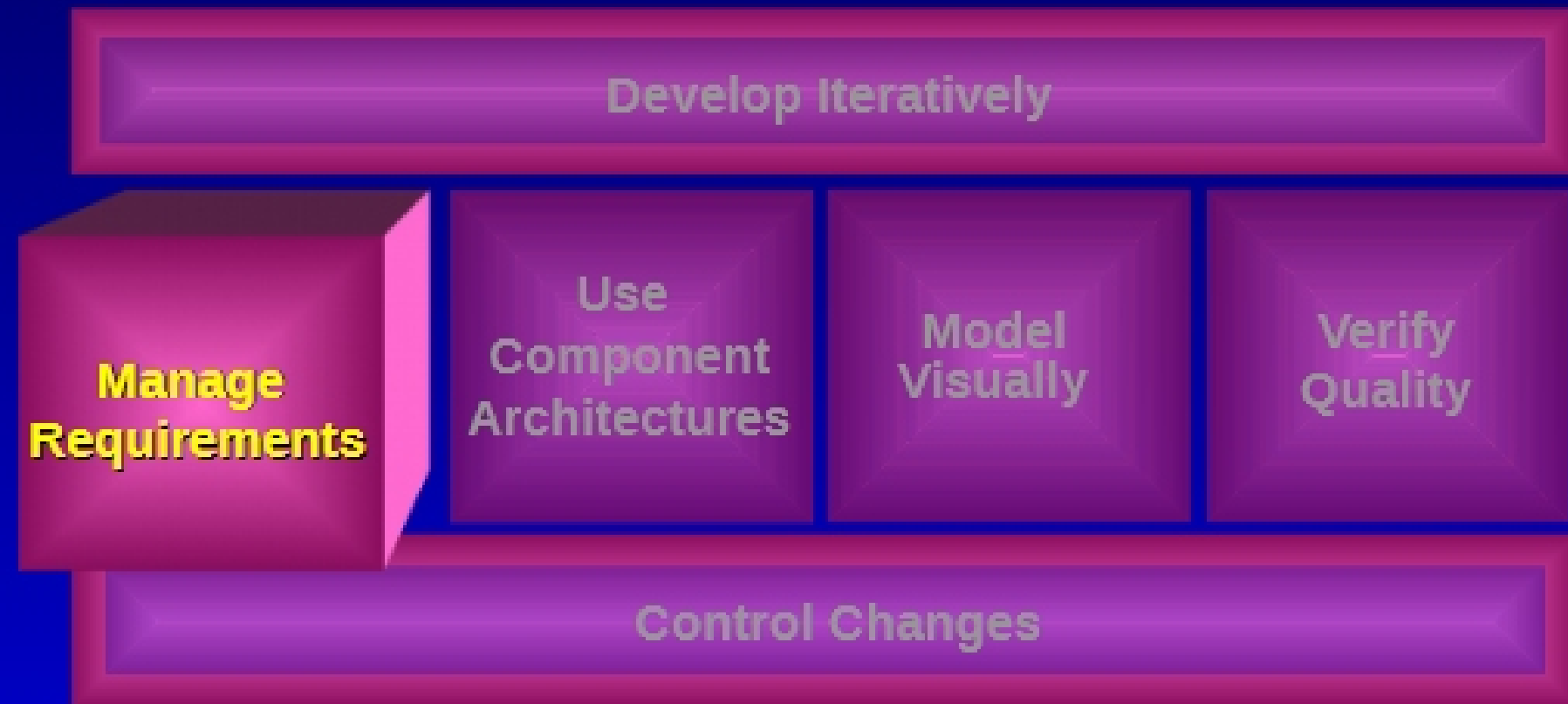


# Practice 2: Manage Requirements



A report by Standish Group cites **forty percent of software projects fail.**

**This failure is attributed to:**

**poor requirements management**; ((eliciting) capturing, (documenting) modeling, verification; prototyping; agreeing w/customer on how changes will be handled, ...)

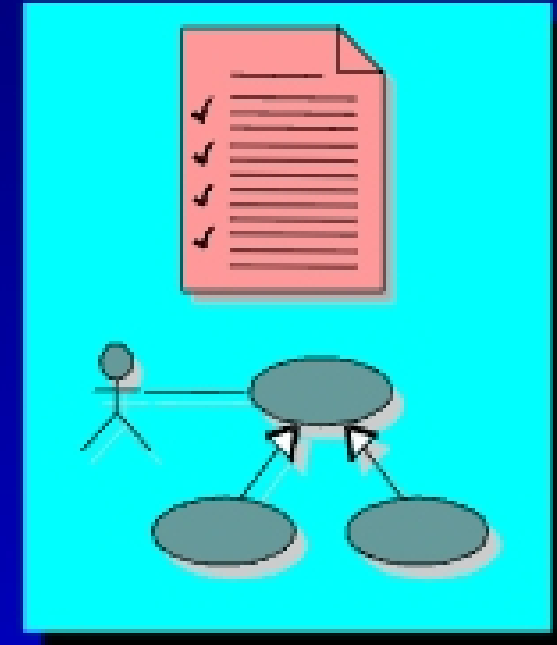
**incorrect definition of requirements** from the start of the project; and (feedback, verification...)

**poor requirements management** throughout the development lifecycle.

**(handling Change!!!)** (Source: Chaos Report, <http://www.standishgroup.com>).

# Practice 2: Manage Requirements

- ◆ Elicit, organize, and document required functionality and constraints
- ◆ Evaluate changes and determine their impact
- ◆ Track and document tradeoffs and decisions
- ◆ Getting comprehensive system requirements is a continuous process!



Obtaining a complete, exhaustive set of requirements prior to development is impossible!

***Requirements are dynamic --***

***Expect them to change during software development***

***Change cannot be stopped, but it can be managed!!***

# Definitions: Requirements and Their Management

---

- ◆ A **requirement** is a condition or capability to which the system must conform
- ◆ **Requirements management** is a systematic approach to
  - Eliciting, organizing, and documenting the requirements of the system, and
  - Establishing and maintaining agreement between the customer/user and the project team on the changing requirements of the system
- ◆ Requirements specify 'what' the system must do - not how!
- ◆ Requirements Management will be successful only if it allows for uncertainty early in development
- ◆ Requirements Management must ensure requirements coverage over time.