

# SE 311- Software Requirements Engineering

## Lecture # 1: Overview

**Ayesha Kanwal**

Department of Computing (DoC),  
SEECS, NUST



# *introduction*

# Agenda

- ❖ Course Outline / Contents
- ❖ Course Grading Criteria
- ❖ Books, material
- ❖ Class rules
- ❖ Introduction to Software Requirements Engineering
  - ❖ Importance / Challenges of Software
  - ❖ Importance of requirements
  - ❖ Need for requirements engineering
  - ❖ What are requirements?



# Why it is important for you?

## ❖ Goals of this course:

- ❖ To enable you to develop your skills in understanding why user needs are so hard to express, capture and specify.
- ❖ To learn about the shortcomings of best practices and to be able to choose among the candidate approach
- ❖ To learn about data-driven methods for requirements engineering
- ❖ To Identify stakeholders and work with them effectively by applying a range of
  - ❖ elicitation techniques,
  - ❖ specify requirements using requirements templates
  - ❖ modeling techniques,
  - ❖ validate and prototype requirements,
  - ❖ negotiate and prioritize requirements,
  - ❖ and manage changing requirements.

# SE-311 Course Topics (18 Weeks / 45 Lectures)

- ❖ Introduction to Requirements Engineering
- ❖ Software Requirements
- ❖ Requirements Engineering Processes
- ❖ Requirements Elicitation and Analysis
- ❖ Requirements Validation
- ❖ Requirements Management
- ❖ Methods for Requirements Engineering
  - ❖ Visual Modeling of Requirements
  - ❖ Overview of Tools for Requirements Modeling / Management
- ❖ Non-functional Requirements
- ❖ Goal-Oriented Requirements Engineering (GORE)
- ❖ Requirements Evolution
- ❖ Viewpoint techniques for requirement engineering
- ❖ Advanced Topics (either of them, if time permits):
  - ❖ Requirements for Self-adaptive Systems
  - ❖ Social / Mobile / Pervasive Requirements Engineering
  - ❖ Contextual Requirements Engineering

# Books we follow (also research papers)...

## ❖ Text Books:

### ❖ **Software Engineering 9/Ed**

By **Ian Sommerville**, *University of St. Andrews, United Kingdom*

ISBN-10: 0321313798 ISBN-13: 9780321313799

### ❖ **Requirements Engineering: Processes and Techniques**

By **Gerald Kotonya** and **Ian Sommerville**

ISBN: 978-0-471-97208-2, Published: Sep 1998

## ❖ Reference Books:

### ❖ **Requirements Engineering: Fundamentals, Principles, and Techniques**

By **Klaus Pohl**, 1<sup>st</sup> Edition,

ISBN-10: 3642125778 , ISBN-13: 978-3642125775, Publication Date: July 23, 2010

### ❖ **Requirements Engineering Handbook,**

By **Young, Ralph**,

Artech House, Publication Date: Nov. 2003

# Grading Policy

- ❖ **15%** OHT-1 Exam.
- ❖ **15%** OHT-2 Exam.
- ❖ **10%** Assignments.
- ❖ **10%** Quizzes.
- ❖ **10%** Term Project.
- ❖ **40%** Final Exam.

\* Minor Changes may be made in the criteria later (if required)

# Class Policy

## ❖ Assignments:

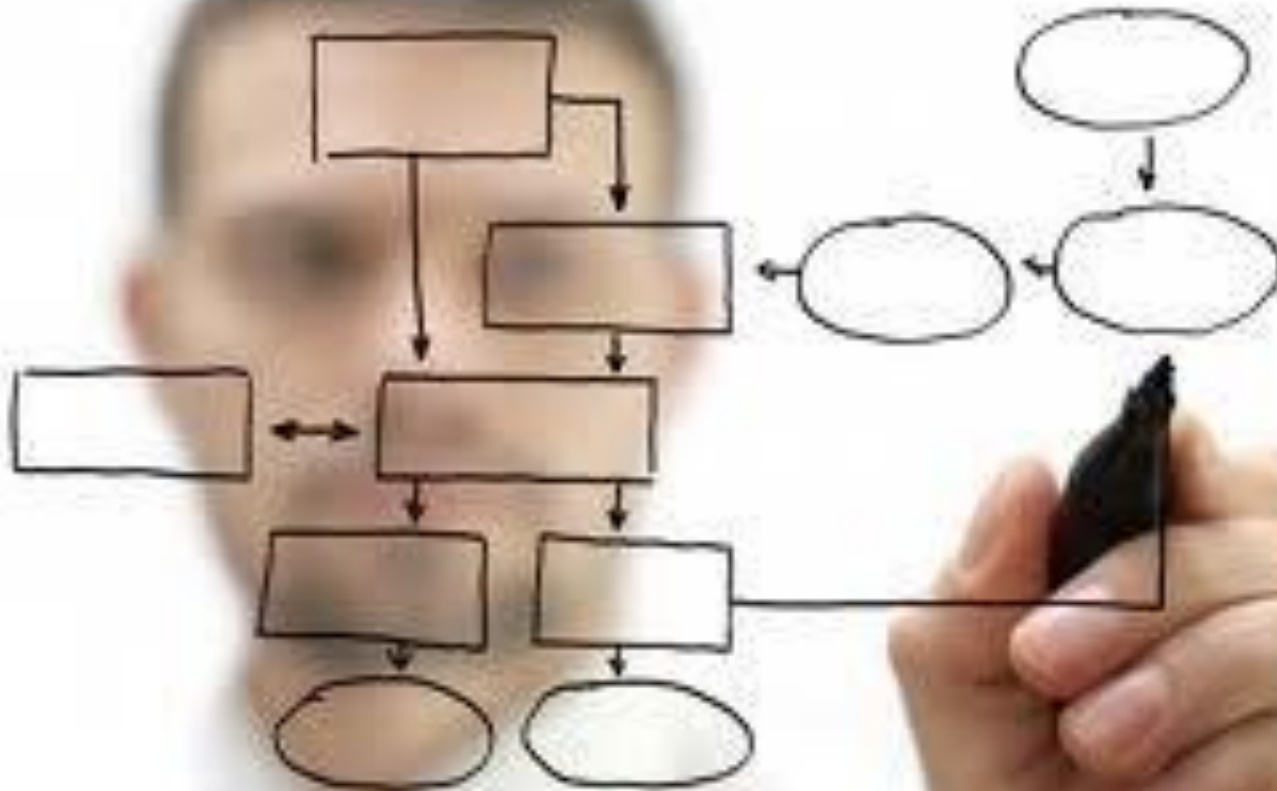
- ❖ Each assignment will count towards the total.
- ❖ Late assignments will not be accepted / graded.

## ❖ Quiz policy:

- ❖ Quizzes may be announced or unannounced.
- ❖ Missed quizzes will not be retaken.
- ❖ No best of quizzes policy.



# Lets be a good Requirements Engineer...



# Software is everywhere!



Images source: Google Images

# Background for Requirement Engineering

- ❖ Computer systems are *designed*, and anything that is designed has an intended *purpose*.
- ❖ If a computer system is unsatisfactory, it is because the system was designed without an adequate understanding of its purpose, or because we are using it for a purpose different from the intended one.
- ❖ Requirements Engineering provides a framework for understanding the purpose of a system and the contexts in which it will be used.



# Software-intensive Systems

- ❖ Systems that are mainly driven by software
  - ❖ **Information systems:** software for general use *\*usually*
  - ❖ **Embedded systems:** integrates hardware and software
- ❖ Fitness for Purpose
- ❖ Complexity of Purpose
- ❖ Because of this complexity of purpose, the design of software-intensive systems belong to a class of problems known as **wicked problems**.

# Software-intensive Systems

- ❖ Dealing with Complexity: Requirements Engineering offers a number of techniques for dealing with complexity of purpose, which are built into the various techniques:
  - ❖ Abstraction
  - ❖ Decomposition
  - ❖ Projection

# Software-intensive Systems

## ❖ Challenges of software-intensive systems

- ❖ Software-based innovations
- ❖ Increasing complexity
- ❖ Pressure to reduce costs
- ❖ Shorter development times
- ❖ Higher quality demands



# Challenges of Software

## ❖ Problems:

### ❖ Increased reliance on software

❖ e.g. cars, cell phones, web services, etc.

### ❖ Software is the biggest cost element for mission critical systems

❖ e.g. Boeing 777, A380 etc...

### ❖ High consequences of failure

❖ e.g. Intel Pentium bug: \$475 million

- Because of the bug, the processor can return incorrect decimal results (Floating Point Error)

# Challenges of Software

## ❖ Key Factors:

### ❖ Certification costs

❖ e.g. Boeing 777: >40% of software budget spent on testing

### ❖ Re-work from defect removal

❖ e.g. Motorola: 60-80% of software budget (was) spent on re-work

### ❖ Changing Requirements

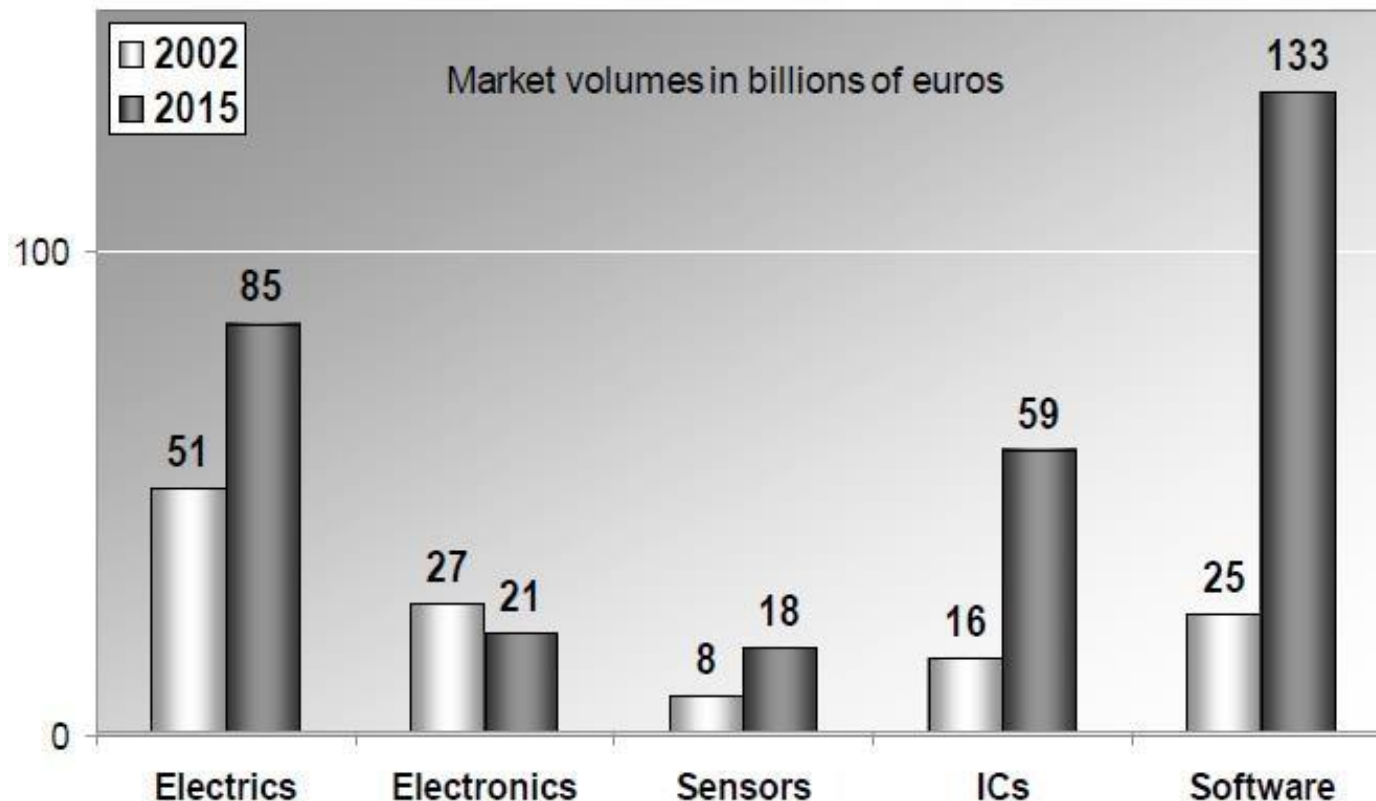
❖ e.g. California Department of Motor Vehicle system

❖ ... & many more!



# Importance of Software

- ❖ *Market volume of software in the automotive sector for the year 2002 and forecast for the year 2015 (according to [Honsig 2005])*



# Importance of Requirements Engineering

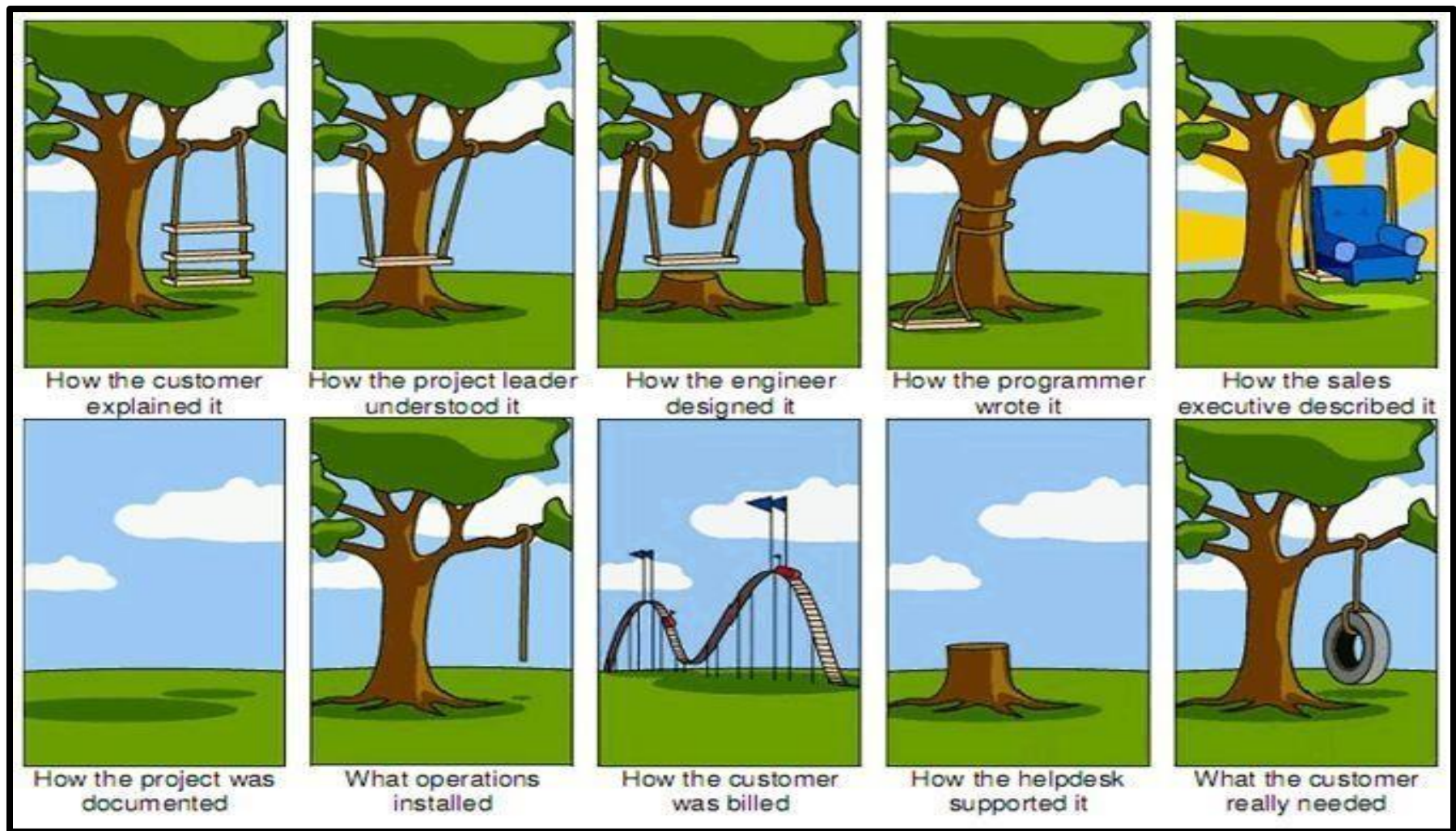


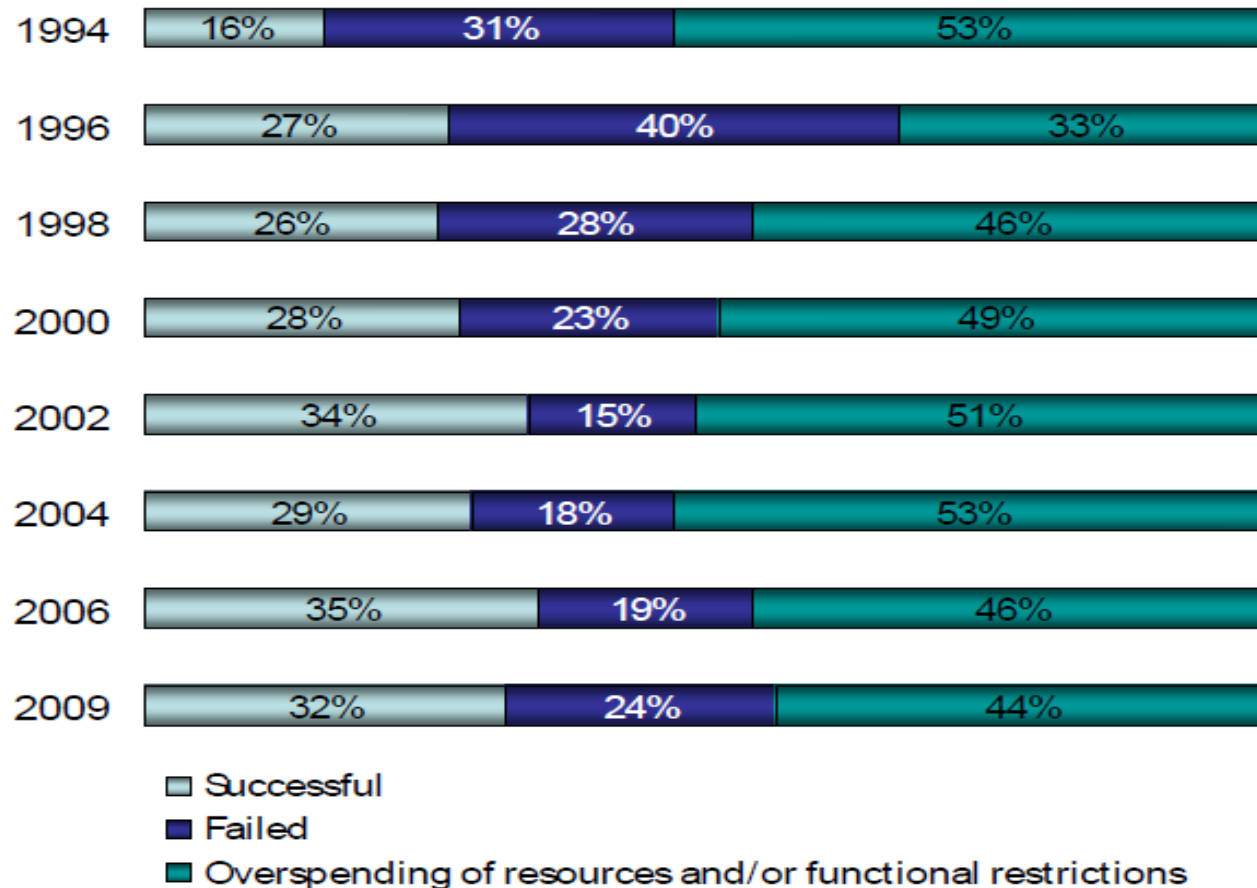
Image source: <http://knovelblogs.com/2012/08/30/the-importance-of-requirements-engineering/>

# Importance of Requirements Engineering

- ❖ High impact on project success
  - ❖ Many projects failed or finished with overspending and/or restricted implemented functionality
- ❖ Requirements defects are reason for approximately 50% of the failed projects
  - ❖ Example: London Ambulance Service
    - ❖ Poor requirements process that did not involve ambulance crews
      - **Result:** system was not able to operate under realistic conditions (i.e. sending too many ambulances to an incident) which endangered patients lives.
- ❖ Defects in Requirements Engineering cause high costs
  - ❖ Defect found during programming increases costs by factor 20
  - ❖ Defect found during acceptance test increases by factor 100!

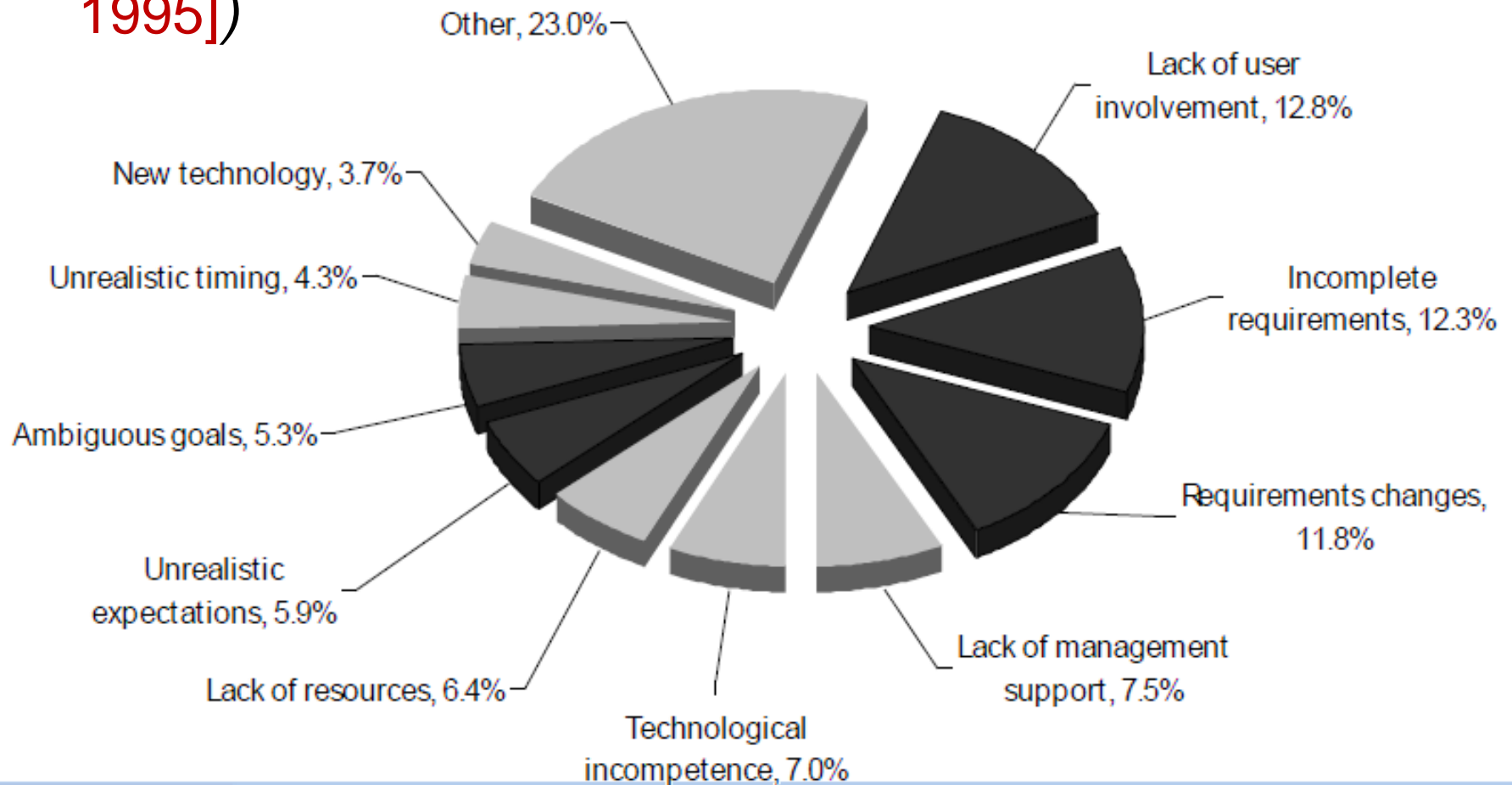
# Importance of Requirements Engineering

- ❖ *Project success rates from 1994 to 2009 taken from the Standish Group (CHAOS) studies [The Standish Group 2009]*



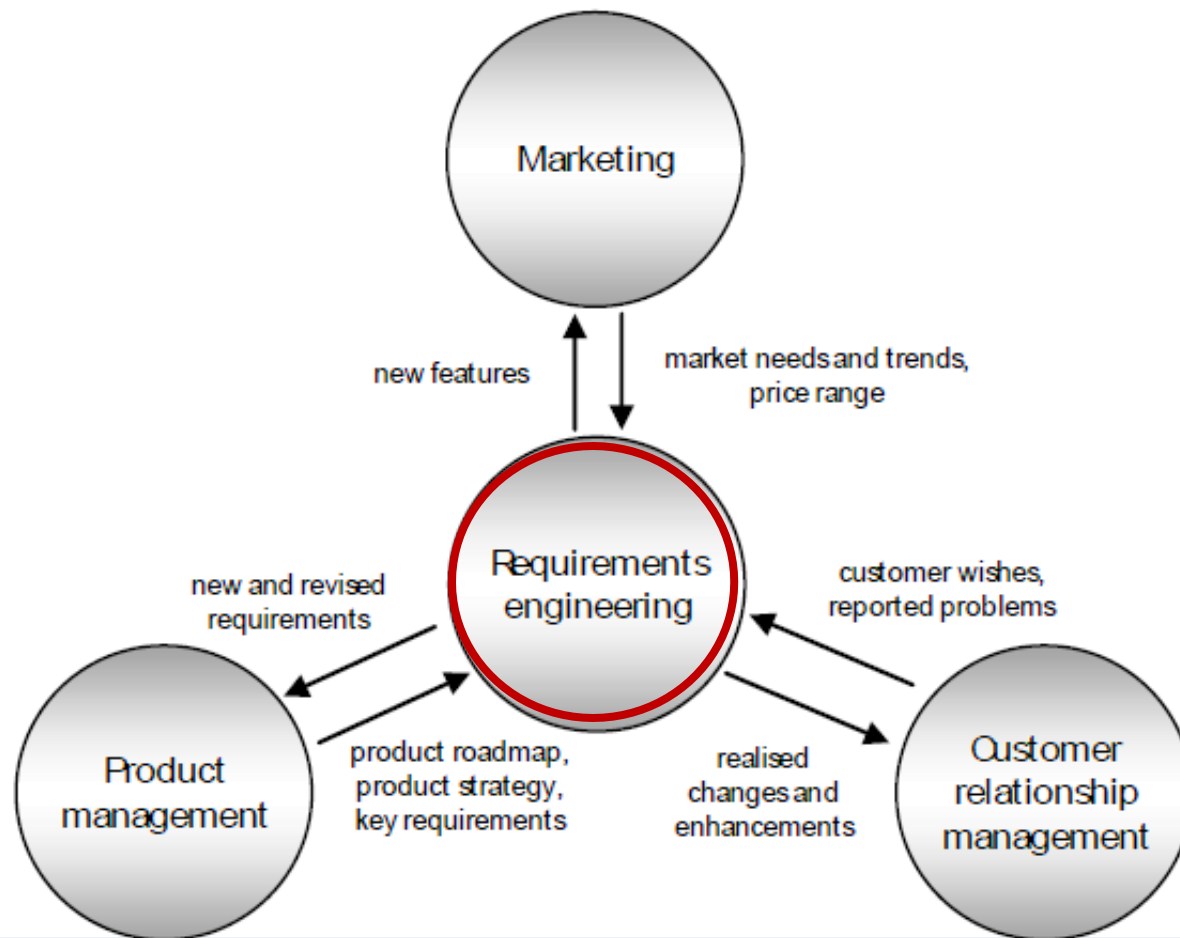
# Importance of Requirements Engineering

- ❖ *Reasons for resource overspend and/or functional restrictions (based on data from [The Standish Group 1995])*

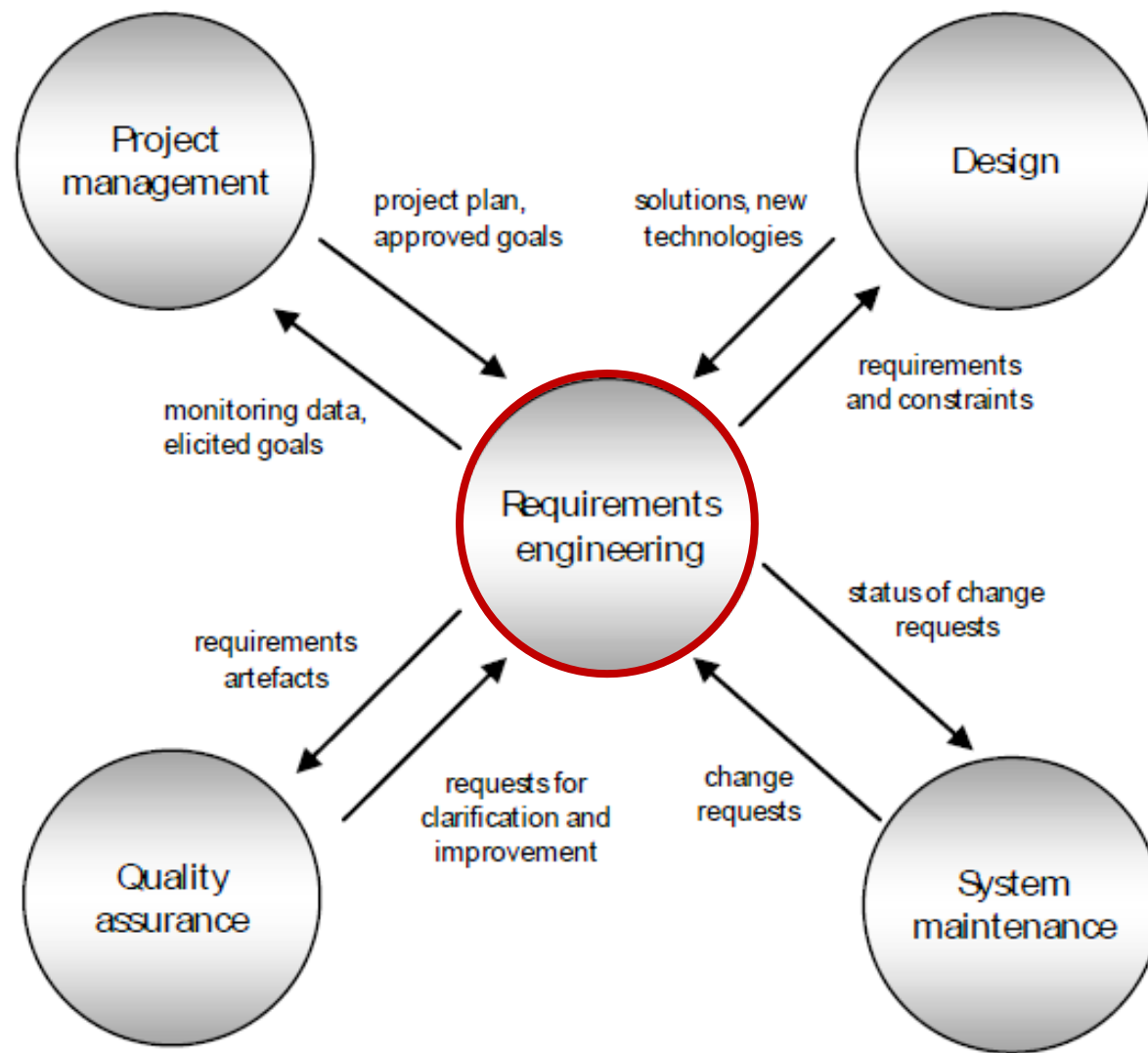


# Requirements Engineering & Organization

- ❖ *Interrelations between requirements engineering and other processes in the organization*



# Requirements Engineering & Organization





# Requirements Engineering Conference!



<http://requirements-engineering.org/>



# References:

- ❖ Lecture Slides from:

- ❖ Text Book:

  - ❖ **Requirements Engineering: Processes and Techniques** by Gerald Kotonya and Ian Sommerville

- ❖ Reference Book:

  - ❖ **Requirements Engineering: Fundamentals, Principles, and Techniques** by Klaus Pohl

# Thank you!

❖ Questions, Comments, Suggestions!