

---

# Chapter 1: The Requirements Problem

---

# Objectives

---

- To define the goal of software development
- To describe the root causes of software project success and failure
- To illustrate the high cost of requirements errors

# The Goal of Software Development

---

- To develop quality software – on time and on budget – that meets customers' real needs.

# A Look at the Data

---

- A study by [the Standish Group \[1994\]](#) reported:
  - In the United States, \$250 billion spent each year on IT application development of approximately 175,000 projects.
  - 31% of projects will be canceled before they ever get completed.
  - 52.7% of projects will cost 189% of their original estimates.

# The Root 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 did not meet expectations)
  - “Impaired” (canceled)

# Factors that Caused Projects to be "Challenged"

---

- The 1994 Standish Group study noted the three most commonly cited factors that caused projects to be "challenged":
  1. Lack of user input: 13 percent of all projects
  2. Incomplete requirements and specifications: 12 percent of all projects
  3. Changing requirements and specifications: 12 percent of all projects
- It appears that at least a third of development projects run into trouble for reasons that are directly related to requirements gathering, requirements documentation, and requirements management.

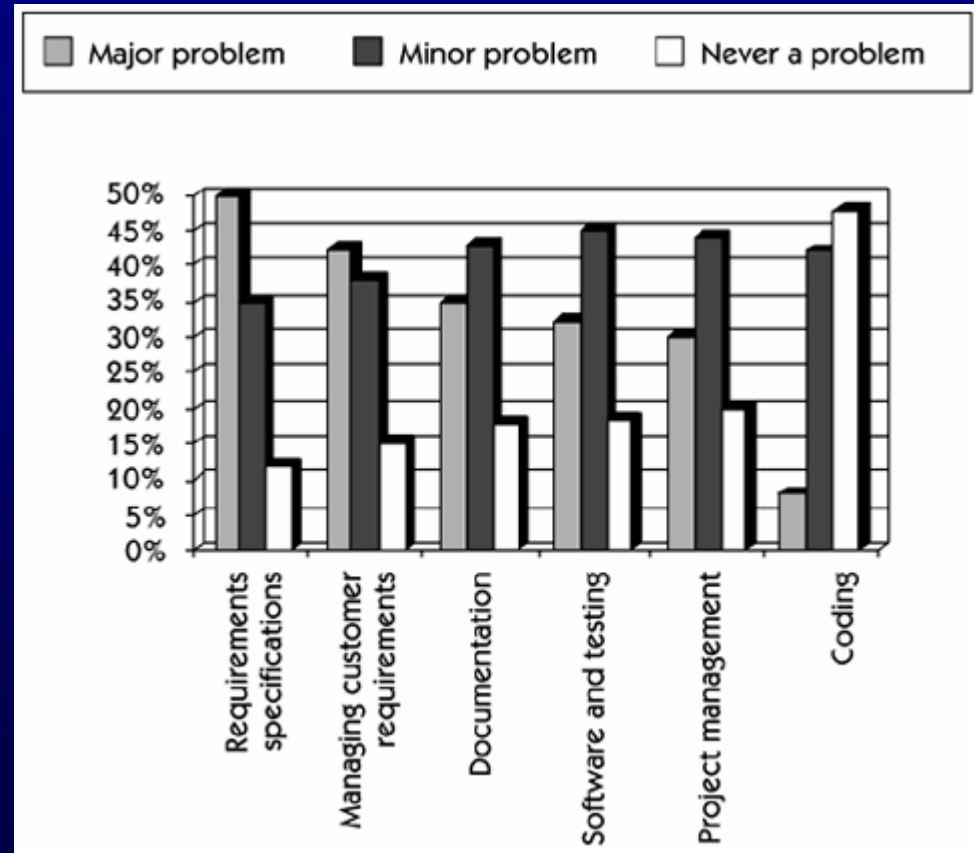
# Factors that Caused Projects to be "Success"

---

- According to the Standish study, the three most important success factors were
  1. User involvement: 16 percent of all successful projects
  2. Executive management support: 14 percent of all successful projects
  3. Clear statement of requirements: 12 percent of all successful projects

# Largest Software Development Problems by Category

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

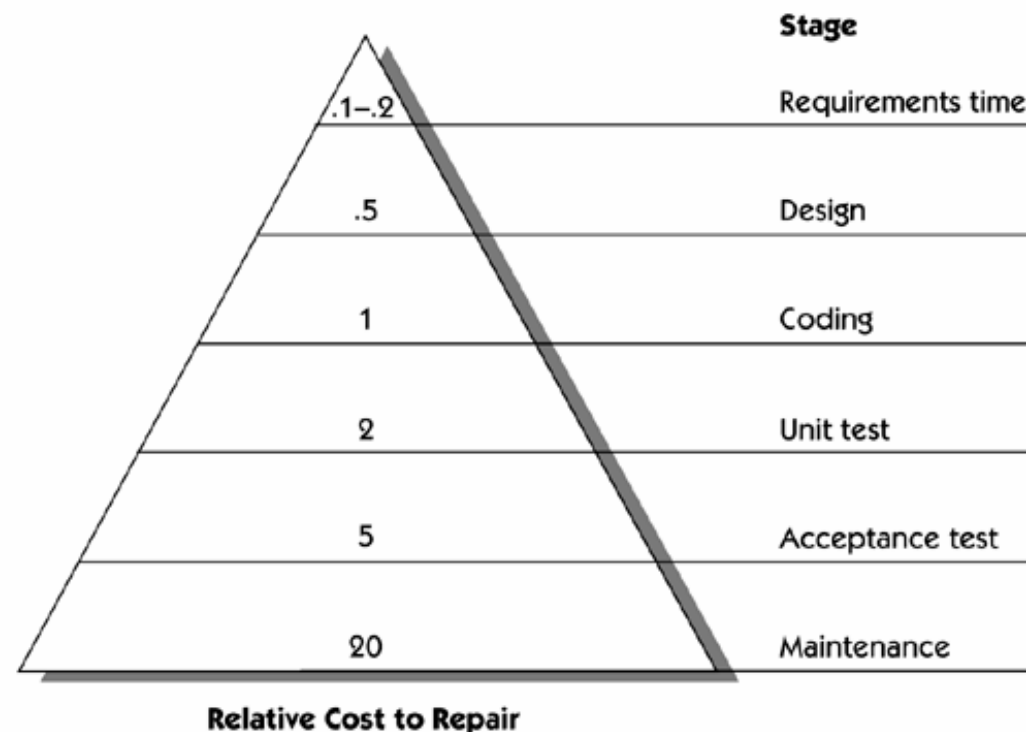


Survey conducted by [European Software Process Improvement Training Initiative \(ESPITI\) \[1995\]](#), 3,800 responses



# 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 *(Cont'd)*

---

- 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, for two reasons (see the next slide).

# The High Cost of Requirements Errors *(Cont'd)*

---

1. By the time the requirements-oriented error is discovered, the development group will have invested time and effort in building a design from those erroneous requirements. As a result, the design will probably have to be thrown away or reworked.
2. The true nature of the error may be disguised; everyone assumes that they're looking for design errors during the testing or inspection activities that take place during this phase, and considerable time and effort may be wasted until someone says, "Wait a minute! This isn't a design mistake after all; we've got the wrong requirements."

# The High Cost of Requirements Errors *(Cont'd)*

- In order to repair a defect, we are likely to experience costs in some or all of the following areas:
  - Re-specification.
  - Redesign.
  - Recoding.
  - Retesting.
  - Change orders.
  - Corrective action—undoing whatever damage may have been done.
  - Scrap— code and design that are based on incorrect requirements.
  - Recall of defective versions of software
  - Warranty costs.
  - Product liability—if the customer sues for damages caused by the defective software.
  - Service costs for a company representative to visit a customer's field location to reinstall the new software.

# Improved Project Performance

- The Standish Group's CHAOS studies show improvements in IT projects in the past decade.\*

Measure	1994 Data	2002 Data	Result
Successful projects	16%	34%	Doubled
Failed projects	31%	15%	Halved
Money wasted on challenged and failed projects	\$140 B out of \$250 B	\$55 B out of \$255 B	More than halved

*\*The Standish Group, "Latest Standish Group CHAOS Report Shows Project Success Rates Have Improved by 50%" (March 25, 2003).*

# 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.
- A few key skills can significantly reduce requirements errors and thus improve software quality.