

Chapter 1

The Requirements Problem

- The goal of software development
- Reasons of projects successes and failures
- The cost of requirements errors

The Goal of Software Development

To develop high-quality softwares
on time and on budget
that meets customers' real needs.

The Standish Group's Survey (1994)

The following is a look at the study done by the Standish group:

- In **USA \$250 billions** spent **each year** on IT application development of approximately **175,000 projects**.
- **31% of projects** will be **cancelled** before they ever get completed.
- **52.7% of projects** will **cost 189%** of their original estimates.
- \$81 billions for vaporware
- \$59 billions complete but late projects

Causes of Project Success and Failure

The Standish Group survey also asked respondents to identify the most significant factors that contributed to projects that were rated:

- **Success**
- **Challenged**: late and didn't meet expectations
- **Impaired**: canceled

Factors that Caused Projects to be “Challenged”

- The 3 most commonly cited factors were:
 1. Lack of user input: 13% of all proj.s
 2. Incomplete req.s & spec.s: 12% of all proj.s
 3. Changing req.s and spec.s: 12% of all proj.s
- At least 1/3 of the development projects run into trouble for reasons that are directly related to
 - requirements gathering,
 - requirements documentation,
 - requirements management.

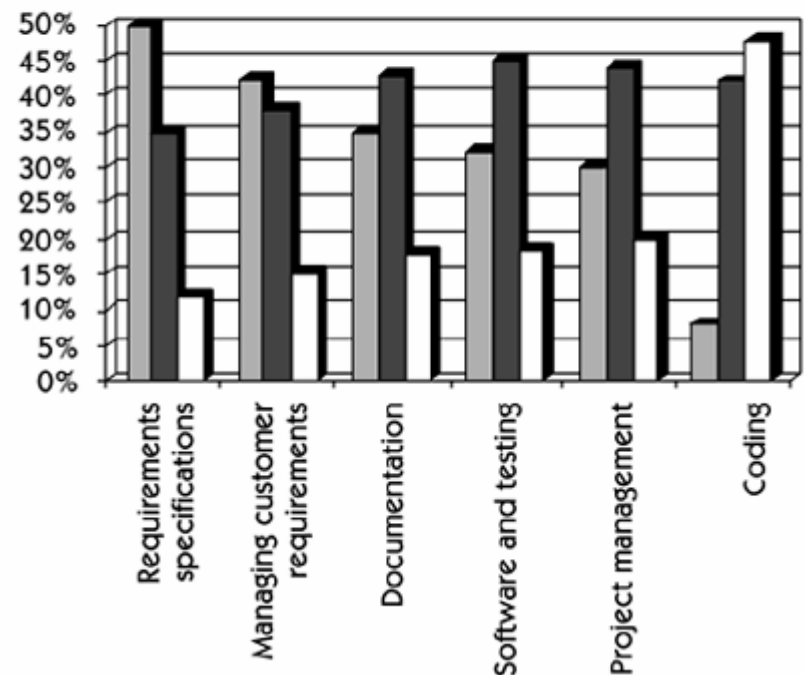
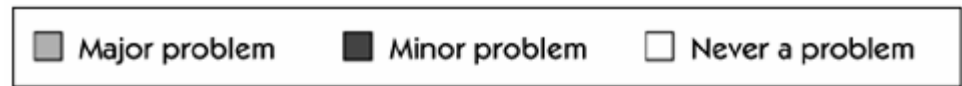
Factors that Caused Projects to be “Success”

- The 3 most important success factors were:
 1. User involvement: 16% of all successful projects
 2. Executive management support: 14% of all successful projects
 3. Clear statement of requirements: 12% of all successful projects

Software Development Problems

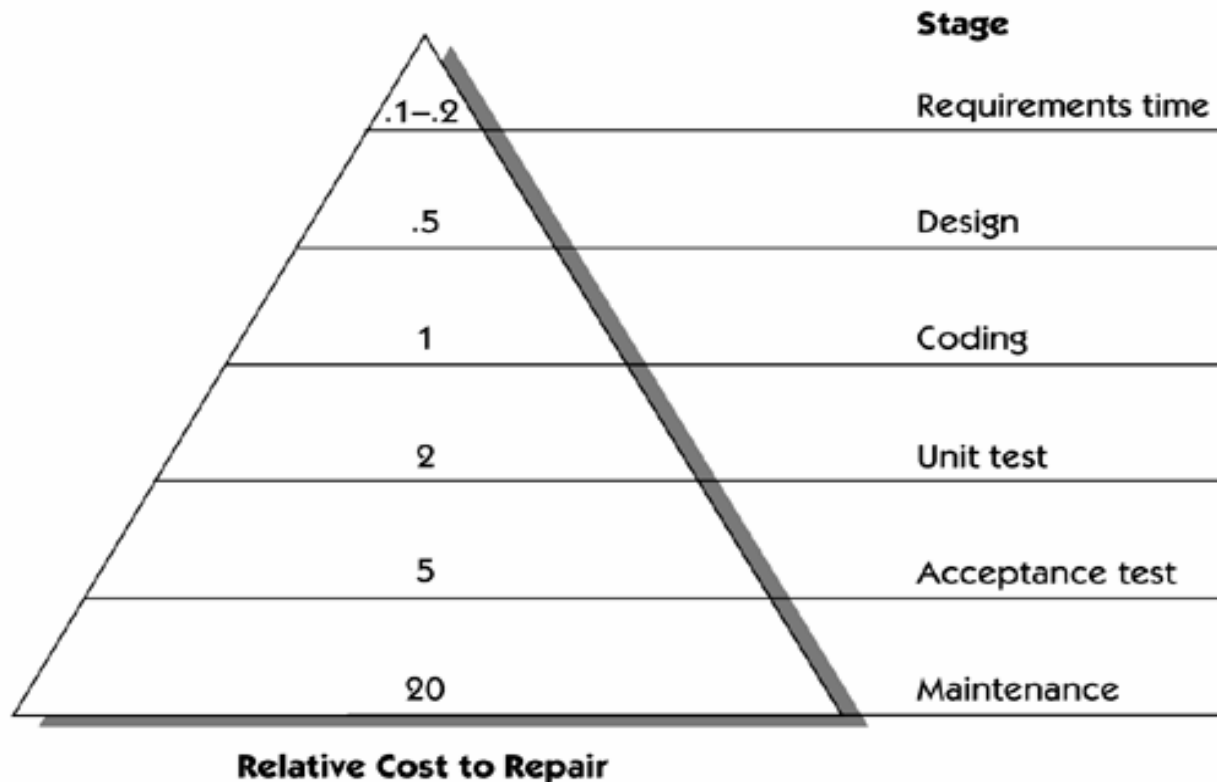
Survey conducted by European Software Process Improvement Training Initiative (ESPITI) [1995], with 3,800 responses.

- The two largest problems, appearing in about half of the responses, were
 1. Requirements specifications
 2. Managing custom requirements



The High Cost of Requirements Errors

Figure 1-2. Relative cost to repair a defect at different lifecycle phases. (Data derived from [Davis \[1993\]](#).)



The High Cost of Requirements Errors

- The errors discovered during the design of a development project could fall into one of two categories:
 1. Errors that occurred **when** the development staff created a technical **design from a correct set of requirements**, or
 2. Errors that should have been detected as **requirements errors** somewhat earlier in the process but that somehow "**leaked**" into the **design phase** of the project.
- It's the **second category** of errors that turn out to be particularly **expensive** .. **Why?**

Because ...

1. **The errors are misleading.** Everyone is looking for design errors during the testing or inspection activities while in fact they are in the requirements.
2. By the time the requirements error is discovered, **time and effort have been lost** in faulty design. So, **the design have to be thrown away or reworked.**

The High Cost of Requirements Errors

- In order to repair a defect, we are likely to experience costs in some or all of the following areas:
 - Respecification, Redesign, Recoding, Retesting,
 - Change orders: replacing defected systems by corrected one,
 - Corrective action: undoing whatever damage may have been done and refund.
 - Scrap: useless code, design and test cases.
 - Recall of defective software (could be embedded)
 - Warranty costs.
 - Product liability: customer can sue for damages
 - Service costs for reinstallation.
 - Documentation

Key points

- The goal of software development is to develop quality software – on time and on budget – that meets customers' real needs.
- Project success *depends on* effective requirements management.
- Requirements errors are the most common type of systems development error and the most costly to fix (25% - 40\$ of budget).
- A few key skills can significantly reduce requirements errors and thus improve software quality.