change and communicating to relevant stakeholders
- Should be a continuous process throughout a software development project
- Focus on *collaboration* (contribution and sharing), *traceability*, and *change management*
- Usually by means of a RM tool…

RM Tools

- IBM Rational Doors & RequisitePro
  - http://www-01.ibm.com/software/awdtools/doors/
  - http://www-01.ibm.com/software/awdtools/reqpro/
- Borland CaliberRM
  - http://www.borland.com/us/products/caliber/index.aspx
- HP Requirement Management
  - http://www8.hp.com/us/en/software/software-solution.html?compURI=tcm:245-937050
- ViTech CORE
  - http://www.vitechcorp.com/products/core.shtml
- Accept 360 RM
  - http://www.accept360.com/solutions/accept360-requirements/

Recourses/Further Readings

- **Must:**
  - UML basics: An introduction to the Unified Modeling Language
  - IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998) (Access to IEEE Xplore needed)
- **Optional:**
  - Sue Black, Paul P. Boca, Jonathan P. Bowen, Jason Gorman, and Mike Hinchey. 2009. Formal Versus Agile: Survival of the Fittest. *Computer* 42, 9 (September 2009), p37-45
  - Book: Karl E. Wiegers "Software Requirements 2", published by Microsoft Press:, 2003
  - Book: Stephen Withall "Software Requirement Patterns", published by Microsoft Press, 2007
  - "Requirements Management Tools Survey". International Council on Systems Engineering. Retrieved 2009-11-10

Assignment Introduction

- The assignment is a project across the whole SDLC
- It consists of a few small tasks;
- Focus on solving real problem using the software reuse technology learnt from this lecture;
- But I am looking forward to more innovation and surprising……

Assessment Assessment

- The assignment will be assessed in the following forms:
  - Presentation – Each team (or individual) will be given 15 minutes to present your assignment, followed by 5 minutes questions/discussions, which will contribute 40% of your final mark
  - Examination – Some questions in the final exam can be answered with the contents of the assignment. This means the assignment is not only for the 40%

Task 1 of Assigment

- *Assume that you want to start up a company to sell cars in an innovative way. The basic idea is a sell car online with the lowest price by:*
  - *Allow a customer to browse & customize different types of cars online*
  - *Allow a customer to put a price enquire for his customized car*
  - *Allow car-dealers to bid for the price enquire*
  - *Collect the cheapest deal to notify the customer (+5% broker fee)*
  - *If the customer accept the deal, pay 10% of the price to secure the deal*
- **Please develop a SRS (Software Requirement Specification) for the project**

Advices for Task 1

- Do not have to fully use the SRS template
- Use the part that you need
- Capture the key requirements from software viewpoints
- A picture is worth a thousand words – try to use 'use case diagrams' to capture the software requirements
- Use the standard notations if possible, such as UML
- You can work on the assignment either individually or in a team
- I expect a number of small teams (3~5 persons)

Any Questions?