

A decorative L-shaped line in a gold color, consisting of a horizontal segment and a vertical segment meeting at a right angle.

# Chapter 1: Introduction

---

Software Development Project  
Getting Started

---

# Objectives

In this chapter, you will learn about:

- The importance of requirements
- The role of RE in Software Development Lifecycle

# Problem Statement

- **What** is the problem that you are trying to solve?

“The hardest single part of building a system is deciding **what** to build.....No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later”

F. P. Brooks

---

# Typical Project Requirements

- Requirements form the basis for:
  - Project Planning
    - Remember: **P-P-P-P-P**
  - Risk Management
  - Acceptance Testing
  - Change Control

# SW Engineering Projects

## Overview

- Software Development usually involves the following stages:

### STAGES

- ❑ Requirements Analysis & Design
- ❑ Systems Design
- ❑ Program Design
- ❑ Writing the Program (coding)
- ❑ Unit Testing
- ❑ Integration Testing
- ❑ System Testing
- ❑ System Delivery
- ❑ Maintenance

### OWNER

Requirements/Systems Engineer

Systems Architect

Programmer

Programmer

Programmer

Systems Tester/Verification

Systems Tester/Verification

Trainer

Trainer & Customer Support

We will talk about the relationships between the **RE** and other SW team members shortly

---

# SW Engineering Projects

## Common Issues

- Most project affecting and critical issues are in the Requirements arena
- Major gaps in Requirements tend to be critical to a project success
- Concise, on-time requirements are a necessary foundation for a successful architecture
  - Quality before Design
- Producing a good set of requirements is likely the most difficult job in software systems development

---

# SW Engineering Projects

## Recurring Problems

- Lack of Functional Requirements
  - ❑ No Requirements have been written
  - ❑ Usage Scenario Not Understood and Documented
  - ❑ Functionality of the System Incomplete or Underestimated
  - ❑ Customer unknown and not contacted
  - ❑ No Acceptance Criteria for the System

---

# SW Engineering Projects

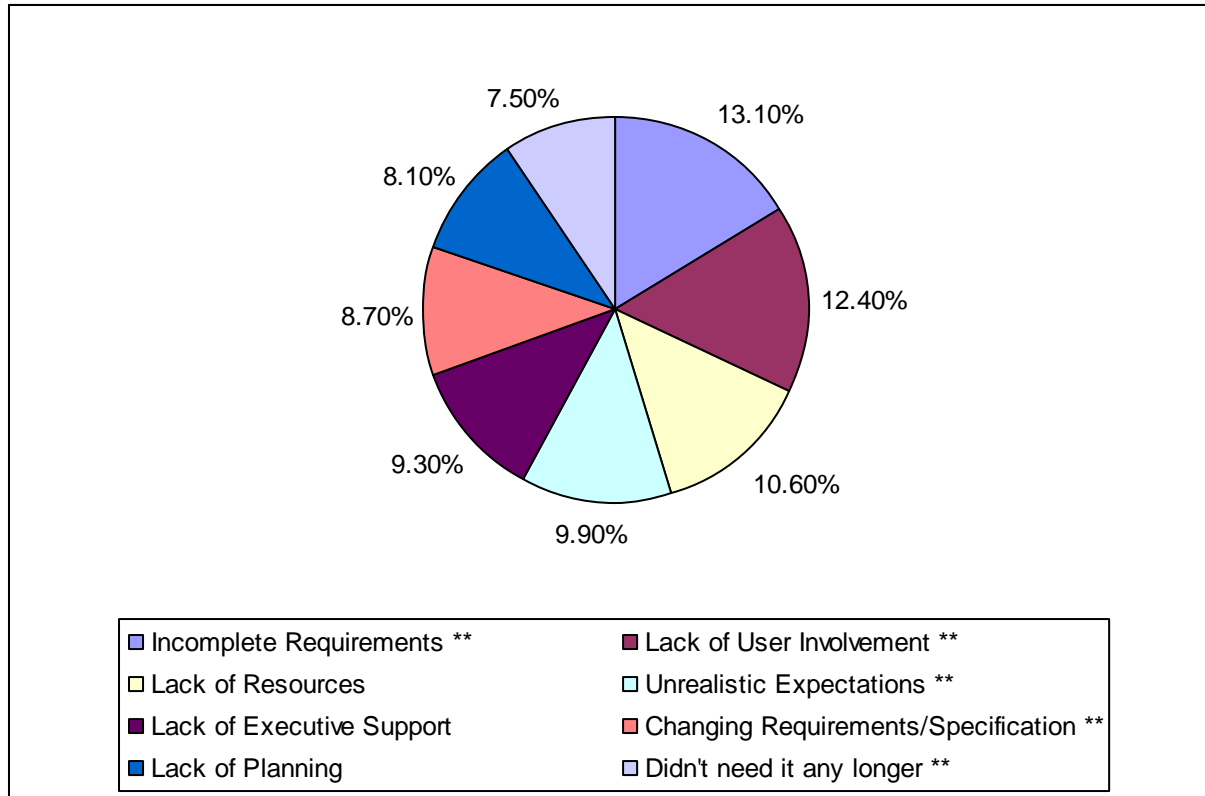
## Recurring Problems

- Lack of Performance and capacity Requirements
  - Number and/or Types of Users Undocumented
  - Transaction and data Volumes Unknown
- Lack of OA&M Requirements
  - No OA&M Requirements Documented
  - No Availability Requirements Documented
    - Availability not Tied to Customer Need
      - “Want 7 x24”, but no business need exists



# SW Engineering

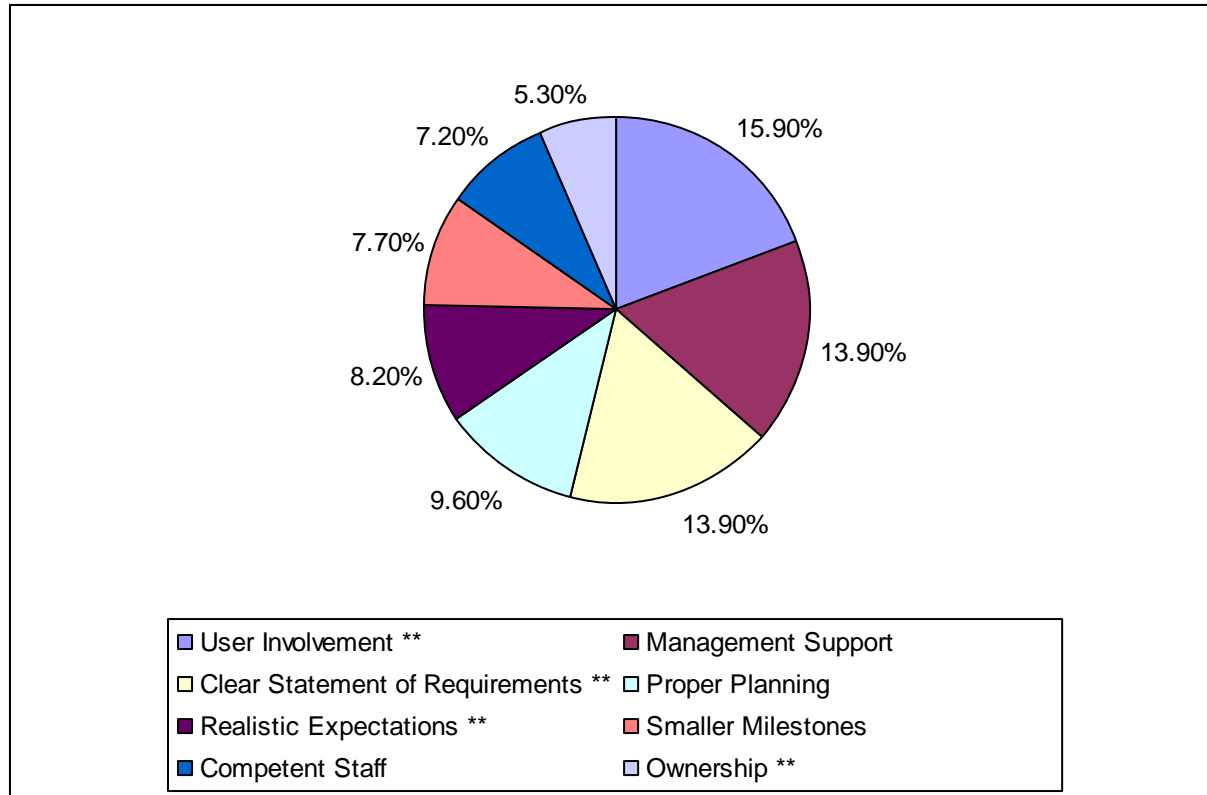
## Reasons for Project Failure



Factors directly related to requirements are identified with \*\*

# SW Engineering

## Project Success Factors



Factors directly related to requirements are identified with \*\*

# Software Engineering Project

## What!

- Good Requirements describe what the system is supposed to do, NOT *how* the system is supposed to do it
- Remember the 5 W's (and an H)
  - Who → Customer
  - **WHAT** → Requirements
  - Where → *Facilities Plan*
  - When → Project Plan
  - Why → Problem Statement
  - How → Architecture & Design

# Requirements are NOT

- Requirements are NOT a description of *how* the system provides the needed functionality.
- Requirements should NOT specify technology or implementation except where those items are customer constraints
  - Implied or hidden
    - “Do such & so?” Such & So must be explicitly stated
  - Invalidated Assumptions
  - Over extended Assumptions
    - Do not assume that requirements for *this* system or *this* customer are the same as they were for *that* system or *that* customer
  - Indecisive
    - Words like “would”, “could”, “should”, “most of the time”, “flexible” do not belong in requirements document. Remember an item *will* or *will not* be in the system
  - Inconsistent or Conflicting

# What Requirements ARE

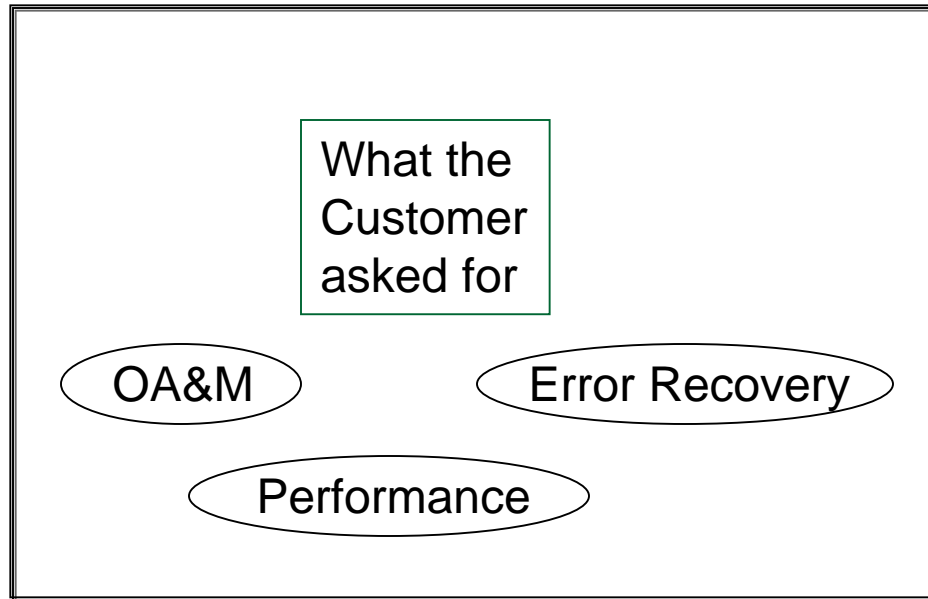
- Description of **WHAT** the entire system is supposed to do:
  - ❑ Reflective of customer need
  - ❑ Constraints imposed by the customer
  - ❑ Constraints imposed by business and marketing needs
  - ❑ Unambiguous (clear and concise)
  - ❑ Complete
  - ❑ Prioritized
  - ❑ Traceable
  - ❑ Implementable within project constraints (e.g., schedule, budget, etc.)
  - ❑ Formally accepted by the customer, systems engineering and development, system test, and under change control
- Requirements are sometimes referenced for contracts (contractual documents)
- Sometimes considered to be contract between systems engineering and development

# Problem Statement

What the  
Customer  
Asked For

# Requirements Engineering

## Solution Space for the Problem



Is customer the same as end-user?

# Customer Needs

## End-User Needs

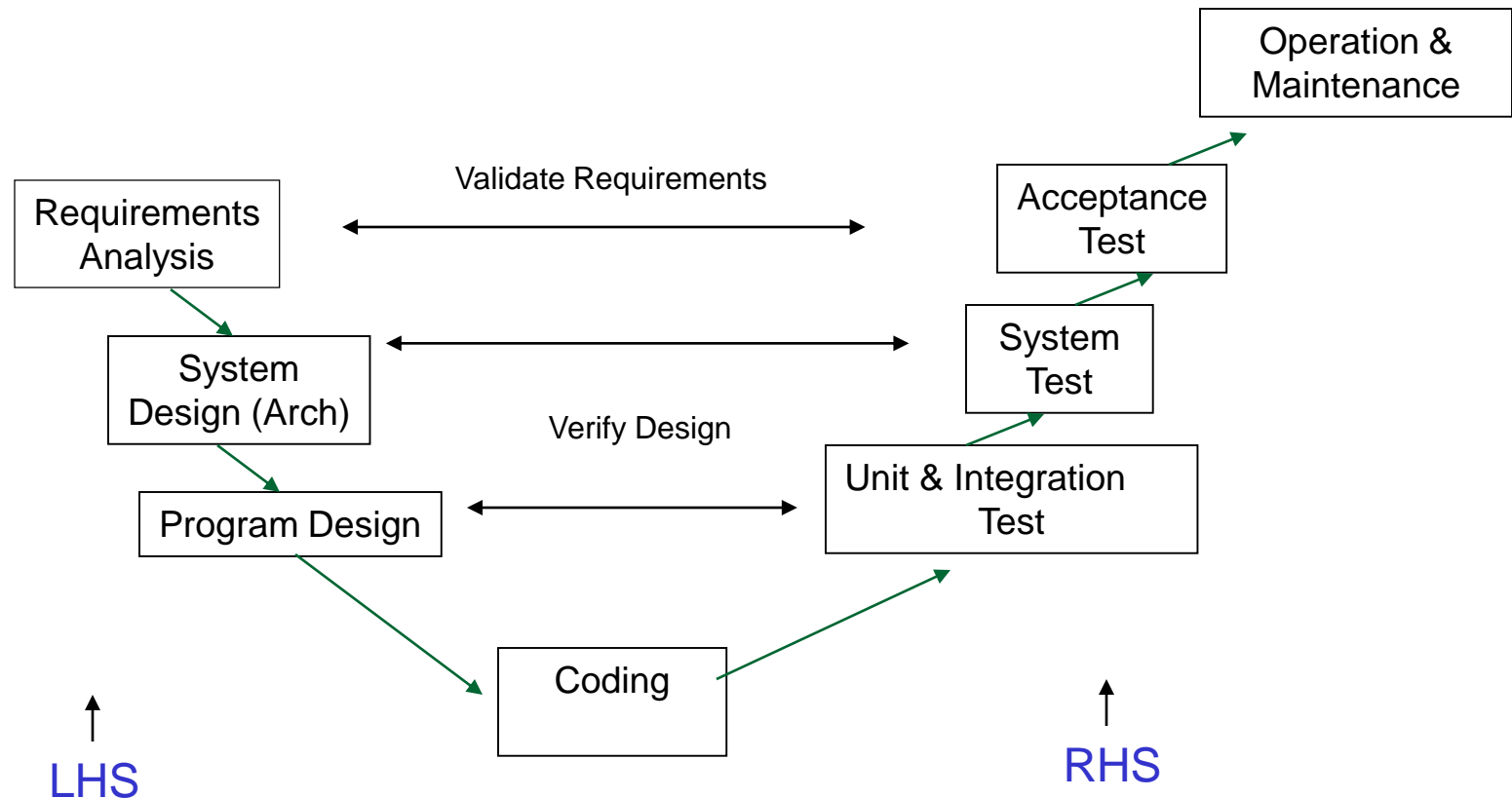
- Customer and end-user are not always the same person
  - The end-user is the person who interacts with the system to get the job done
  - The customer typically pays for the system
- Customer and End-user may have similar or conflicting objectives
- Sometimes delighting your End user delights the customer
- But who determines the fate of the system?
  - The customer because s/he pays for the job
    - Very important to get customer involved in the initial phase of RE, if at all possible



# SW Development Lifecycle

## Example Model

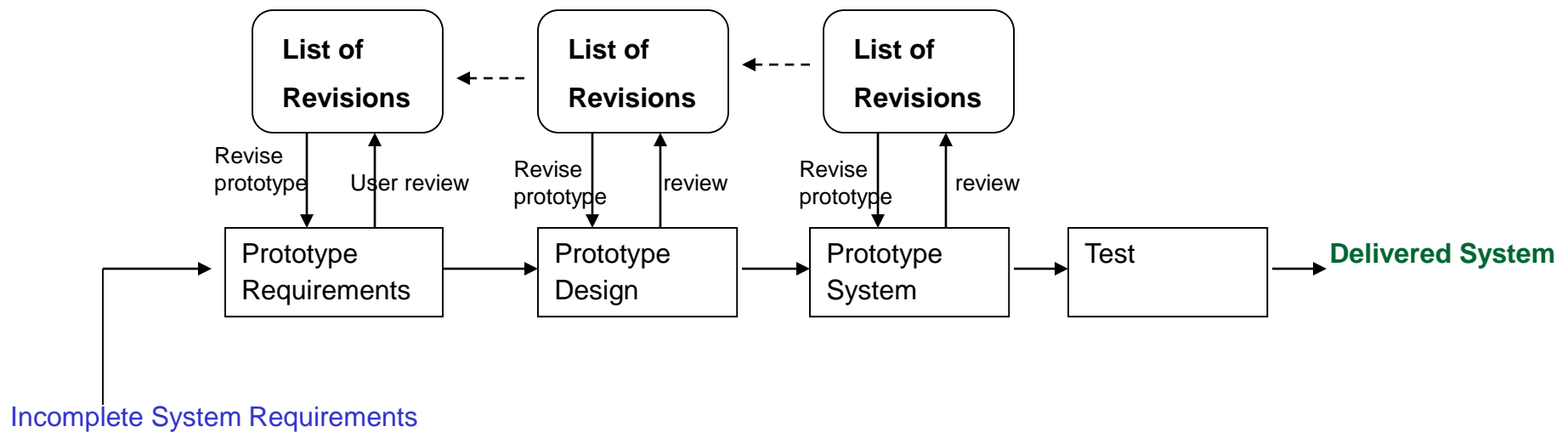
- **V Model:** If problems are found during Verification or Validation Phase, the LHS of the V is re-executed to fix the problem
  - **More explicit of re-work (unlike Waterfall Model)**



# SW Development Lifecycle

## Example Model

- **Prototyping:** Allows all or part of the system to be constructed quickly in the hopes of clarifying/understand issues
  - Iterate requirements and design to ensure common understanding

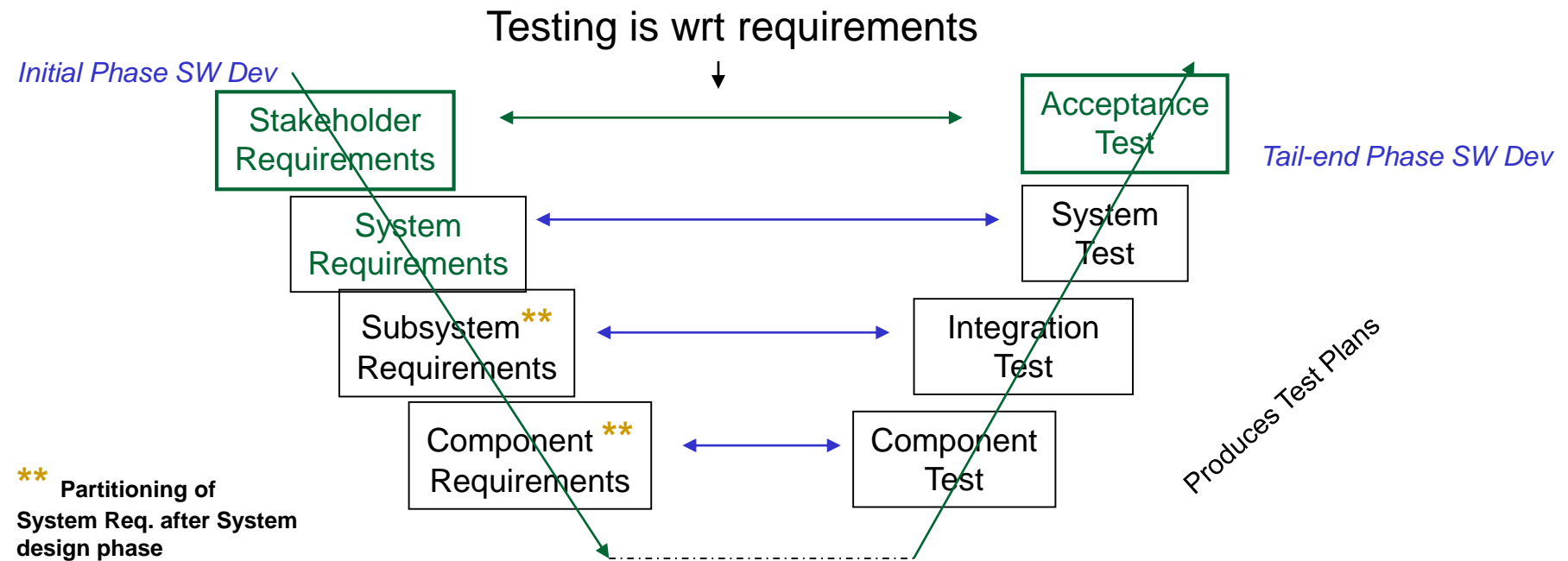


So what role does Requirements Engineering play in SW lifecycle?

# SW Lifecycle

## Role of Requirements Engineering

- Requirements influence the whole Development from Start to End:
  - **Testing is with respect to the requirements**
  - **A system is accepted against Stakeholder's Requirements during Acceptance Test Phase**

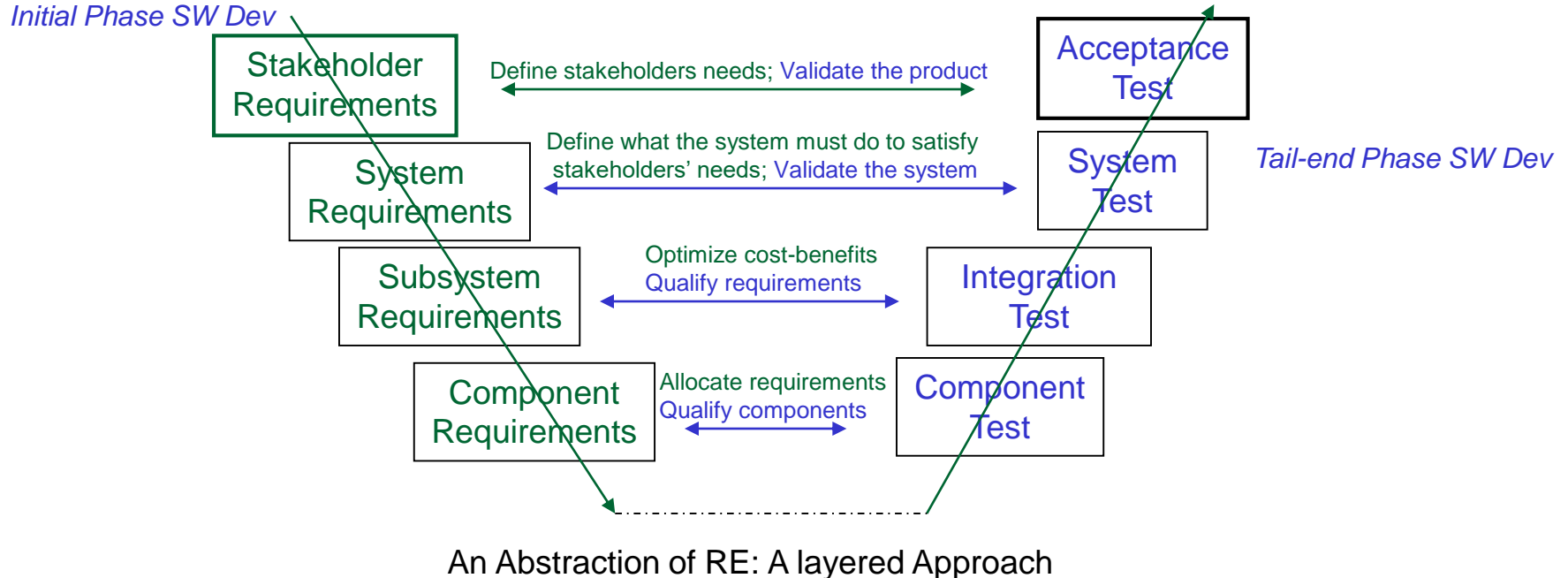


The V-Model illustrates relationships between Initial and End Phases in SW Development

# SW Lifecycle

## Role of Requirements Engineering

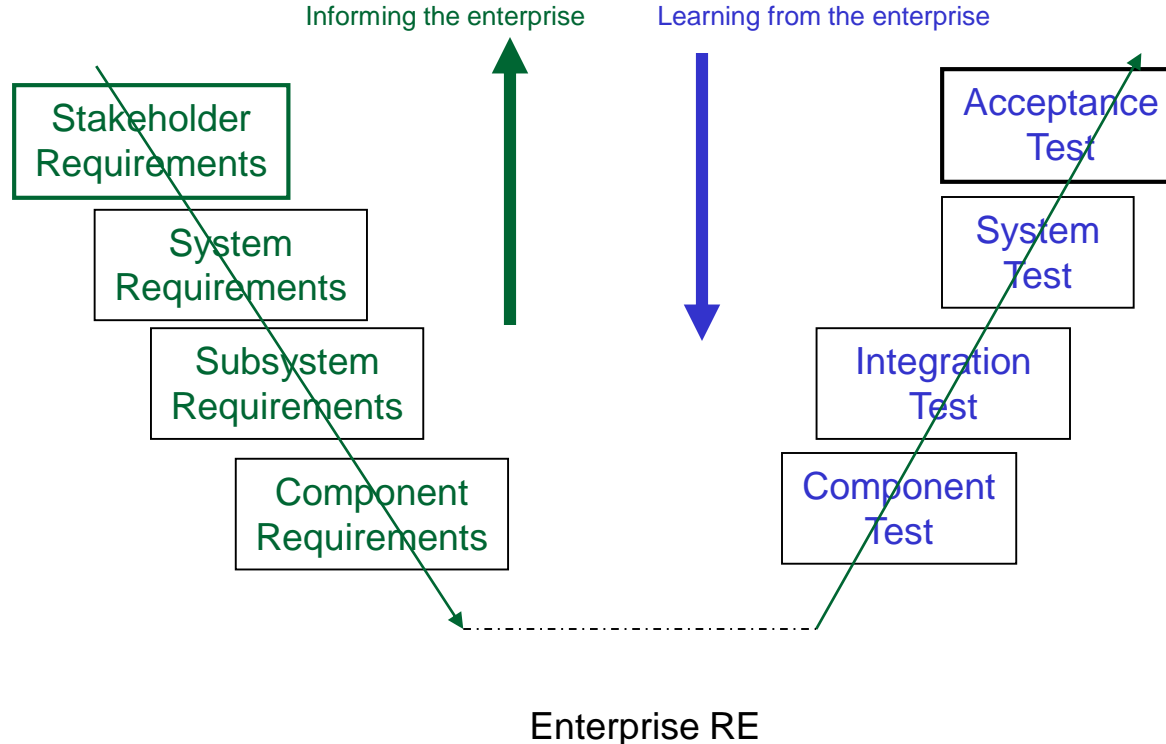
- Main Concerns of RE at each layer
  - An abstraction of the requirements engineering document



# SW Lifecycle

## Role of Requirements Engineering

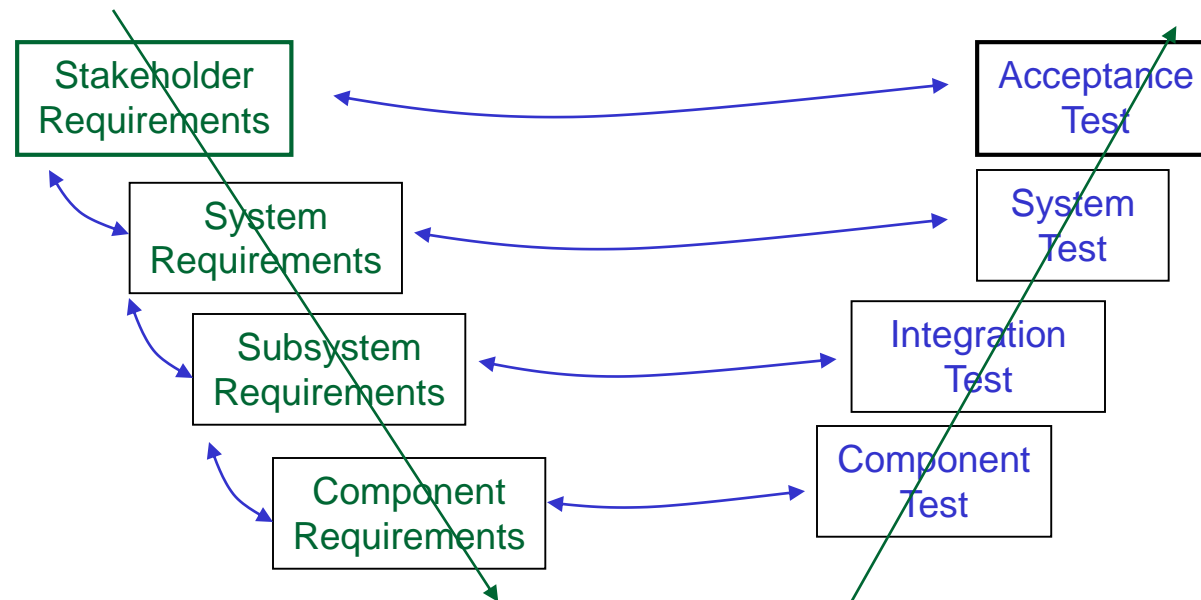
- RE provides communications amongst projects
  - Re-use of artifacts
  - Stakeholder Requirements (non-tech description) used by Management for contracts, bids, proposal etc
  - System requirements used/referenced by architects, developers in describing at a high-level their piece of the project



# SW Lifecycle

## Role of Requirements Engineering

- **Requirements Traceability:** The ability to understand how High-level requirements (goals, objectives etc) are transformed into low-level requirements (mapping between layers of information: *one-to-many usually*)
  - Stakeholders req. met by system req. → partitioned into subsystem req. → implemented as components
  - Ability to assess impact changes introduced at various phases of development sw lifecycle (change management)
  - Ability to track progress



Using traceability, track the impact of a change in System Requirements as an example

# Traceability

## Role of Requirements Engineering

- **Traceability:** Use Requirements Management Tools (e.g., **Doors**) to link requirements statements in one layer with statement in another – Drag & Drop tool
  - **Easy to answer questions like : What is the impact of making changes to one or more requirements? Who will be affected by these changes and what is the [derived] cost of implementing the change (Cost-benefit analysis)?**

