



CSSE 372 Software Project Management: Software Processes and SimSE Game

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

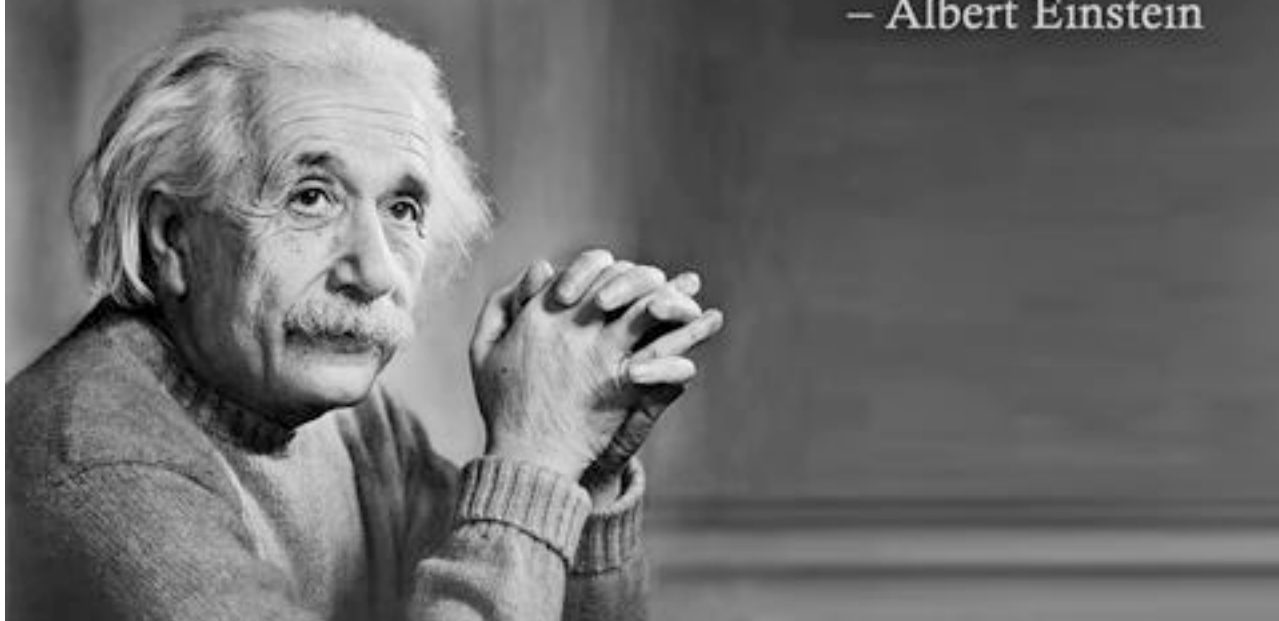


ROSE-HULMAN
INSTITUTE OF TECHNOLOGY



If you can't explain it **simply**, you don't understand it well enough.

– Albert Einstein

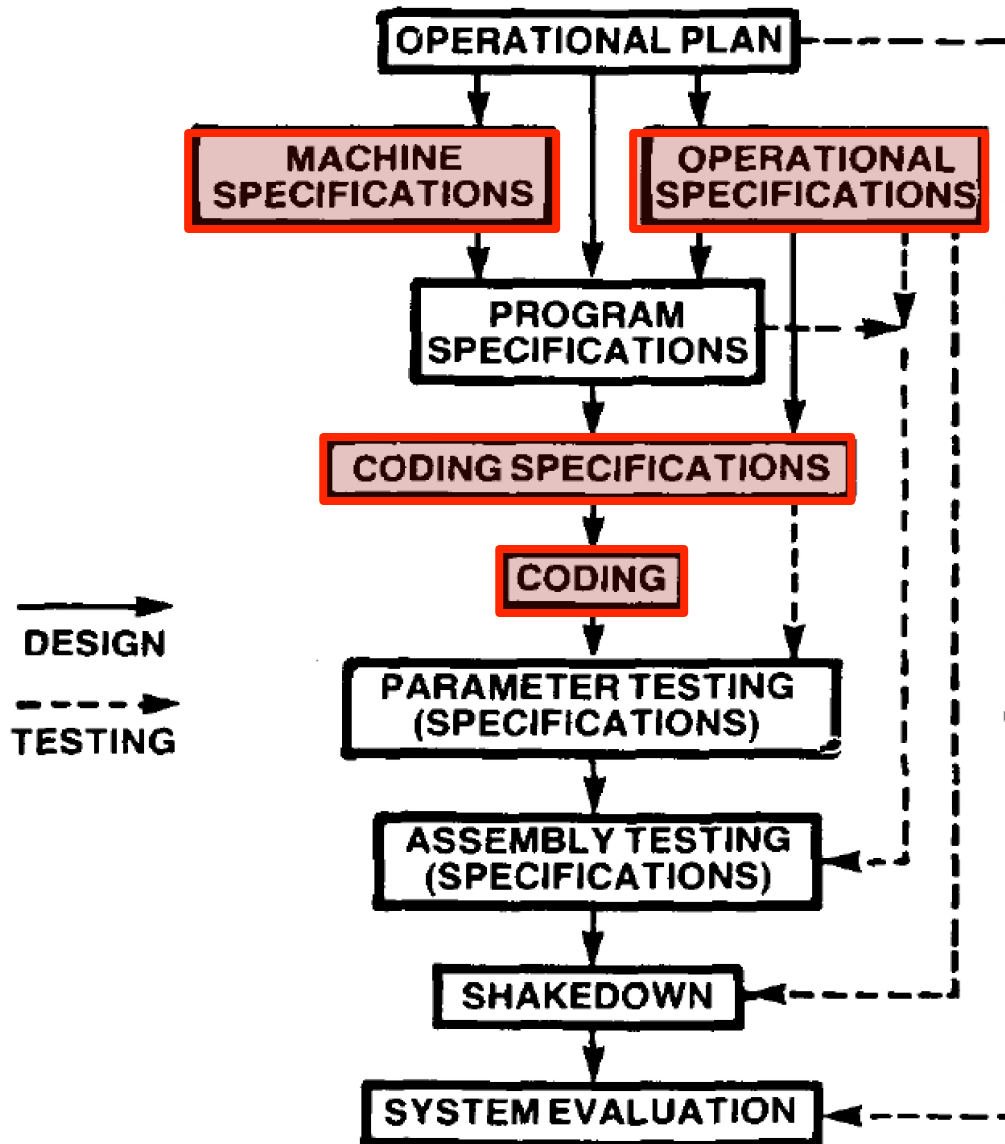


Learning Outcomes: Life Cycle

Explain and employ contemporary software life cycle processes, activities, and work products.

- Examine history of software process
- Experience a software project through simulation or game
- Answer questions about a software life cycle

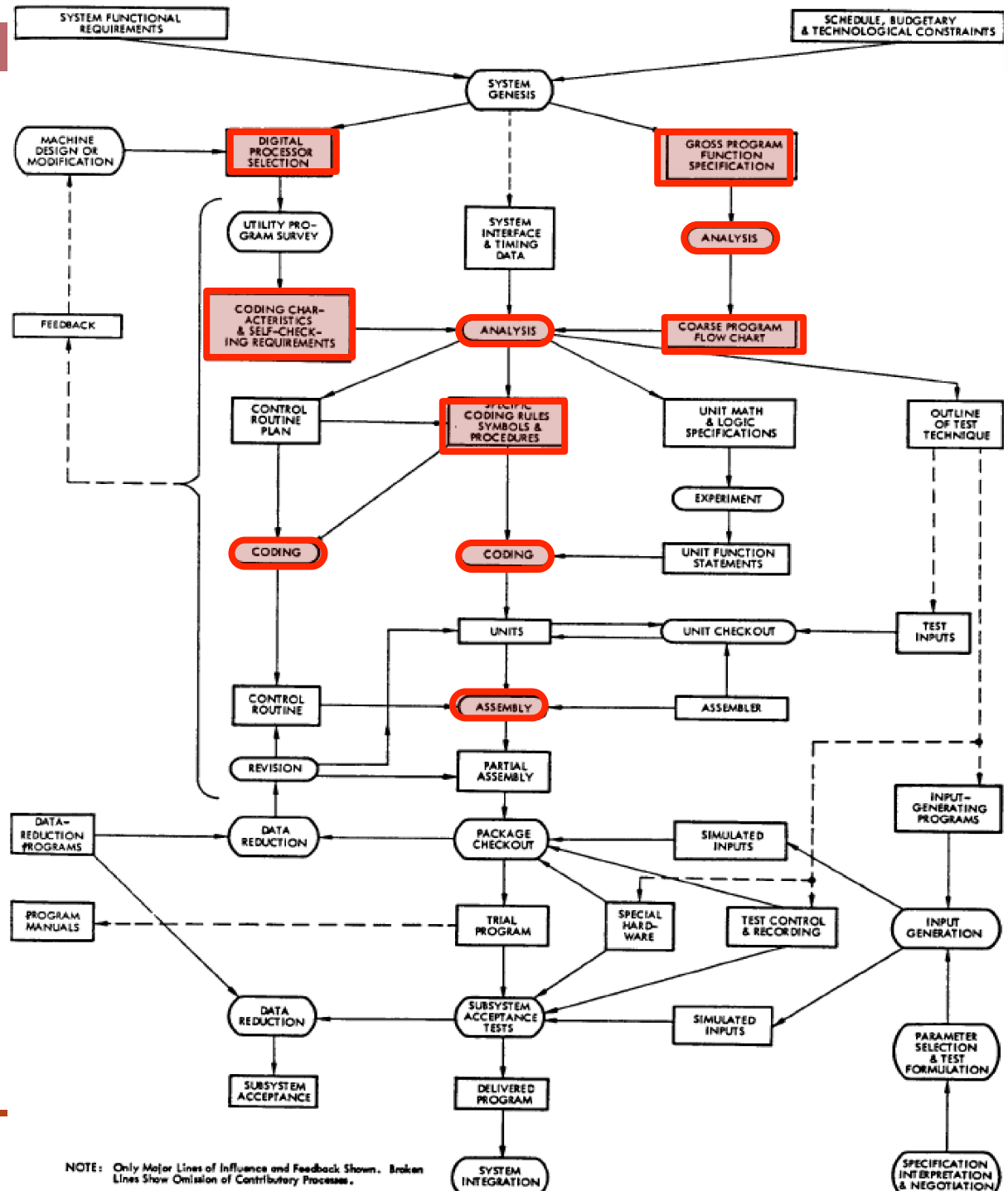




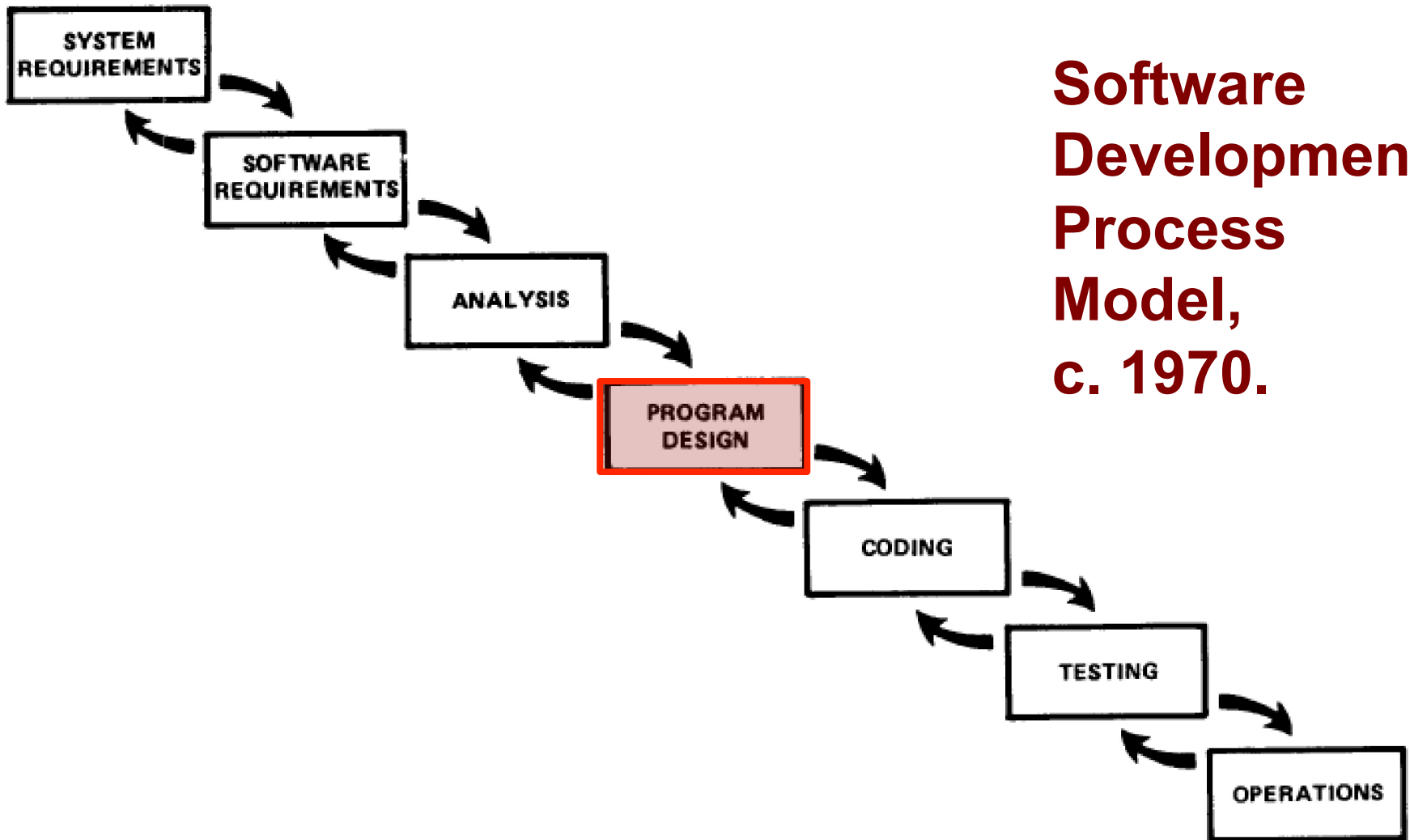
**Software
Development
Process Model,
c. 1955.**

*It's all about
coding...*

Software Development Process Model, c. 1960.



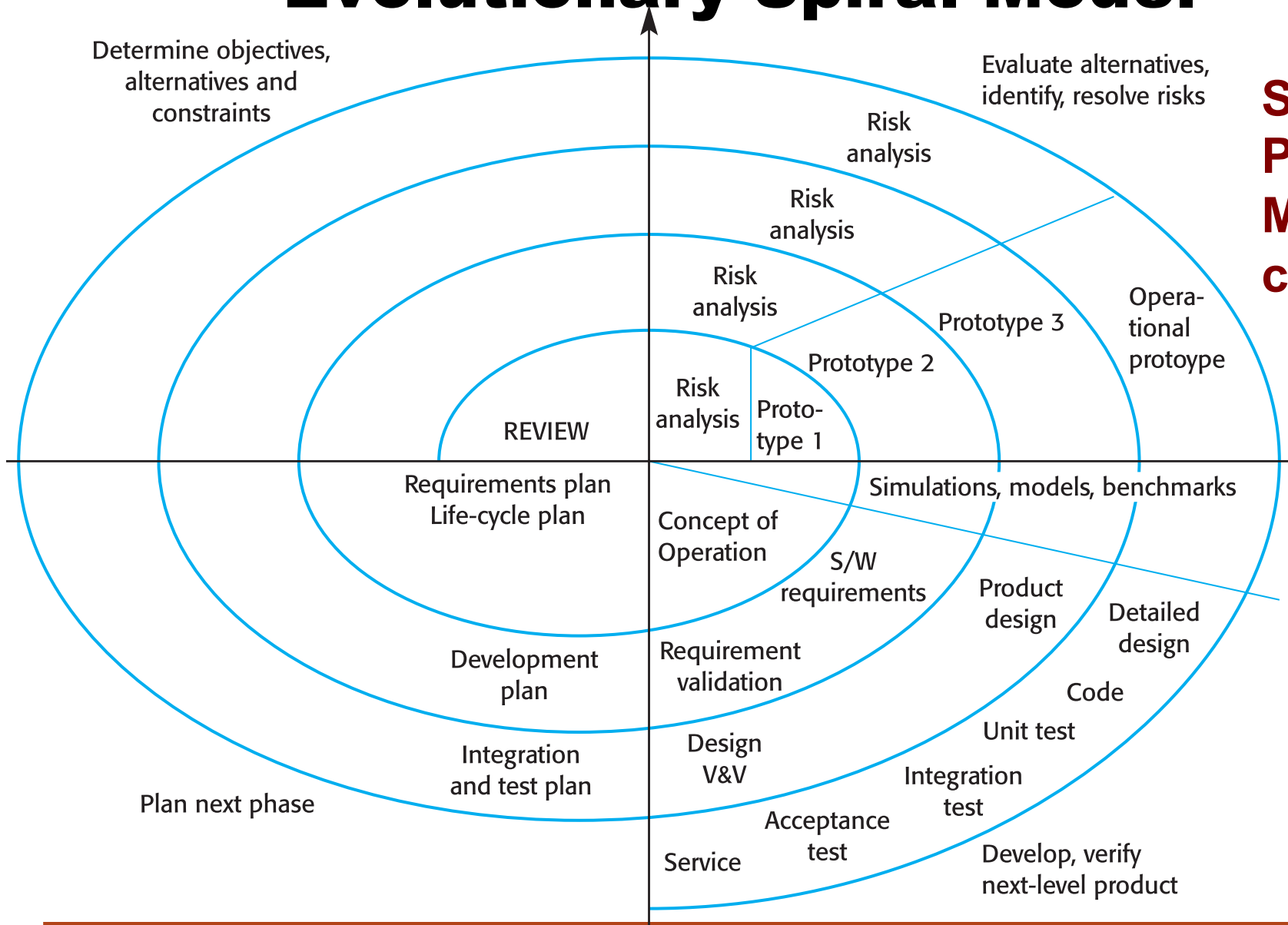
NOTE: Only Major Lines of Influence and Feedback Shown. Broken Lines Show Omission of Contributory Processes.



Software Development Process Model, c. 1970.

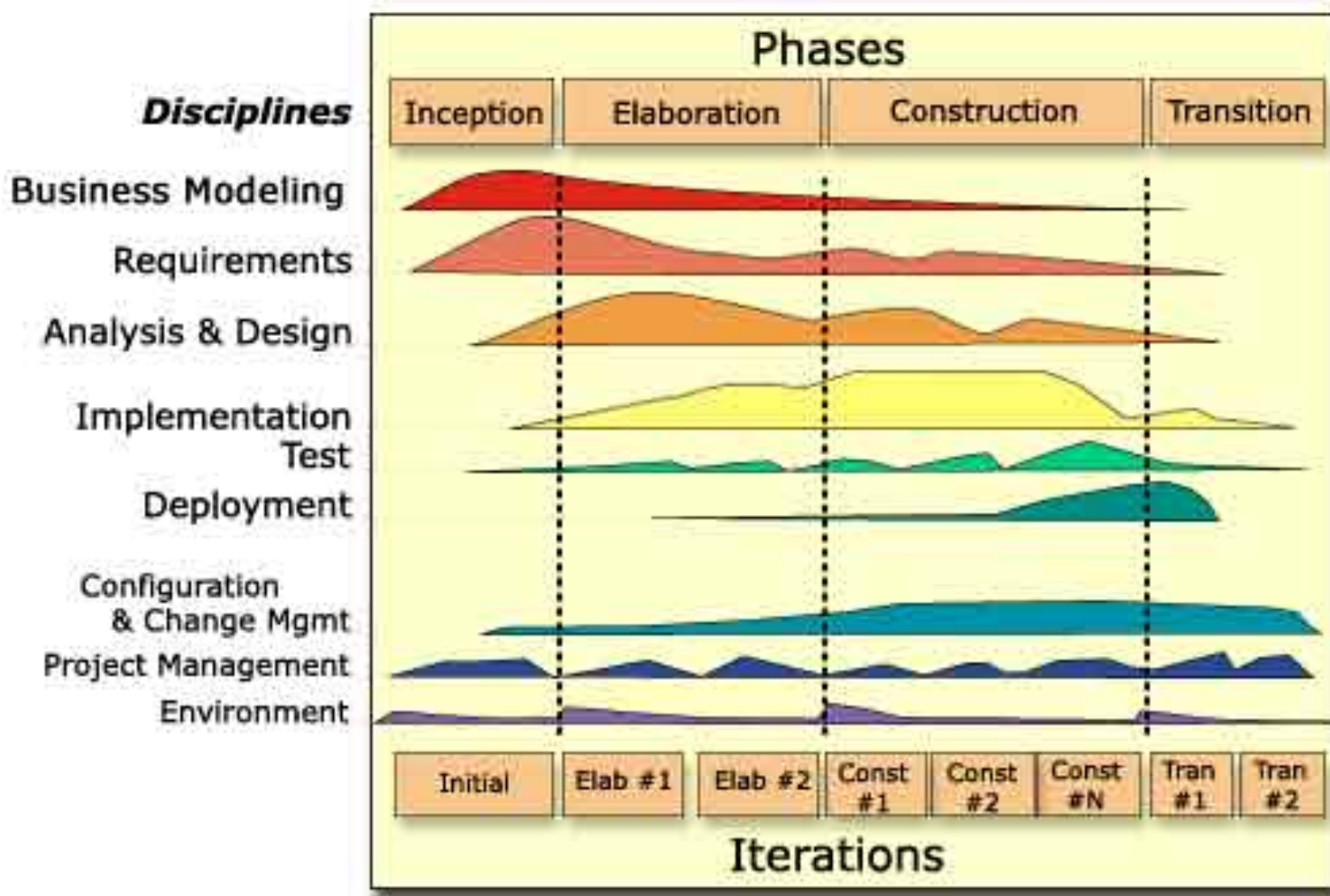
Figure 3. Hopefully, the iterative interaction between the various phases is confined to successive steps.

Evolutionary Spiral Model



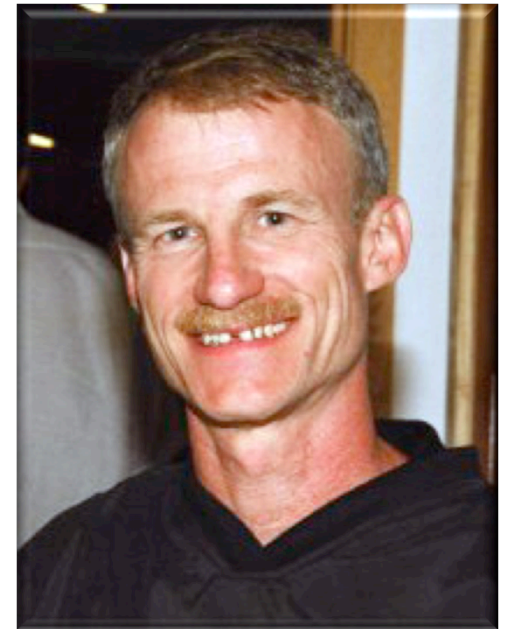
**Software
Process
Model,
c. 1987.**

Unified Process (UP)



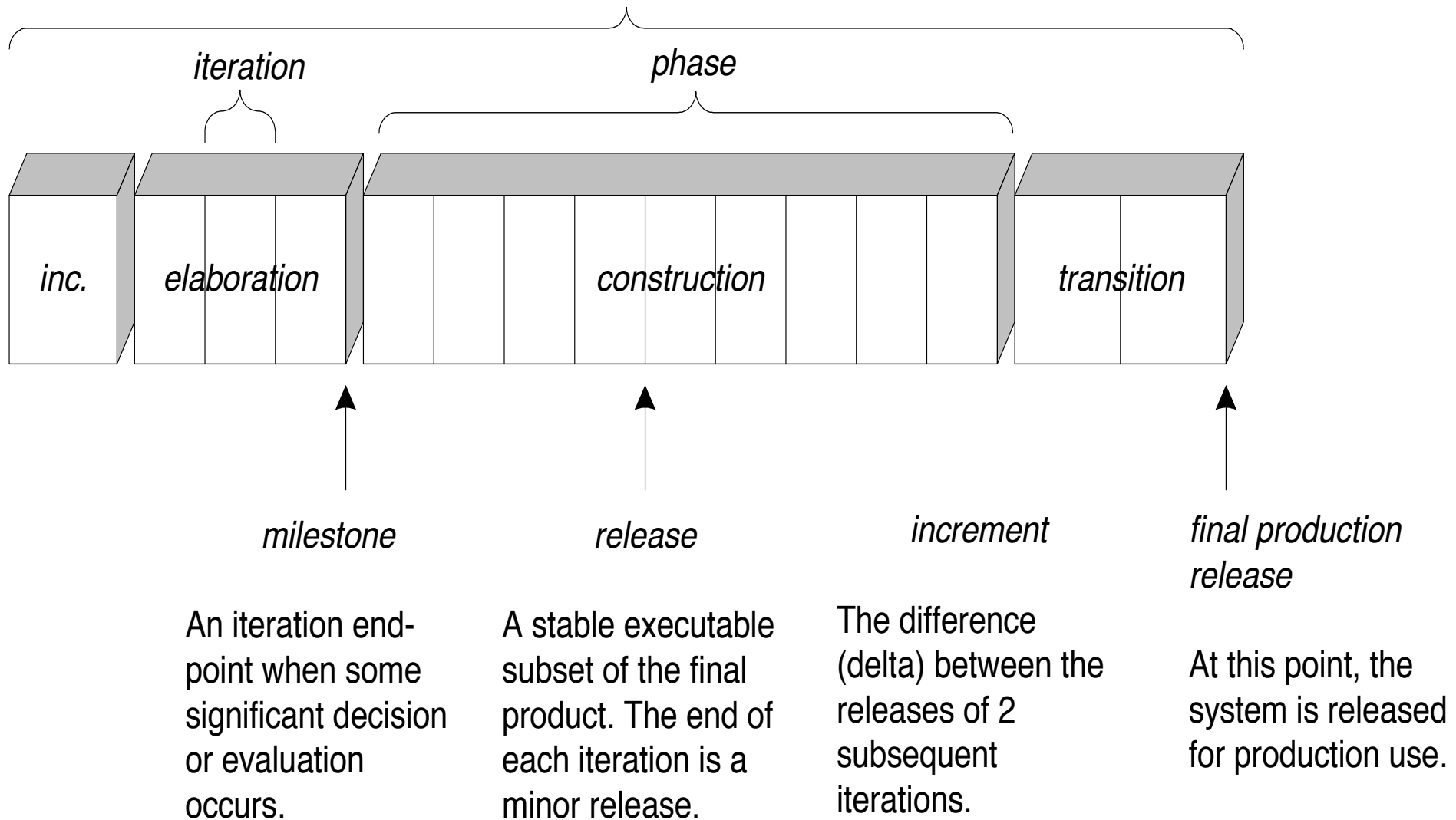
Winston and Walker Royce

- **Winston Royce derived Waterfall Model from Systems Engineering**
- **Walker Royce was on the Rational Team that developed Unified Process**



Unified Process (UP) Vernacular

development cycle



Process Improvement Gone Funky

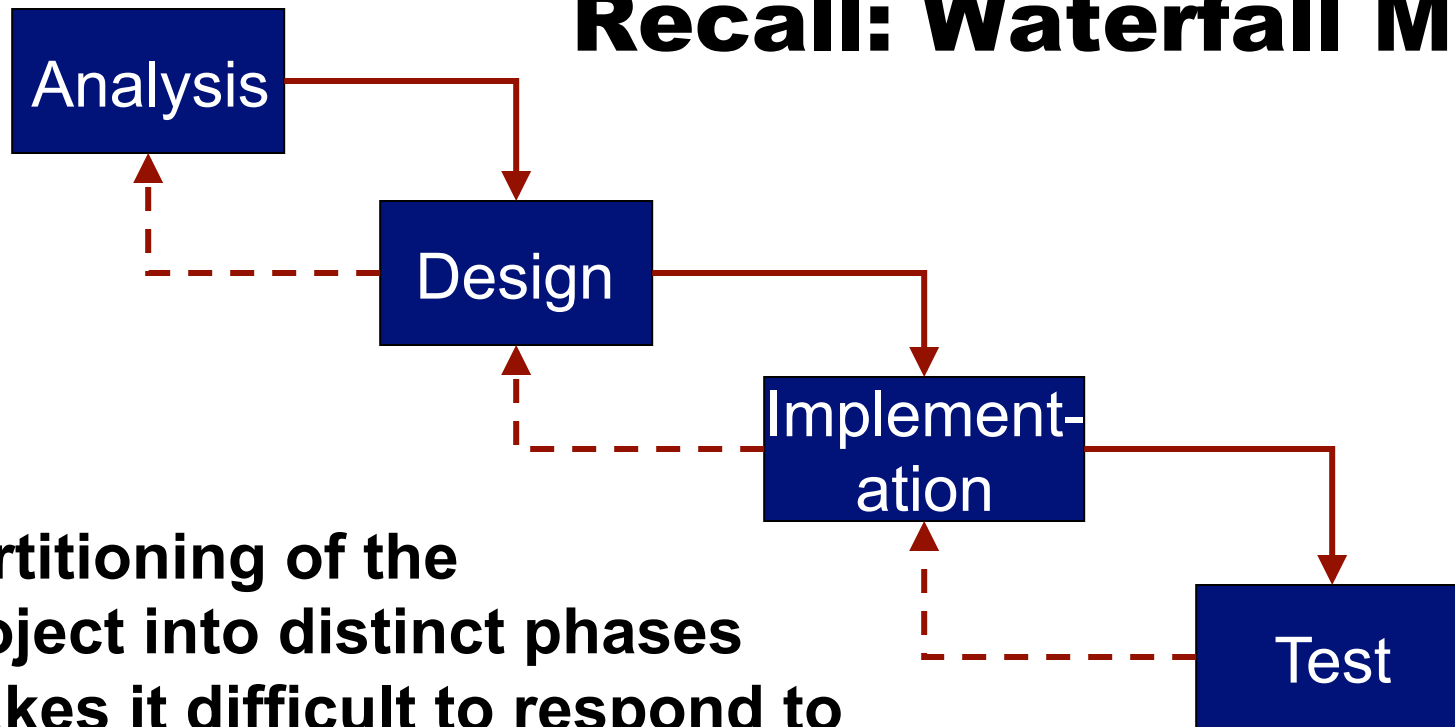
Savage Chickens

by Doug Savage



www.savagechickens.com

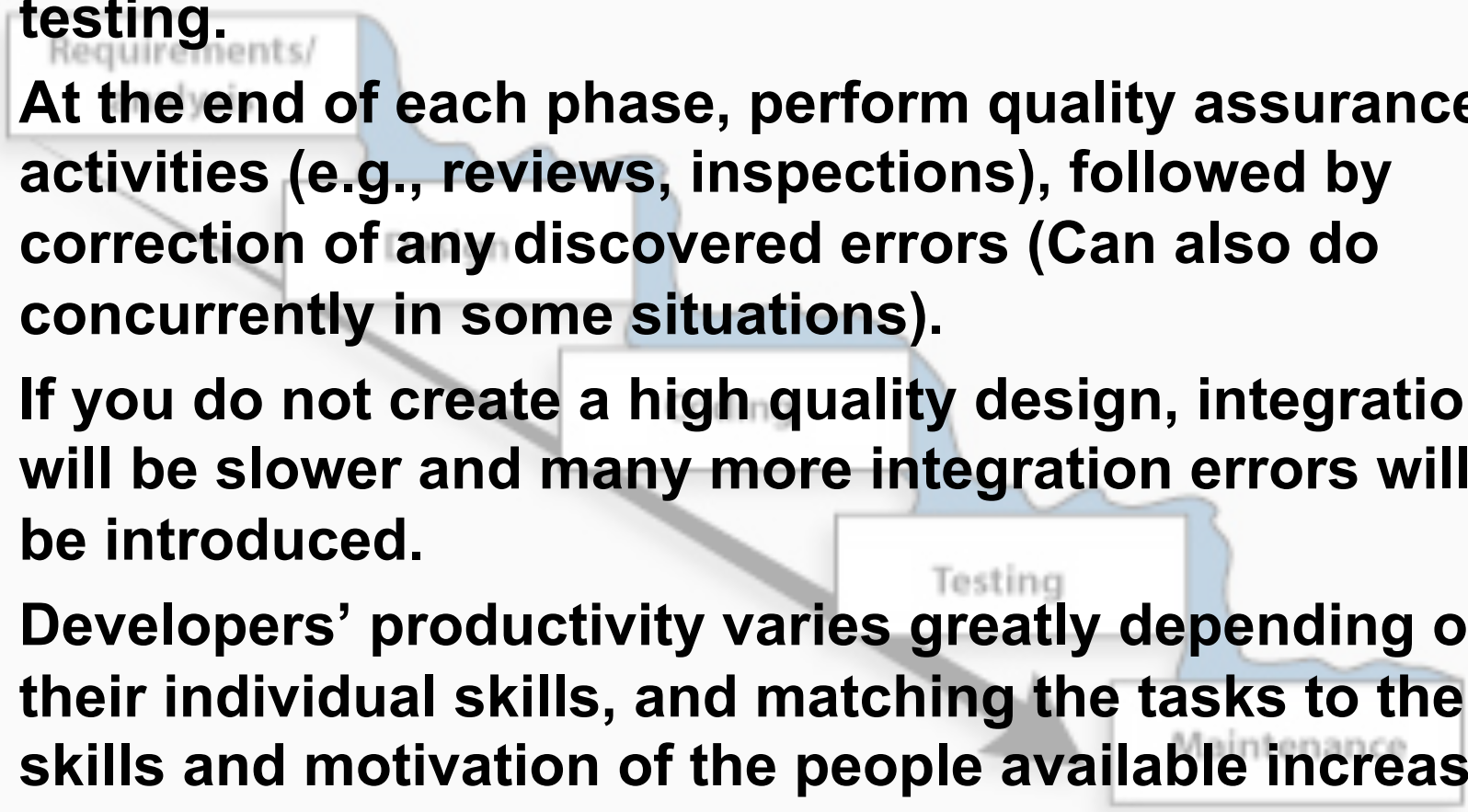
Recall: Waterfall Model



- 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

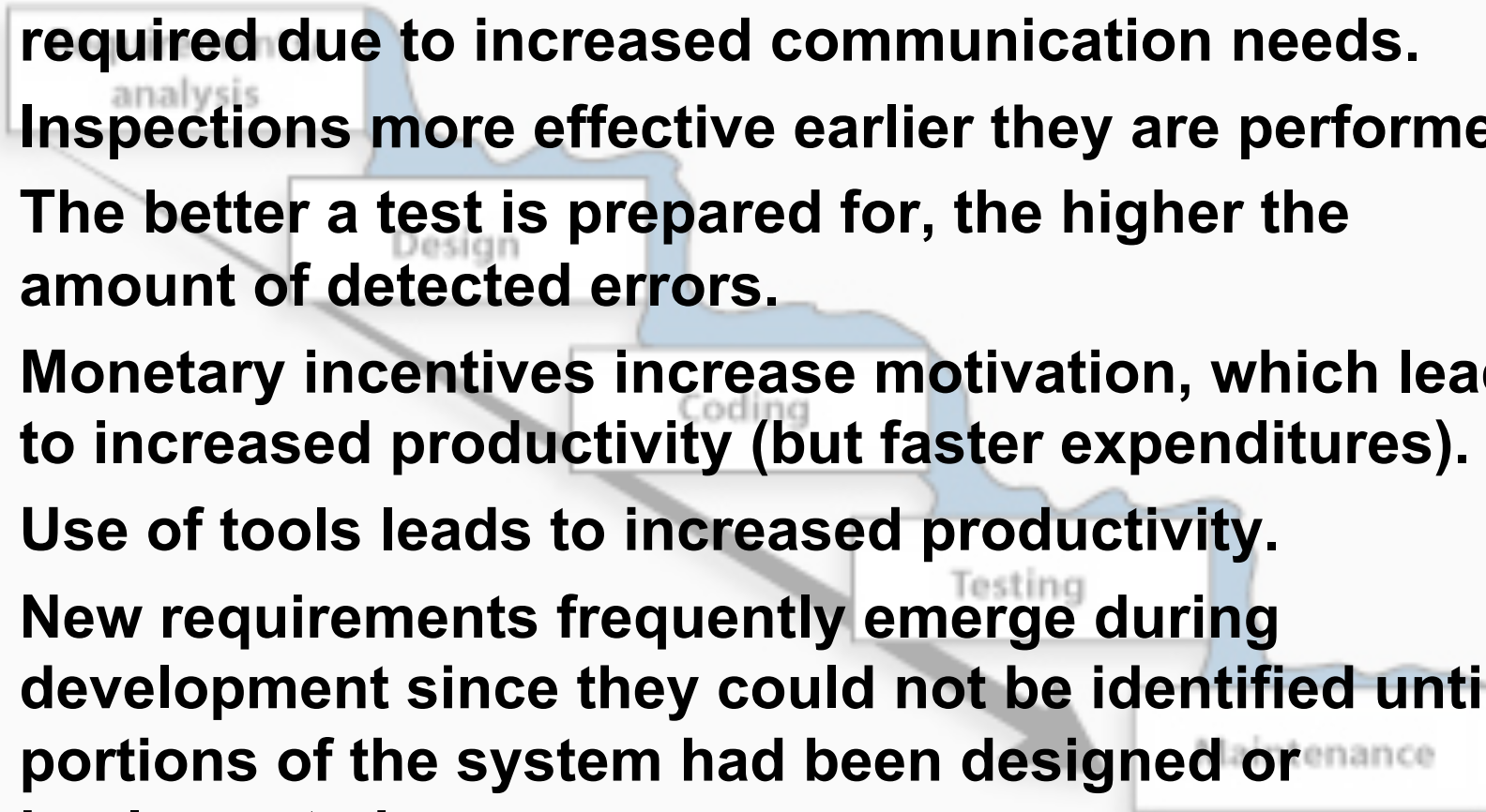


Waterfall Observations 1/2

- Do requirements, followed by design, followed by implementation, followed by integration, followed by testing.
 - At the end of each phase, perform quality assurance activities (e.g., reviews, inspections), followed by correction of any discovered errors (Can also do concurrently in some situations).
 - If you do not create a high quality design, integration will be slower and many more integration errors will be introduced.
 - Developers' productivity varies greatly depending on their individual skills, and matching the tasks to the skills and motivation of the people available increases productivity.
- 



Waterfall Observations 2/2

- More developers working on a task simultaneously, the faster the task is finished, but more effort is required due to increased communication needs.
 - Inspections more effective earlier they are performed.
 - The better a test is prepared for, the higher the amount of detected errors.
 - Monetary incentives increase motivation, which leads to increased productivity (but faster expenditures).
 - Use of tools leads to increased productivity.
 - New requirements frequently emerge during development since they could not be identified until portions of the system had been designed or implemented
- 

Exercise: SimSE Waterfall Model

Task: create Groceries@Home, a Web-based system for people to place orders over Net for groceries

Customer: Grocery Home Delivery Service

Budget: \$280K and 1,350 clock ticks to complete the project

Objectives:

1. Complete, error-free code, integrated code
2. Budget and schedule performance

Notes:

1. Employees always paid every clock tick
2. Must specify tools when you assign the task





SimSE Exercise

- **Download and Bring up Waterfall SimSE Game**
 - <http://www.ics.uci.edu/~emilyo/SimSE/downloads.html>
 - Or Thumb drive...
- **Look at Properties of Players and Artifacts**

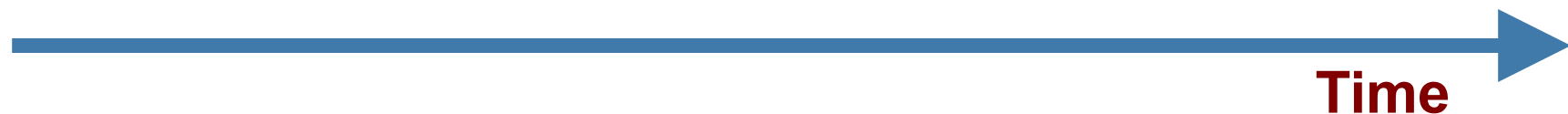
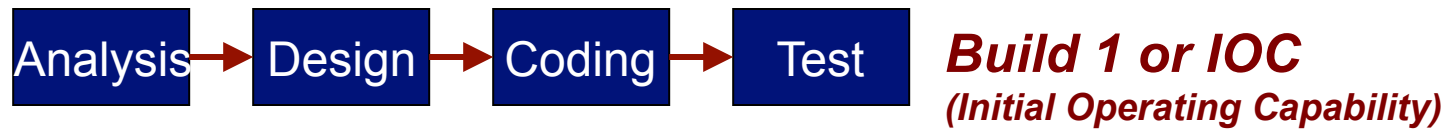
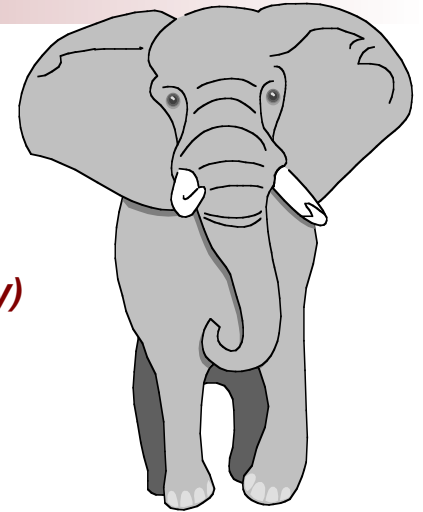
- **Start by getting tools**
- **Get team members to do some requirements**
- **Get others to review and correct requirements**
- **Check project numbers & Graph actions for clues...**
- **Do Design...**
- **Do Implementation...**
- **Do Testing...**



Questions to Consider:

- **Describe the process (sequence of possible steps that you can take in the game) that this game rewards.**
- **What is the effect of giving an employee a bonus?**
- **What is the effect of giving an employee a pay raise?**
- **Is it worth it to purchase tools?**
- **How is the outcome of the game affected if you fire Andre right at the beginning?**
- **What are the effects of skipping the requirements phase?**
- **What are the effects of skipping the design phase?**

Recall: Incremental Process



Systematic divide and conquer strategy for completing projects using concurrent activities



Recall: Incremental Process Model

- **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 Selecting Software Process Paper**

- **Complete Homework 2 – Play SimSE Game and report results according to assignment**
 - Due by 5pm, Tuesday, September 18th, 2012

- **Get a Head Start on COCOMO**
 - Look over COCOMO-II User Manual and Model Manual
 - Download COCOMO-II Software (from Angel)