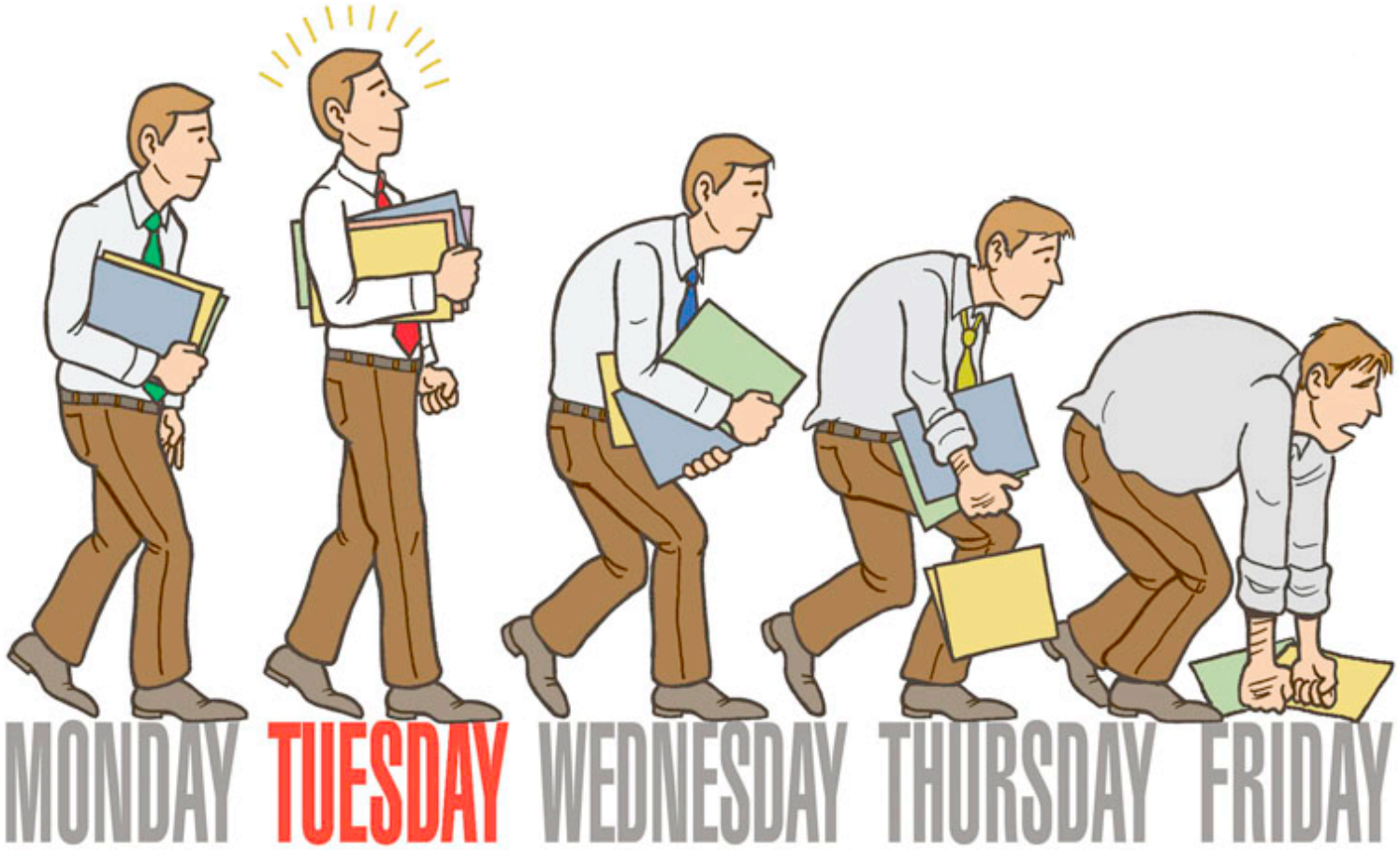
CSSE 372 Software Project Management:
Software Processes: An Introduction

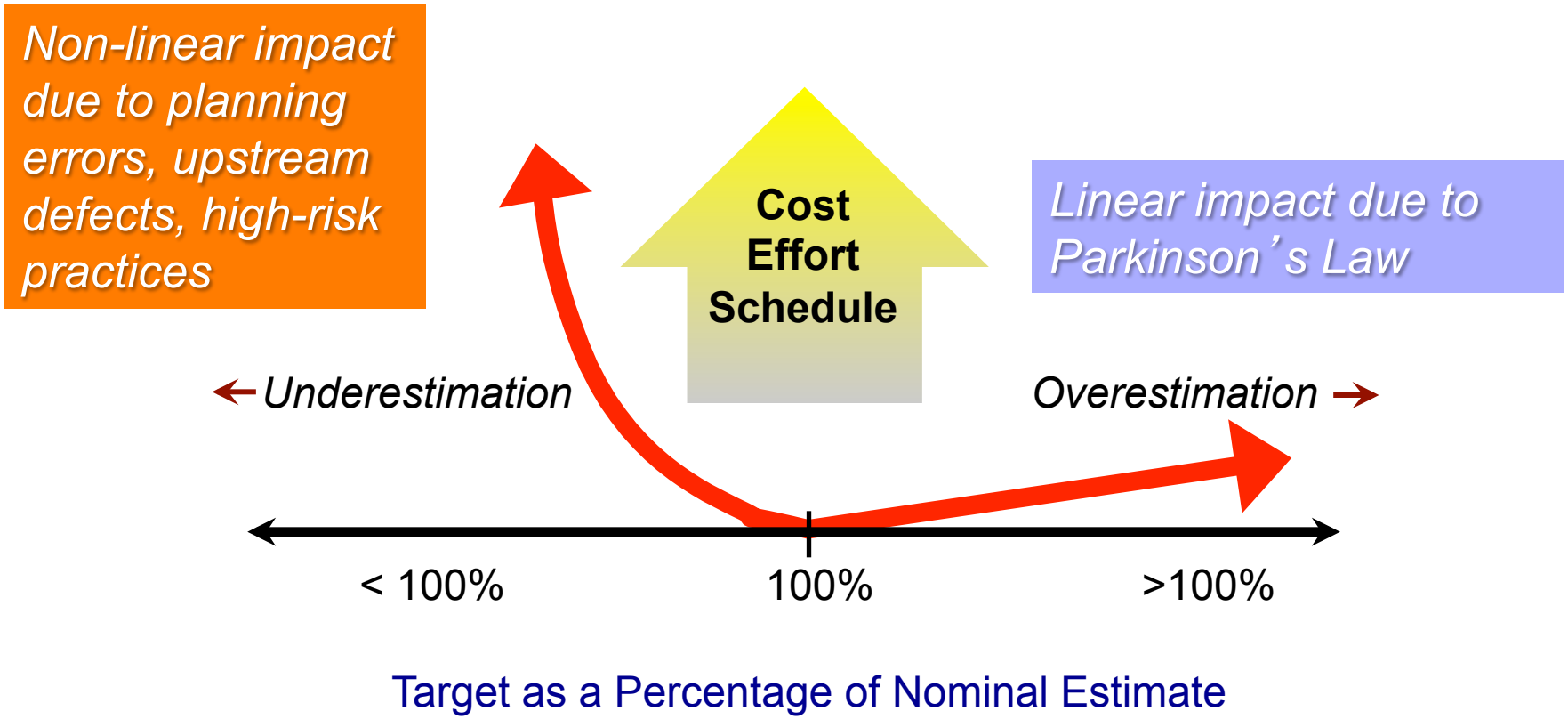
Shawn Bohner
Office: Moench Room F212
Phone: (812) 877-8685
Email: bohner@rose-hulman.edu

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

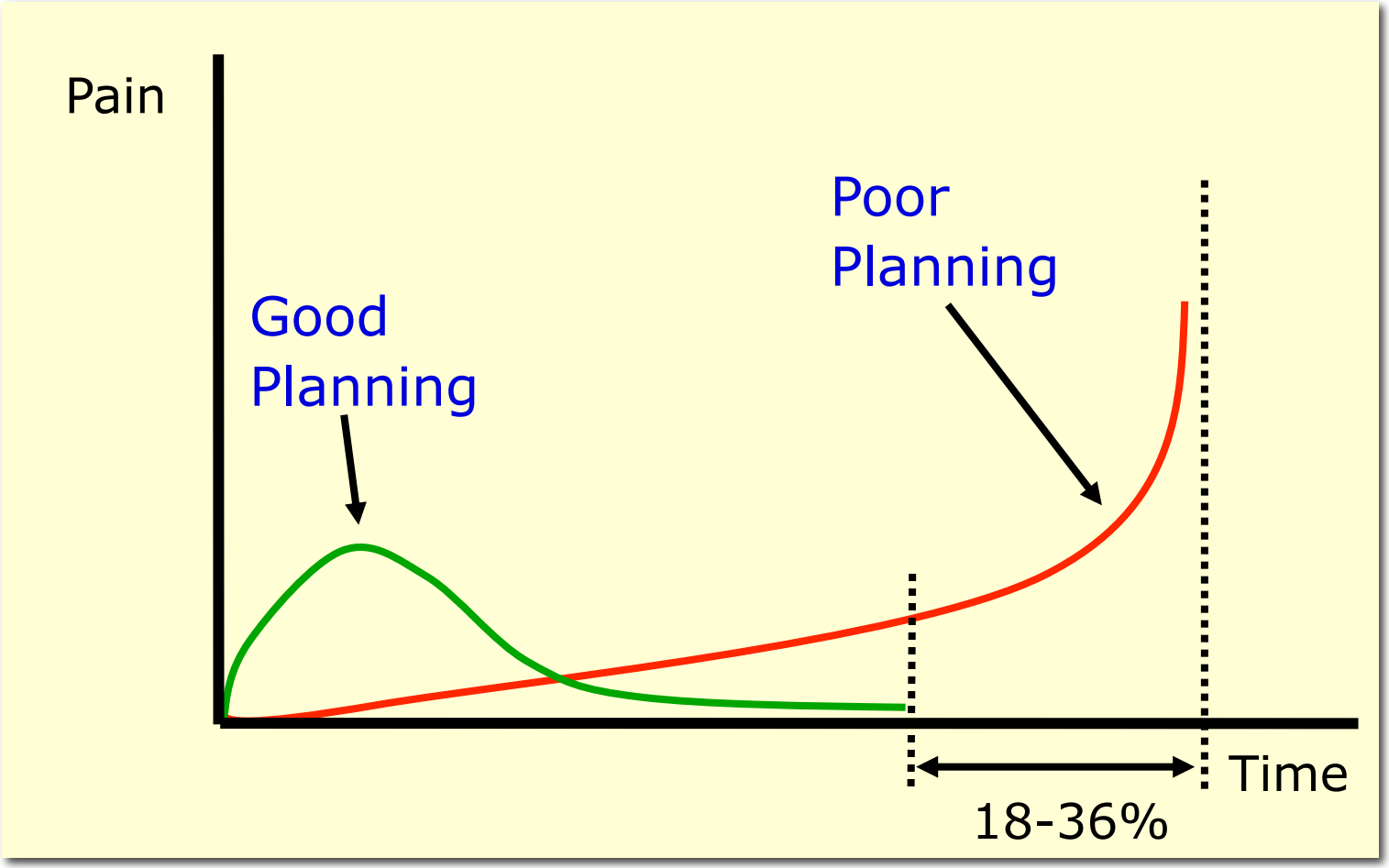
It’s Tuesday… are we productive yet?



Effect of Estimation Accuracy



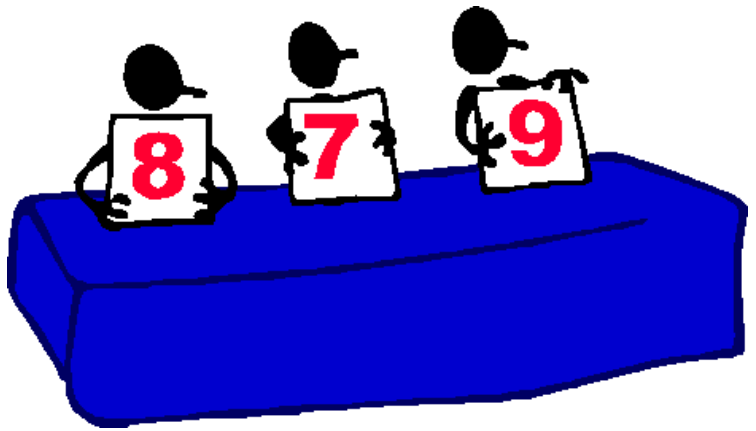
The Pain Curve



The great thing about not planning is that failure comes as a complete and utter surprise!

Importance of Planning

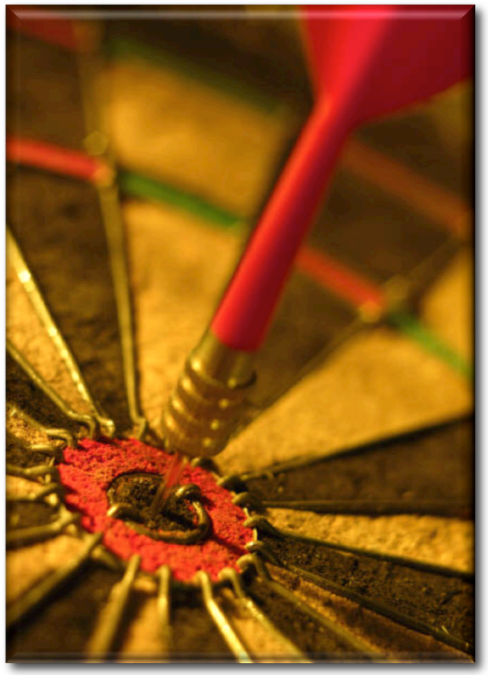
- Planning Reduces Uncertainty
- Planning Increases Understanding
- Planning Improves Efficiency



Software Project Planning Packages

Yes

- Very large projects
- Distributed teams
- Extensive use of vendors and contractors



No

- Small projects
- Short duration project increments
- Adds too much non-value-added work

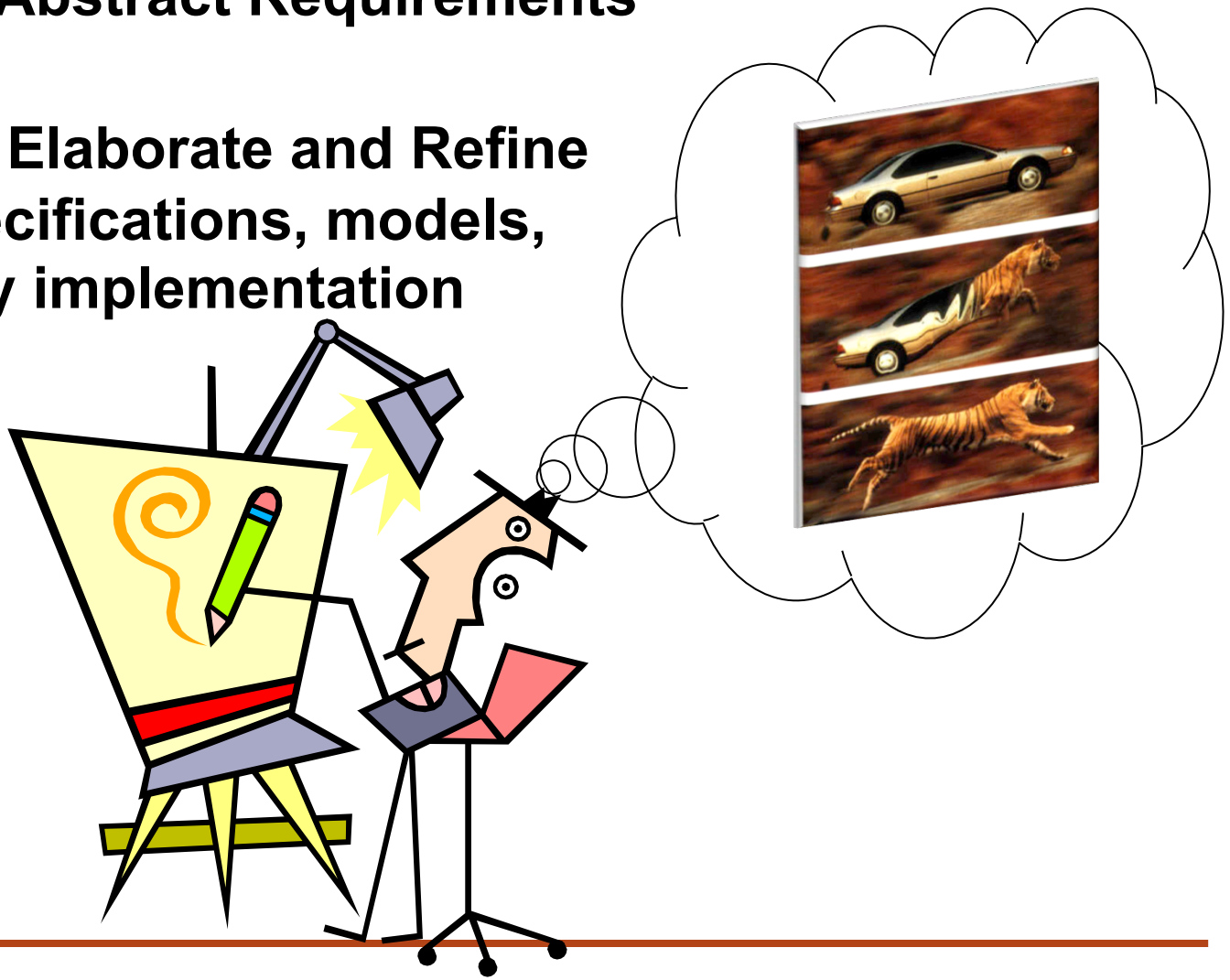
Learning Outcomes: Plan (verb)

Create a plan for an intermediate size software project & manage to the plan as project evolves.

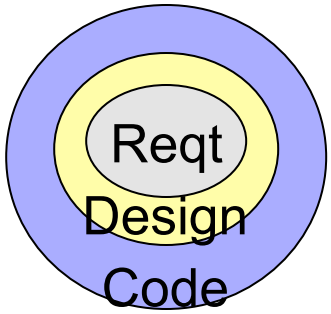
- Define elements of software life cycle process
- Relate software artifacts to software process activities
- Outline key software process models and applications

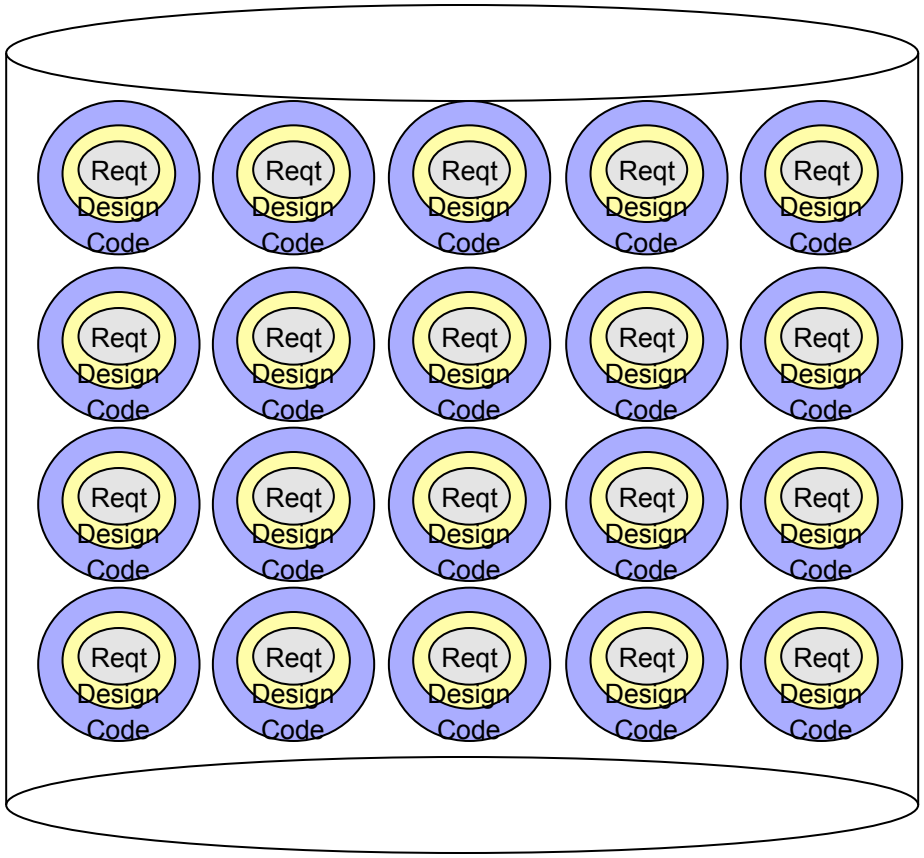
Elaboration and Refinement…

- Starting with Abstract Requirements
- Successively Elaborate and Refine them into specifications, models, and ultimately implementation



From Abstract Requirements to Concrete Systems





Software Activities & Components

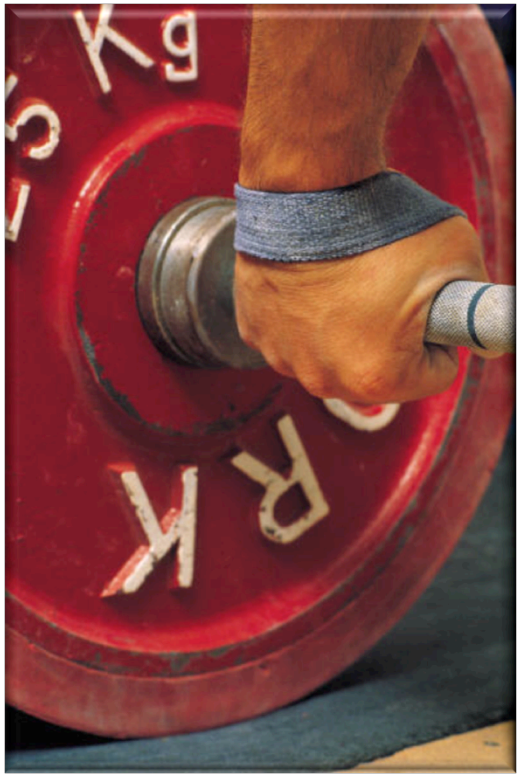
- Analysis
- Design
- Construction
- Testing

- Editor
- Compiler
- Optimizer
- Pretty printer

- Editor Requirements
- Editor Design
- Editor Code
- Editor Test Cases
- Compiler Req’ts
- Compiler Design
- Compiler Code
- Compiler Test Cases
- Optimizer …
- Pretty printer …

SW Engineering is about Scale & Quality

- People have done projects for a long time
- Their experience recorded in process models that can help you organize your project
- Small projects often need less process …

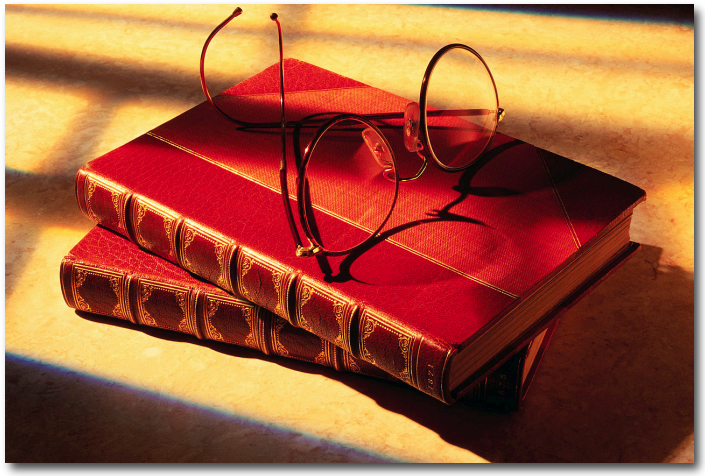


When does a small project need more process?

What is a Software Process?

- A **Software Process** organizes the life cycle activities related to the creation, delivery, and maintenance/evolution of software systems

- **Software Life Cycle Activities** are organized within a software process to provide a systematic approach for producing software products

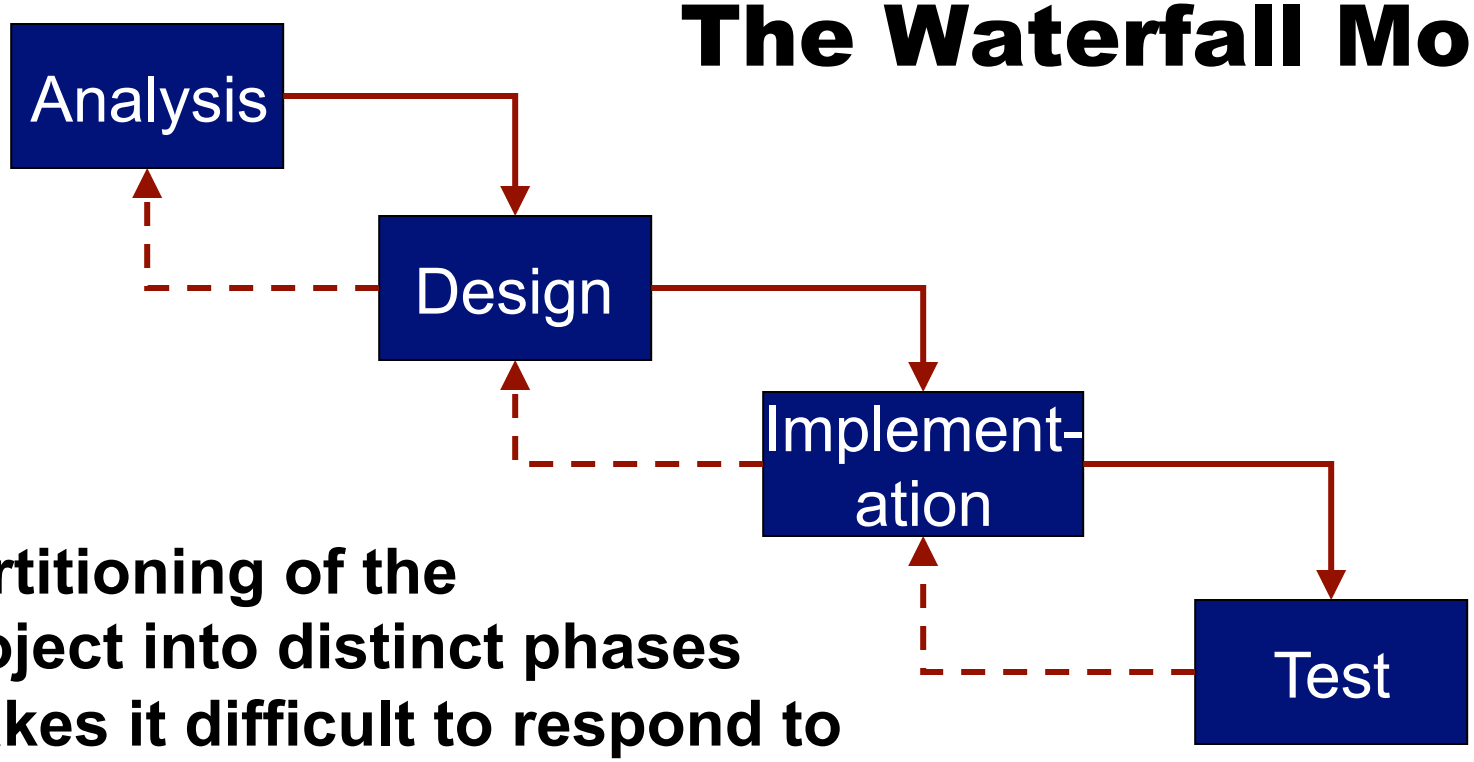


Software Process Models

- A **software process model** is an abstract description of the activities to produce a software system from a particular perspective
- This "model" is used as an archetype for the process used in a specific project

Healthy dose of Skepticism…

The Waterfall Model

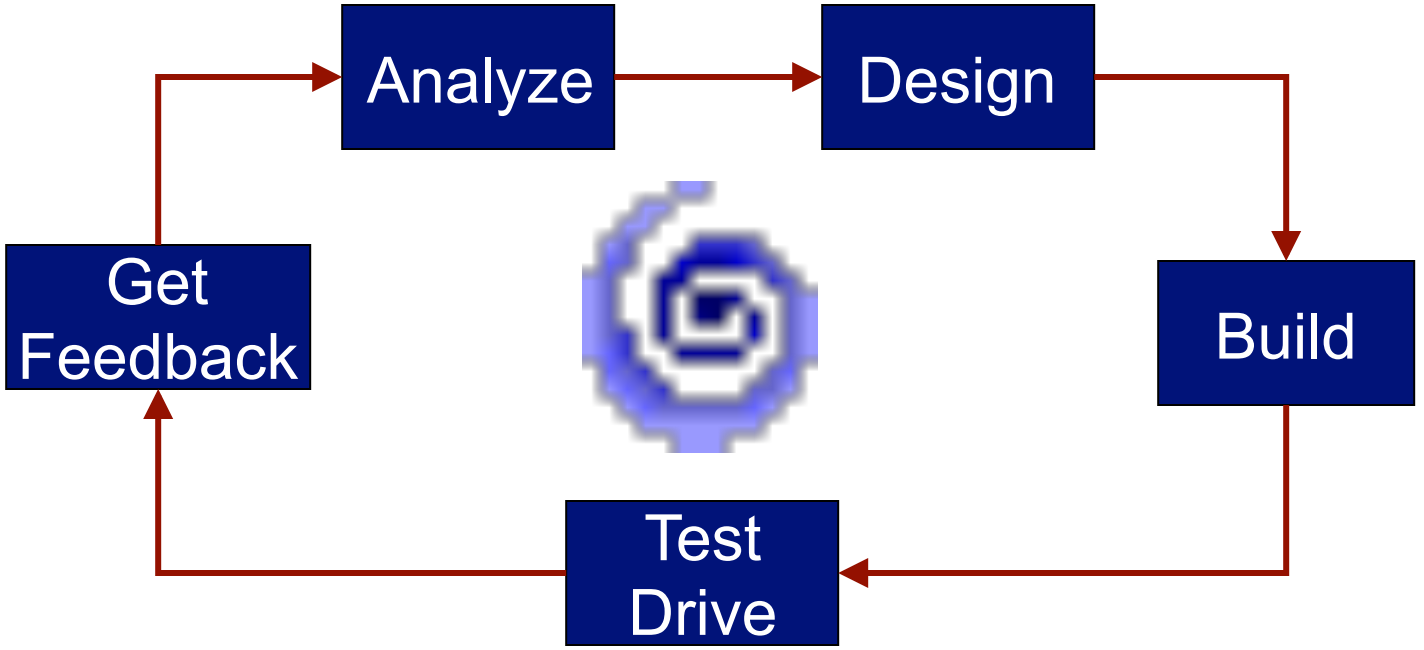


- Partitioning of the project into distinct phases makes it difficult to respond to changing customer requirements
- Appropriate when the requirements are well-understood and changes will be governed/limited
- Mostly used for large systems engineering projects where a system is developed at several sites

Cross-Life-Cycle Activities

- Cross-Life-Cycle activities are used to gate or control the life cycle activities for better flow and throughput
  - Project management
  - Risk management
  - Configuration management
  - Quality assurance
  - Measurement

Iterative Software Process Models



- AKA “Build a little, test a little” or “Learn as you go”
- Used when requirements evolve or are clarified during the course of a project
- Iteration applied to any of the generic process models

Evolutionary Development Process

- Exploratory development
  - Objective is to work with customers and to evolve a final system from an initial outline specification
  - Start with well-understood requirements and add new features as proposed by the customer

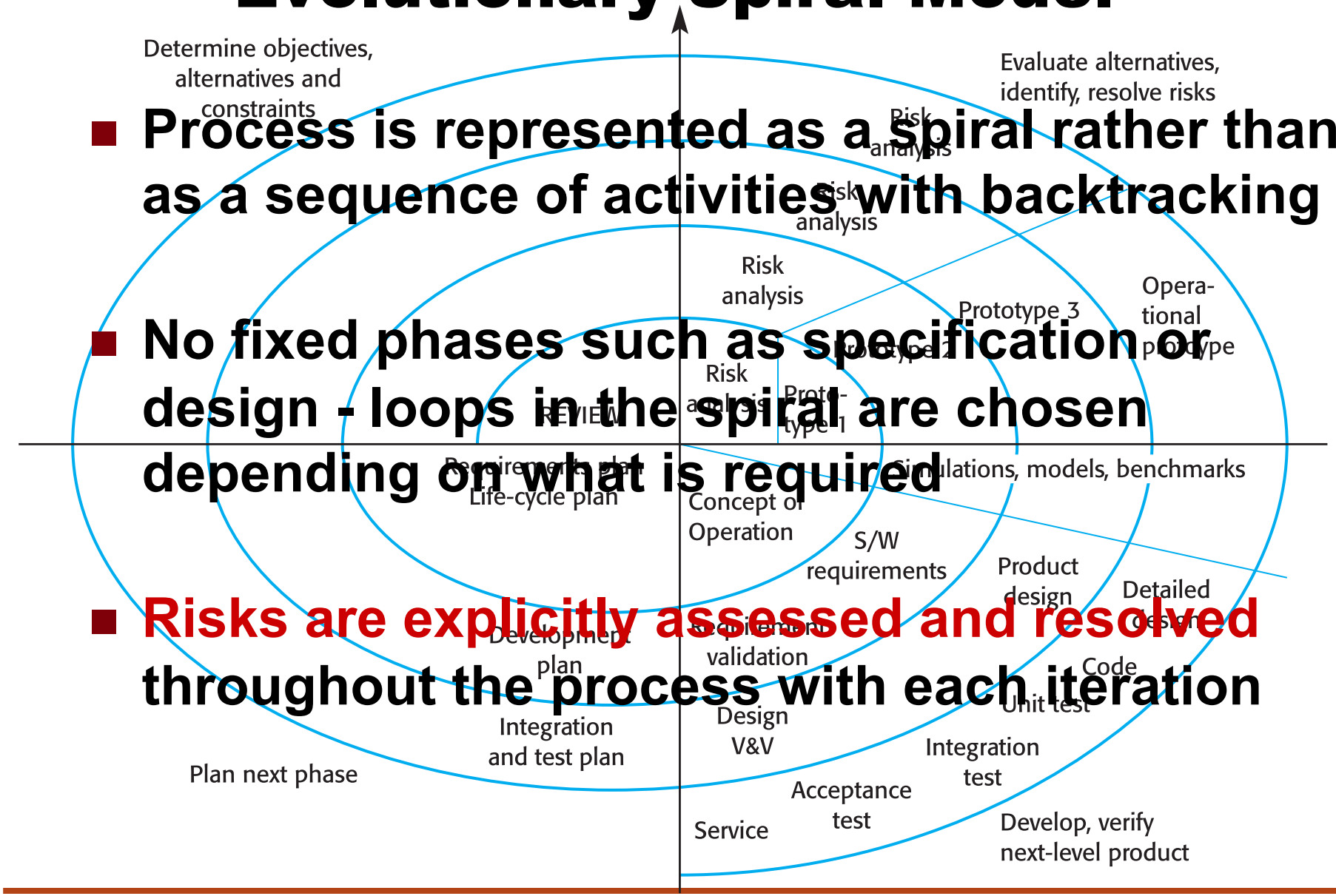
- Throw-away prototyping
  - Objective is to understand system requirements
  - Start with poorly understood requirements to clarify what is really needed

Evolutionary Development (continued)

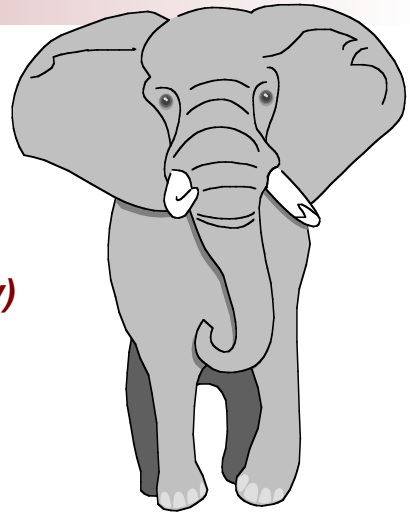
- Challenges
  - Lack of process visibility can lead to ad hoc results
  - Systems are often poorly structured since the evolution is more organic and less systematic
  - Special skills (e.g. in languages for rapid prototyping) may be required
- Applicability
  - For small or medium-size interactive systems
  - For parts of large systems (e.g. the user interface)
  - For short-lifespan systems

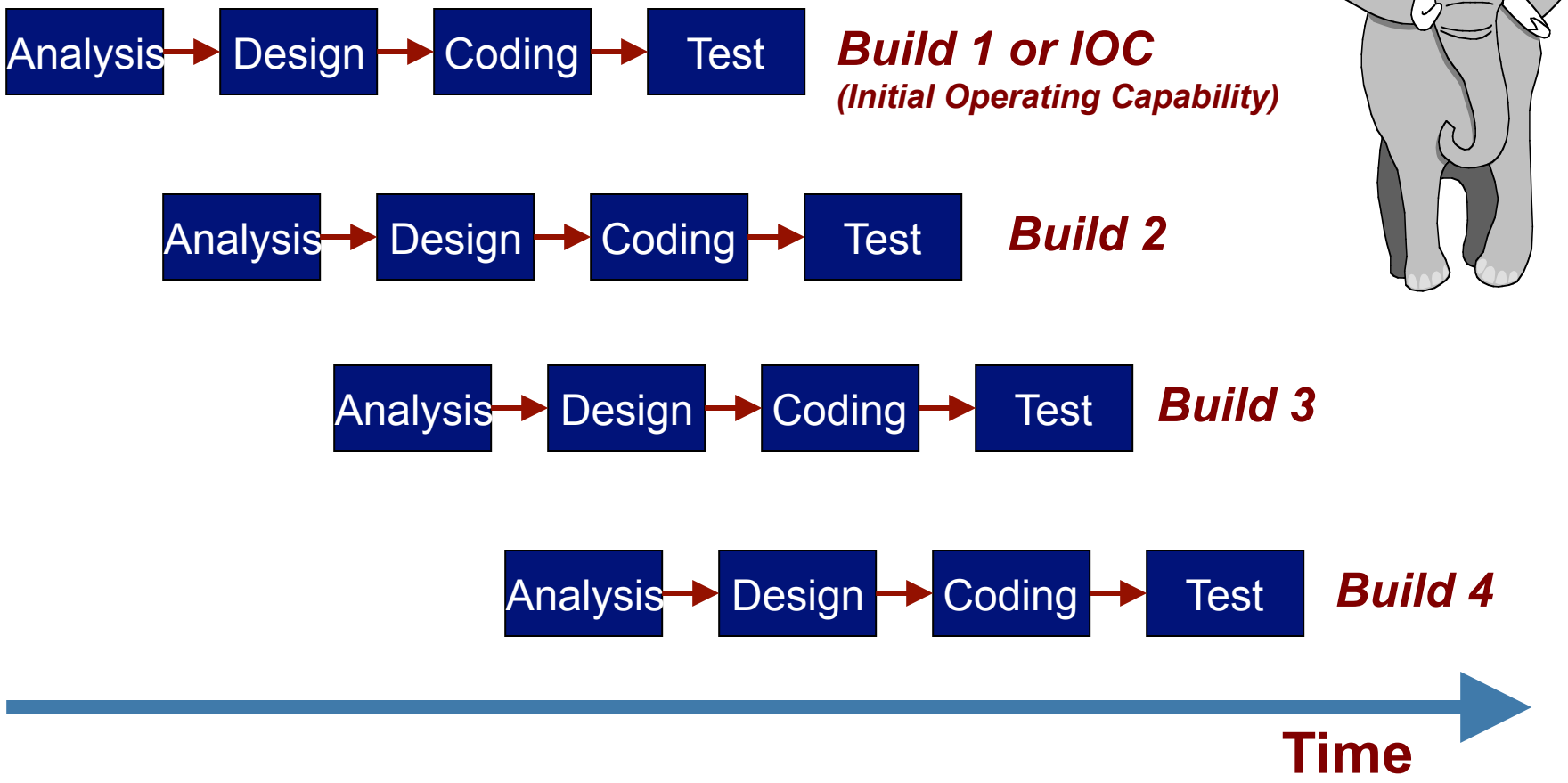
Evolutionary Spiral Model

- Process is represented as a spiral rather than as a sequence of activities with backtracking
- No fixed phases such as specification or design - loops in the spiral are chosen depending on what is required
- Risks are explicitly assessed and resolved throughout the process with each iteration



Incremental Process Models





Systematic divide and conquer strategy for completing projects using concurrent activities

Incremental Development/Delivery

- Starting with an initial operating capability (IOC), delivery is broken down into increments
  - Each increment delivers set of required functionality
  - High-priority requirements in early increments
  - When increment starts, the increment requirements are frozen
- Advantages of Incremental Development
  - Customer value can be delivered with each increment so system functionality is available earlier
  - Early increments act as a prototype to help elicit requirements for later increments
  - Reduced schedule scope => Lower risk of project failure
  - High priority system services tend to receive the most testing
- Applied when the components are isolated and there are teams that can work concurrently

Homework and Reading Reminders

- Read SimSE Introduction
- Watch SimSE Tutorials
- Complete Homework 2 – Play SimSE Game and report results
  - Due by 5pm, Tuesday, September 18th, 2012.