RE Overview

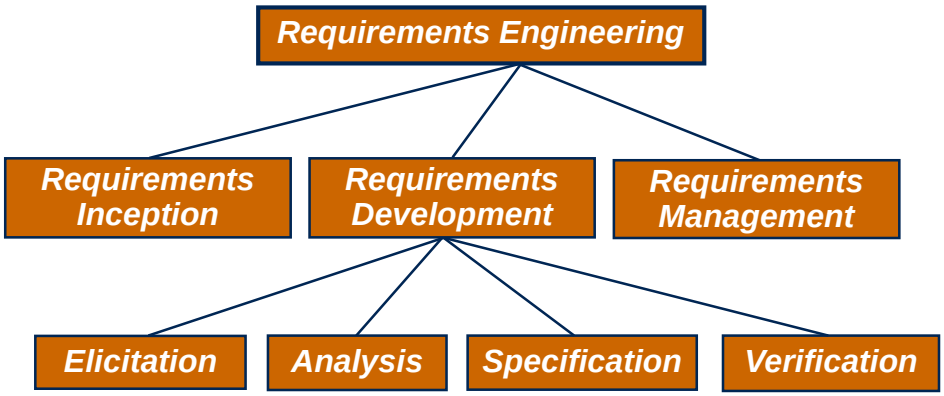
Based on presentations by G. Mussbacher, G.V Bochmann, N. Niu

What is "Requirements Engineering"?

- **Requirements Engineering (RE)** is:
  - The activity of development, elicitation, specification, analysis, and management of the stakeholder requirements, which are to be met by a new or evolving system
  - RE is concerned with identifying the purpose of a software system… and the contexts in which it will be used
    - How/where the system will be used
    - Big picture is important
  - Captures real world needs of stakeholders affected by a software system and expresses them as artifacts that can be implemented by a computing system
    - Bridge to design and construction
    - How to communicate and negotiate?
    - Is anything lost in the translation between different worlds?

2

Requirements Engineering Activities



Source: Larry Boldt, Trends in Requirements Engineering People-Process-Technology, Technology Builders, Inc., 2001

3

About these RE Activities…

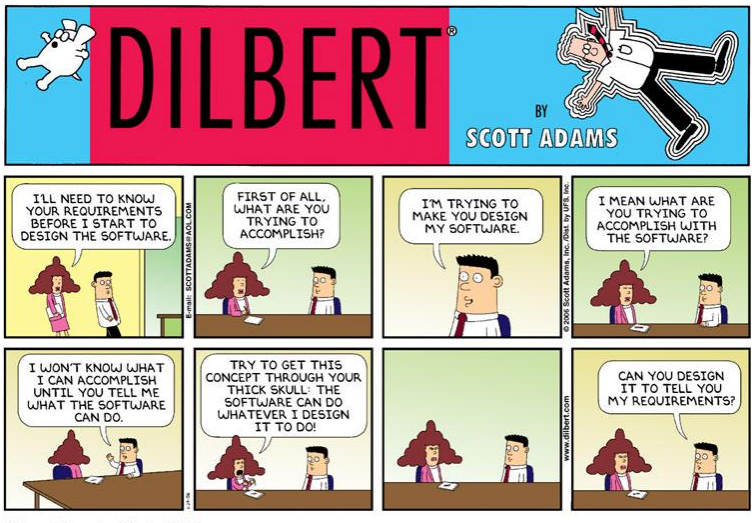
- **Inception**
  - Start the process (business need, market opportunity, great idea, ...), business case, feasibility study, system scope, risks, etc.
- **Requirements elicitation**
  - Requirements discovered through consultation with stakeholders
- **Requirements analysis and negotiation**
  - Requirements are analyzed and conflicts resolved through negotiation
- **Requirements specification**
  - A precise requirements document is produced
- **Requirements validation**
  - The requirements document is checked for consistency and completeness
- **Requirements management**
  - Needs and contexts evolve, and so do requirements!

4

General Problems with the Requirements Process

- Lack of the right expertise (software engineers, domain experts, etc.)
- Initial ideas are often incomplete, wildly optimistic, and firmly entrenched in the minds of the people leading the acquisition process
- Difficulty of using complex tools and diverse methods associated with requirements gathering may negate the anticipated benefits of a complete and detailed approach

5



6

Statistics from NIST Report

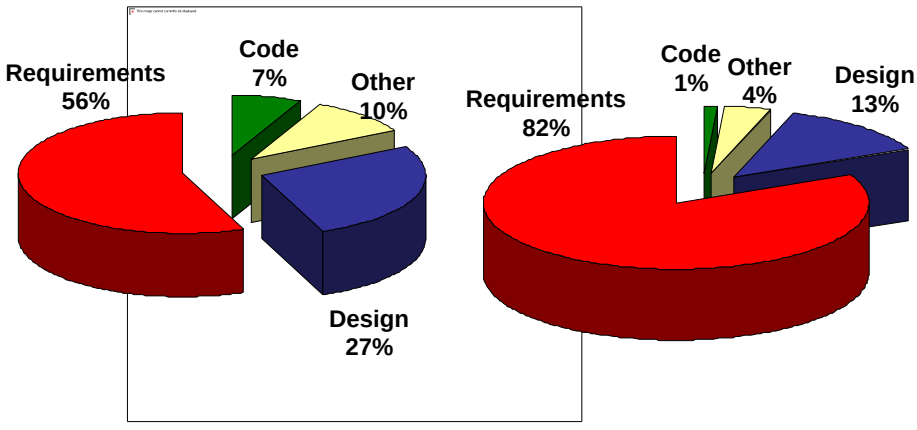
- NIST (National Institute of Standards and Technology) has published a comprehensive (309 pages) and very interesting report on project statistics and experiences based on data from a large number of software projects[1]
  - 70% of the defects are introduced in the specification phase
  - 30% are introduced later in the technical solution process
  - Only 5% of the specification inadequacies are corrected in the specification phase
  - 95% are detected later in the project or after delivery where the cost for correction on average is 22 times higher compared to a correction directly during the specification effort
  - The NIST report concludes that extensive testing is essential, however testing detects the dominating specification errors late in the process

[1] http://www.nist.gov/public_affairs/releases/n02-10.htm (May 2002)

7

Why Focus on Requirements ?

- Distribution of Defects
- Distribution of Effort to Fix Defects



Source: Martin & Leffinwell

8

CHAOS Report (2004)[1]

RESOLUTION OF PROJECTS

This year's results show that 29% of all projects succeeded (delivered on time, on budget, with required features and functions); 53% are challenged (late, over budget and/or with less than the required features and functions); and 18% have failed (cancelled prior to completion or delivered and never used), as shown in Figure 2.0.

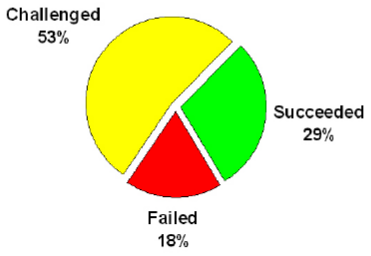
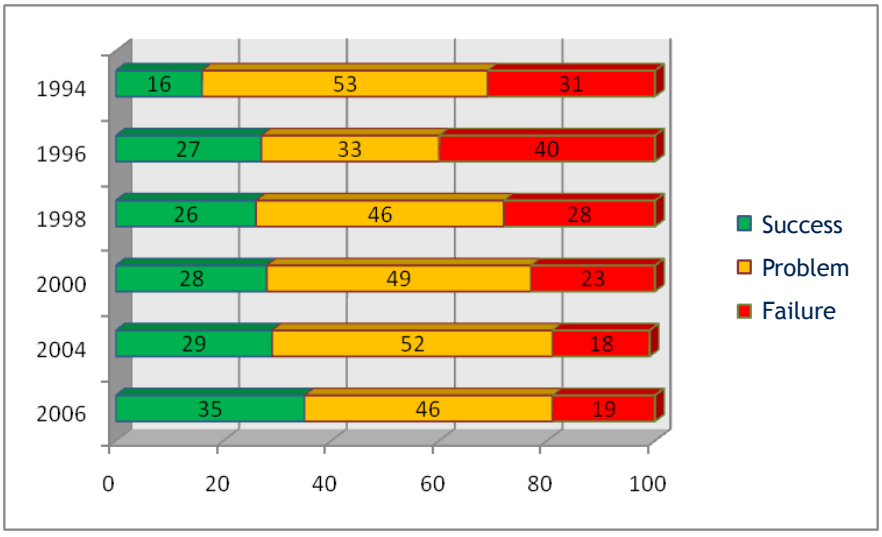


Figure 2.0

[1] Standish Group Inc., 2004

9

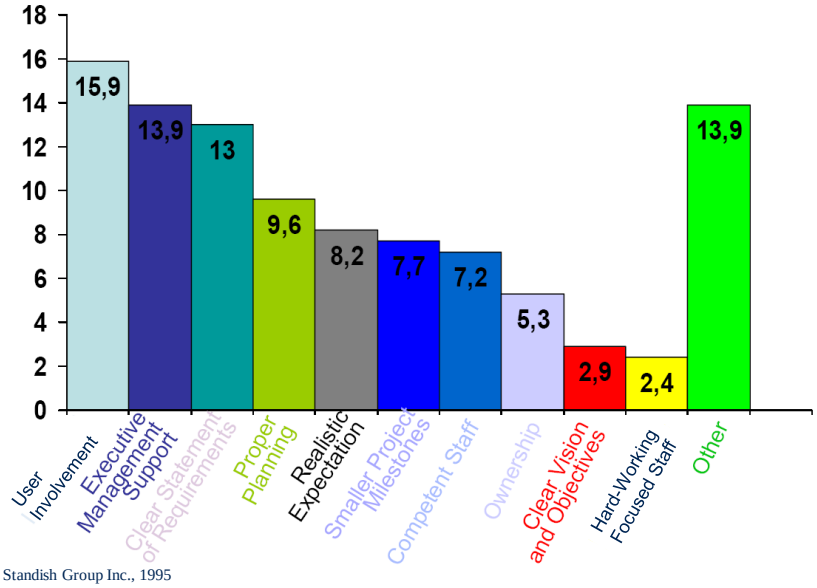
Progression since 1994



Source: Standish Group Inc., 1994-2006

10

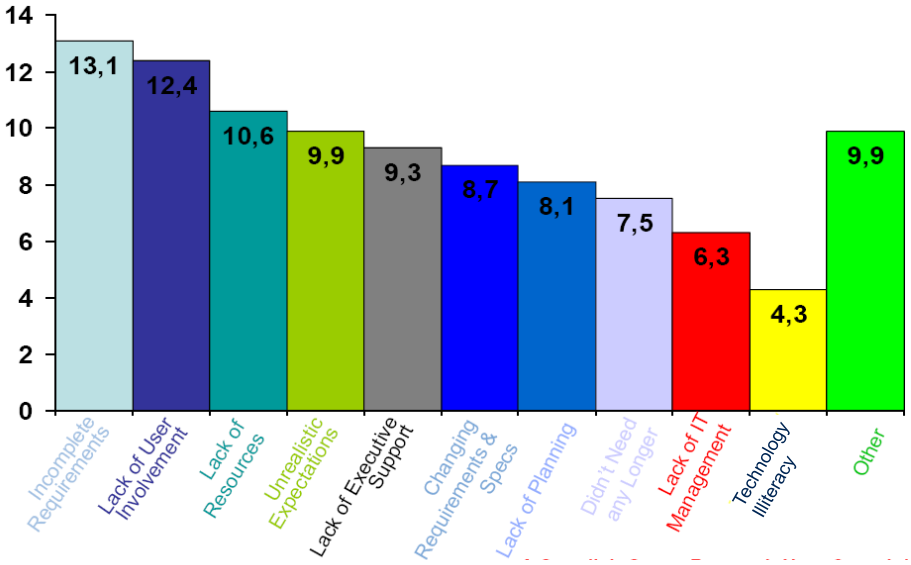
Success Factors



Source: Standish Group Inc., 1995

11

Problem Causes



Source: Standish Group Inc., 1995

12

Evolution of Success Factors

CHAOS			
1994	1996	1999	2000
User involvement	User involvement	User involvement	Executive management support
Executive management support	Executive management support	Executive management support	User involvement
Clear statement of requirements	Clear statement of requirements	Clear statement of requirements	Experienced project manager
Proper planning	Firm basic requirements	Experienced project manager	Clear business objectives
Realistic expectations	Competent staff	Small project milestones	Minimized scope
Small project milestones	Small project milestones	Firm basic requirements	Standard software infrastructure
Competent staff	Experienced project manager	Competent staff	Firm basic requirements
Ownership	Proper planning	Proper planning	Formal methodology
Clear vision and objectives	Ownership	Ownership	Reliable estimates
Hard-working, focused staff	Other	Other	Other

Involvement/ Ownership | Requirements | Management Support | Planning | Staff

Source: Standish Group Inc., 2000

13

Managing Evolving Requirements

"Changing requirements is as certain as death and taxes"

Requirement tools: These seem to have the biggest impact on the success of a project. This may seem strange since "Firm Basic Requirements" is number six on the top ten list. However these tools, if used as a platform for communications between all the stakeholders, such as executive sponsors and users, can provide enormous benefits. This tool needs to be at the top of the shopping list for any firm involved in developing software applications.

Source: http://standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf, 1999

14

Types of Requirements

So Many "Requirements"… (1)

- A goal is an objective or concern that guides the RE process. It can be used to discover and evaluate functional and non-functional requirements
  - A goal is not yet a requirement…
- Note: All requirements must be verifiable (by some test, inspection, audit etc.)
- A functional requirement is a requirement defining functions of the system under development
  - Describes what the system should do
- A non-functional requirement is a requirement that is not functional. This includes many different kinds of requirements. – Therefore one often considers the following sub-categories:

16

Different types of non-functional requirements

- Performance requirements, characterizing system properties such as expected performance, capacity, reliability, robustness, usability, etc.
- Design constraints (also called process requirements), providing constraints on how the system should be designed and built – related to development process, documentation, programming language, maintainability, etc.
- Commercial constraints, such as development time frame and costs.

17