

LEC # 04

SOFTWARE ENGINEERING

Software Development Phases

- Design / Definition

 - ▣ What?

- Development

 - ▣ How?

- Support

 - ▣ Change

 - Corrective

 - Adaptive

 - Enhancement

 - Prevention

Software Development Process

- Process
 - ▣ A set of (ordered) activities
- Software Process
 - ▣ A structured set of activities required to develop a software system
 - ▣ The roadmap to building high quality software products
- *"If you don't know where you're going, any road will do."*
- *"If you don't know where you are, a map won't help."*

Watts Humphrey, *Managing the Software Process*

Software Process Models

- What is a model?
 - ▣ A model is a simplification of reality. We model because we cannot comprehend the complexity of a system in its entirety.
 - ▣ A model is a description of a system from a particular perspective.
- Software Process Model
 - ▣ An abstract representation of a process.
 - ▣ It presents a description of a process from some particular perspective

Software Process Models

□ Principles of Modeling

- The choice of what models to create has a profound influence on how a problem is attacked and how a solution is shaped.
- Every model may be expressed at different levels of precision.
- The best models are connected to reality.

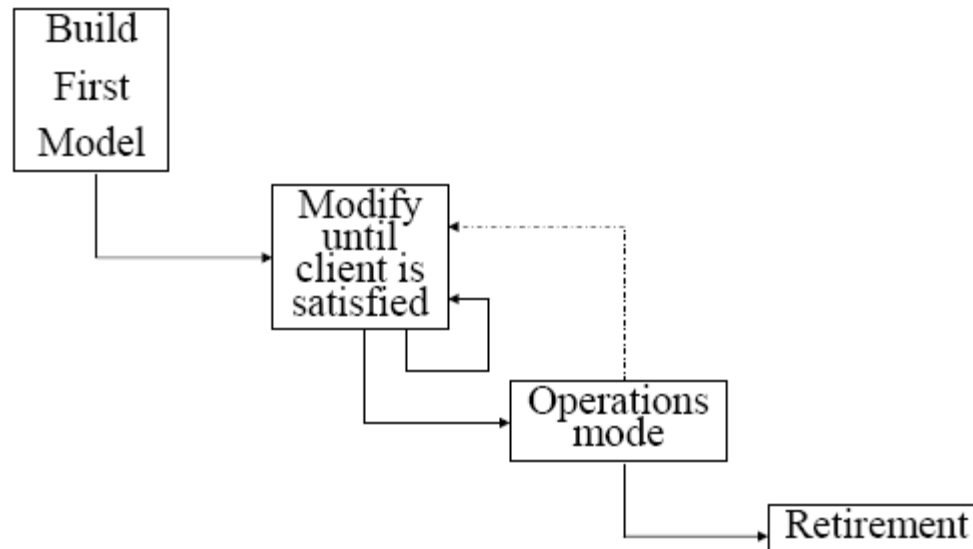
Process and Product

- The quality of a product is largely governed by the quality of the process used to build it.
- The quality of a *software* product is largely governed by the quality of the *software* process used to develop and maintain it.
- A process description should be detailed enough to be useful, not so detailed that it is unusable.

Software Life-Cycle Models

- ☐ Build and Fix
- ☐ Water fall
- ☐ Prototyping
- ☐ RAD
- ☐ Incremental
- ☐ Spiral
- ☐ Win Win Spiral
- ☐ RUP
- ☐ Agile

Build and Fix



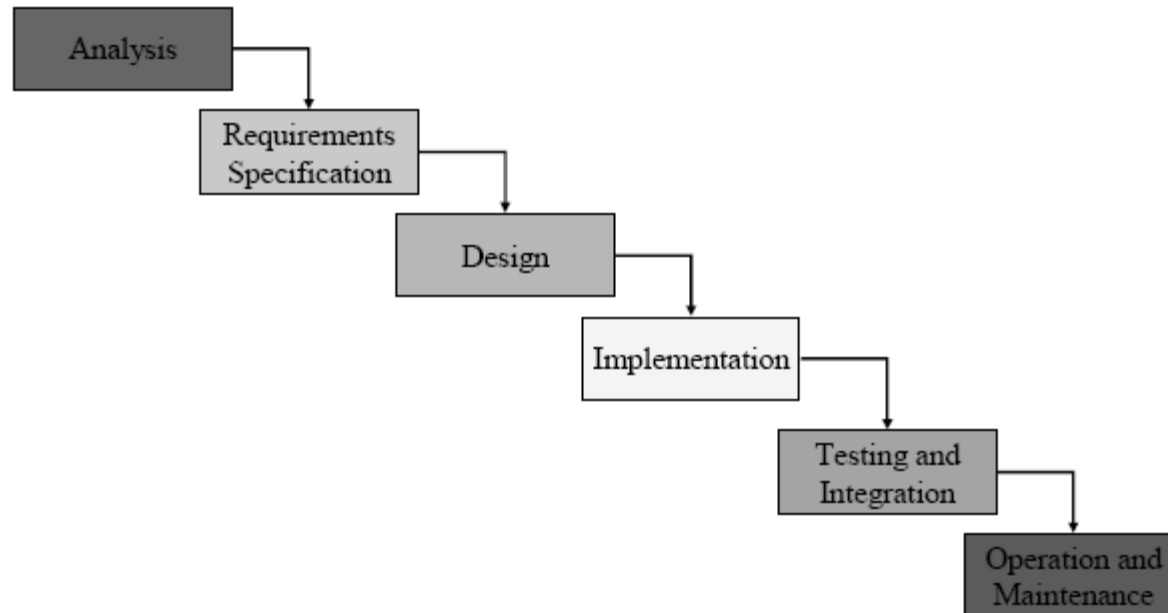
Build and Fix



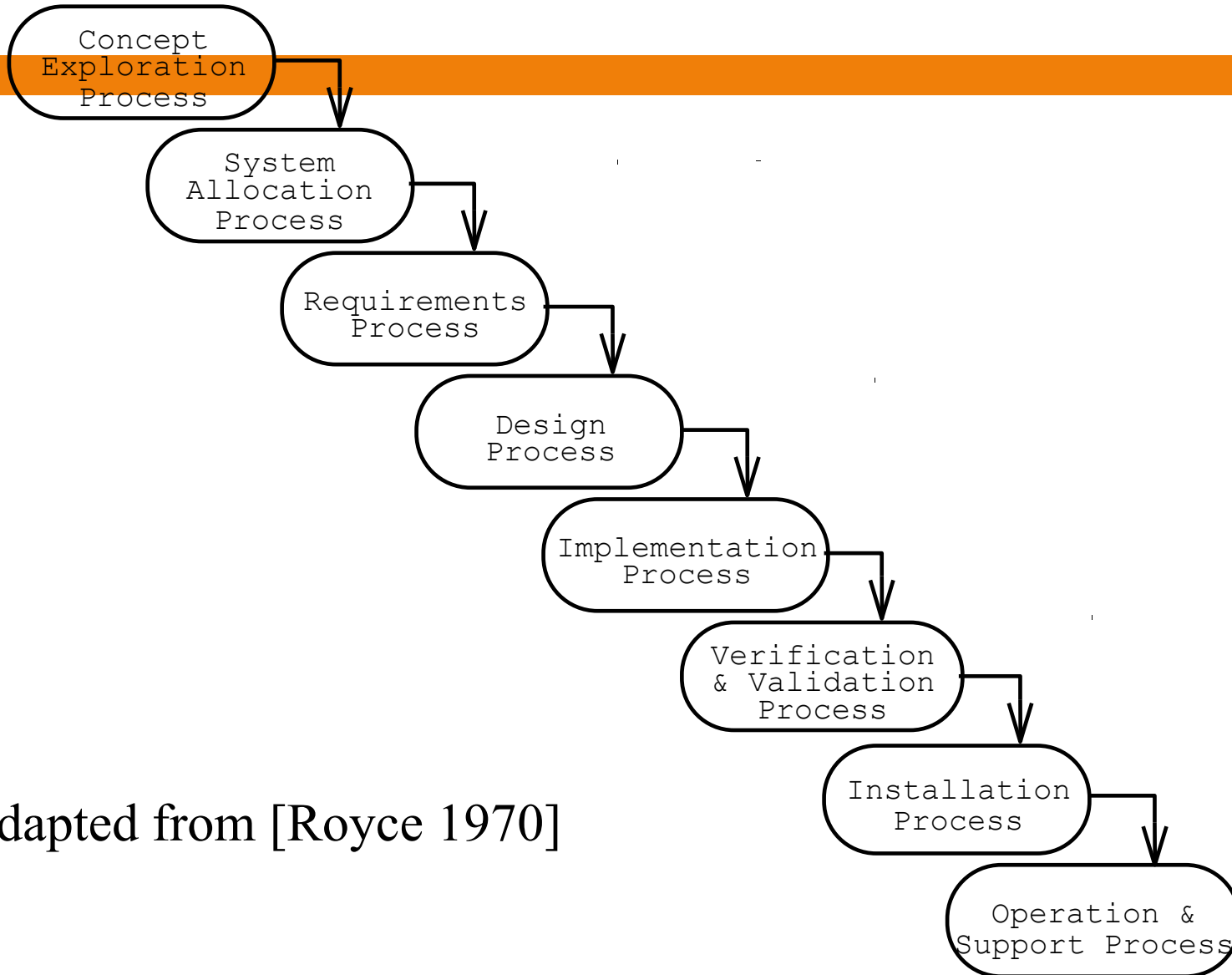
- ❑ No formal specifications
- ❑ No formal design
- ❑ No documentation
- ❑ Rework is out of control
- ❑ Loss of client confidence
- ❑ High cost
- ❑ Future maintenance difficult/impossible

Classical Waterfall Model

Linear sequential model



Waterfall Model of the SDLC

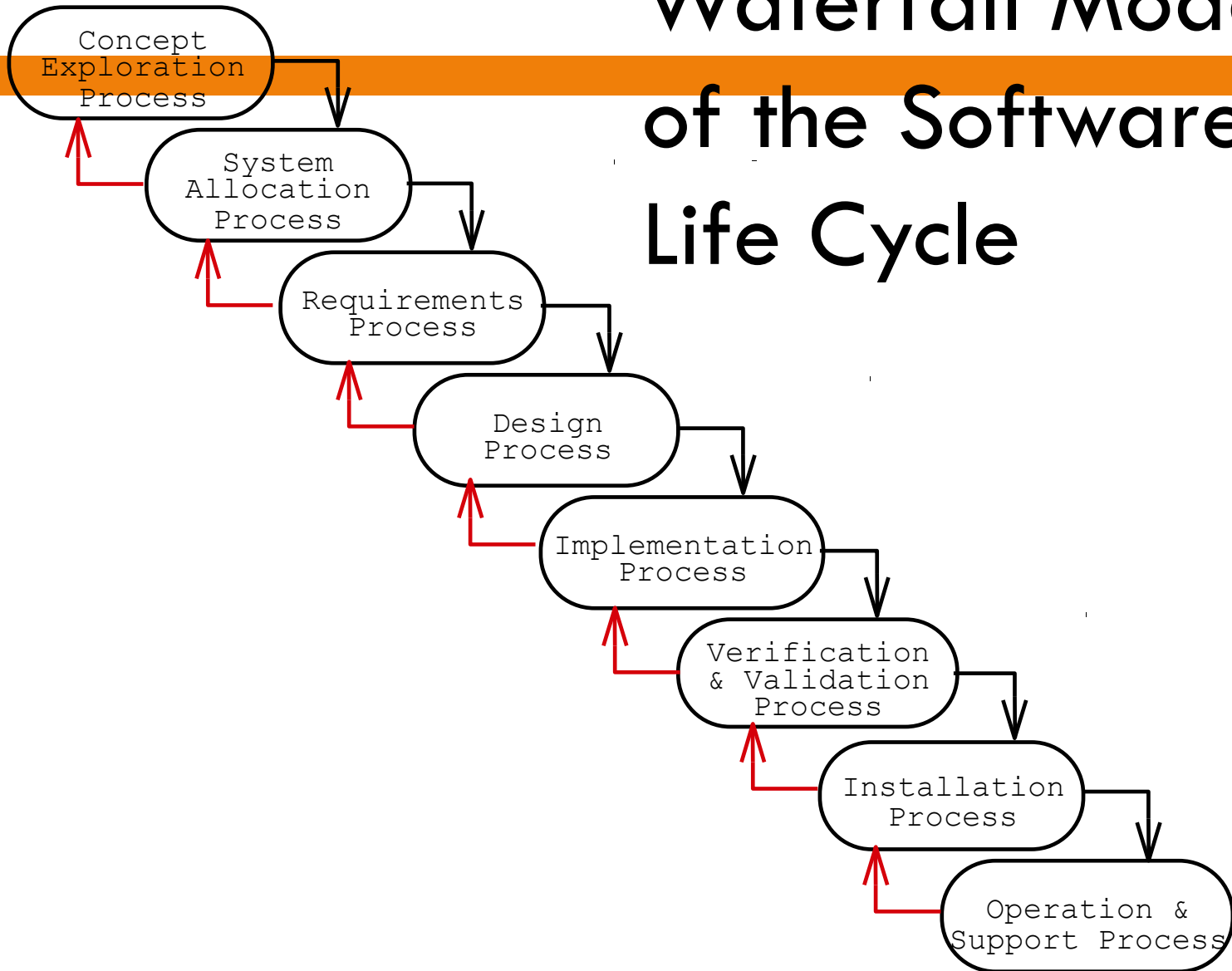


adapted from [Royce 1970]

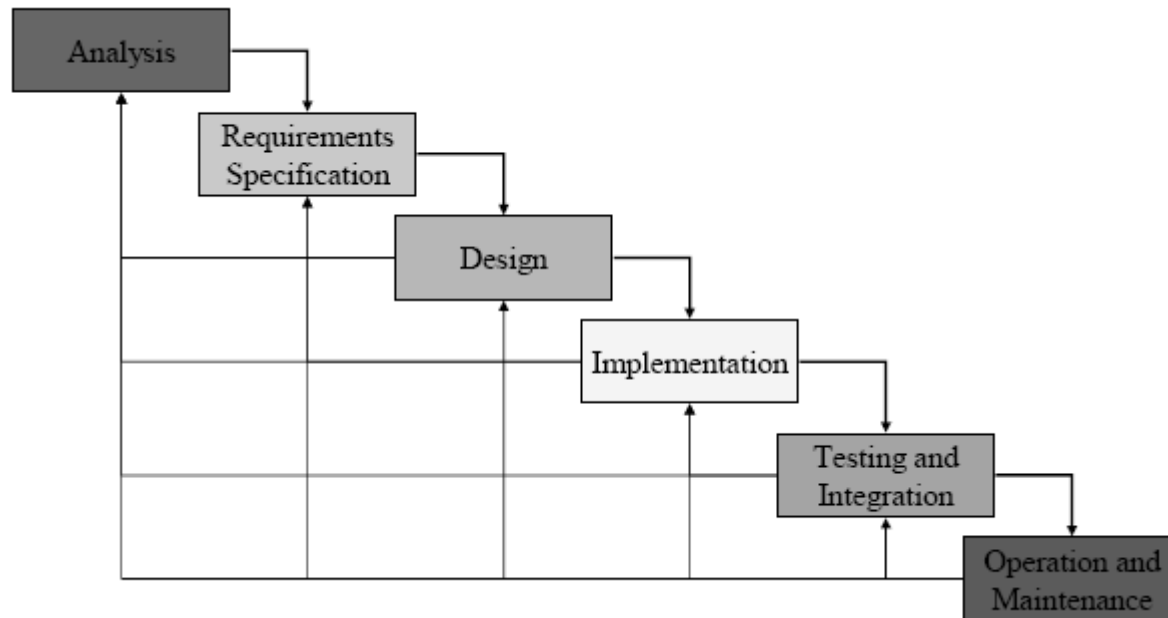
Problems with Waterfall Model

- Managers love waterfall models:
 - Nice milestones
 - No need to look back (linear system), one activity at a time
 - Easy to check progress : 90% coded, 20% tested
- Different stakeholders need different abstractions
 - => V-Model
- Software development is iterative
 - During design problems with requirements are identified
 - During coding, design and requirement problems are found
 - During testing, coding, design& requirement errors are found
 - => iterative models
 - Iterative Waterfall
 - Spiral Model
- System development is a nonlinear activity
 - => Issue-Based Model

The Iterative Waterfall Model of the Software Life Cycle



Iterative Waterfall model- A Reality



Waterfall Model

□ Advantages

- ▣ Built-in documentation and quality assurance at each stage
- ▣ Standard procedure and techniques
- ▣ Better control over development
- ▣ Feedback statistics for future projects

□ Disadvantages

- ▣ Heavy demand on client/users in testing each stage
- ▣ Focus on documentation as the end product
- ▣ Reluctance of clients to commit to sign-off
- ▣ Reluctance of developer to alter signed off requirements
- ▣ Inability of clients/users to understand what is being proposed

Prototyping

- Build a 'limited function' model to illustrate design
- Identify initial/known requirements
- Quick design
- Build prototype
- Evaluation by client and refinement of the requirements
- Iterate

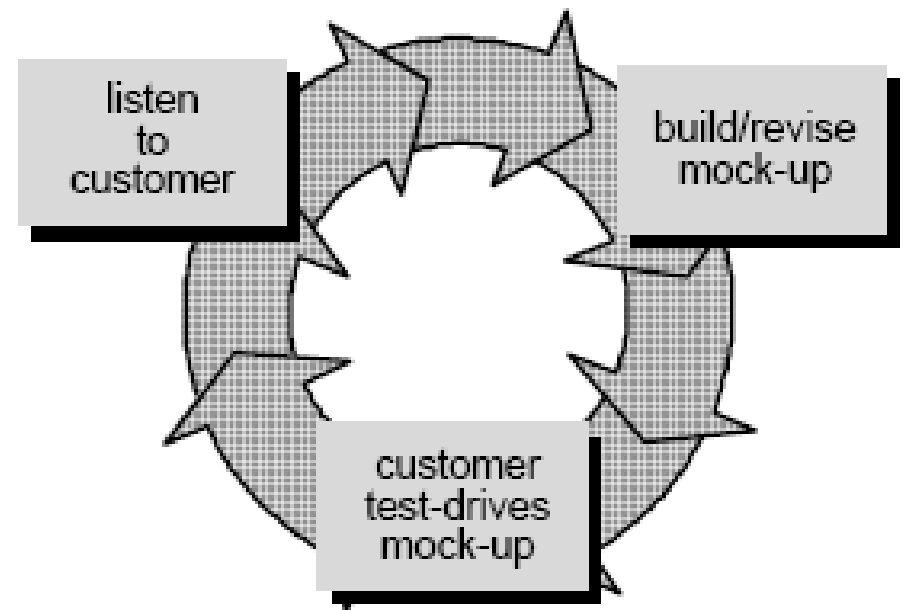


Figure © Pressman Associates

Prototyping

□ Advantages

- ▣ Users/Clients “see and feel” the proposed product before it is fully implemented
- ▣ Users get an earlier sense of something happening
- ▣ Developers get a better feel for the design
- ▣ Developers work closer with users in design

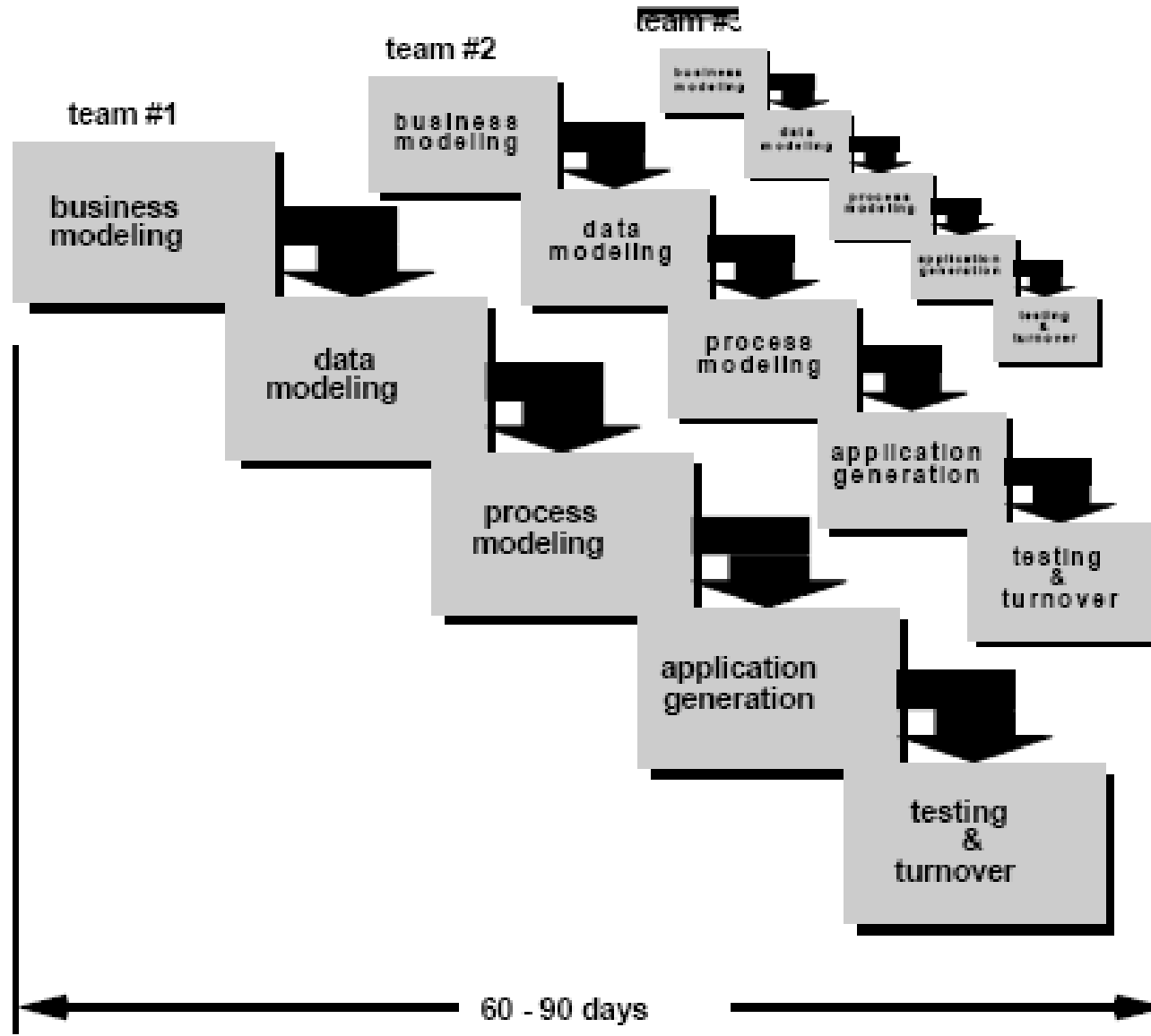
□ Disadvantages

- ▣ User expectation are for early product delivery
- ▣ Temptation for developers to simply “beef-up” the prototype.
- ▣ Prototype may encourage excessive user requests based on minor design issues

The RAD Model

- ❑ Rapid Application Development - James Martin (1991)
- ❑ A linear sequential software development process that emphasizes an extremely short development cycle.
- ❑ Rapid development is achieved by using a component-based construction approach.
- ❑ Used primarily for information systems applications.
- ❑ RAD projects are typically staffed with small integrated teams comprised of developers and end users.

The RAD Model



The RAD Model

- Main Philosophy – “Survival of the Fastest”
- “I have been a software engineer for more than 20 years, and schedules, market windows, and release dates have dominated my work process, my technical designs, and my life. This is generally true of my colleagues as well”. Neil C. Olsen (IEEE Software 9/95)
- When it works
 - ▣ The application will be run standalone.
 - ▣ Major use can be made of preexisting class libraries (APIs).
 - ▣ Performance and reliability is not critical.
 - ▣ System can be split into several independent modules.
 - ▣ The required technology is more than a year old.

The Incremental Model

- Linear sequential + Iteration
- Same as Waterfall Model until general design stage when modules are identified
- Each module is delivered incrementally.
- For each module
 - ▣ Detailed design
 - ▣ Module implementation
 - ▣ Module integration

The Incremental Model

□ Advantages

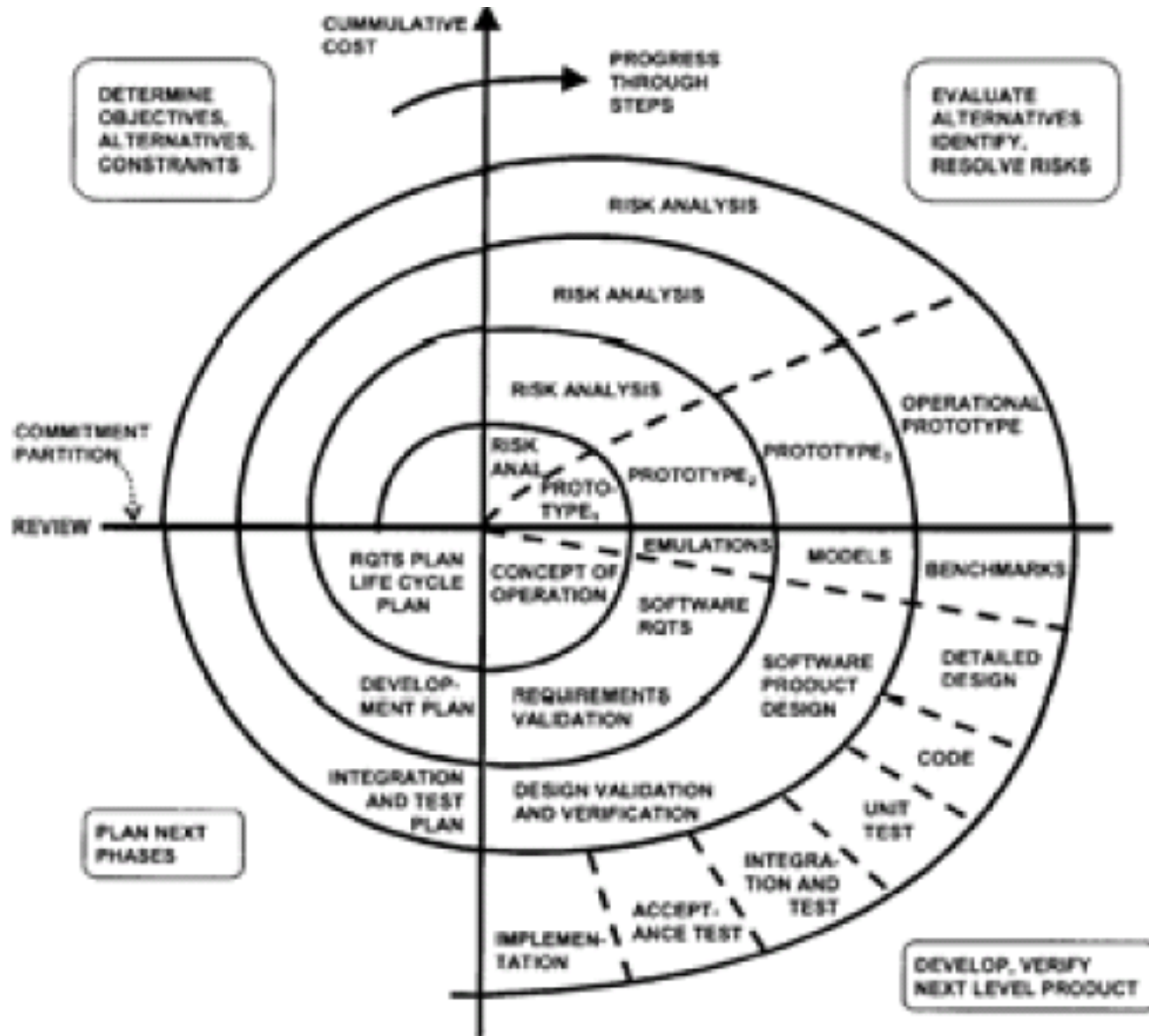
- Operational quality portion of product within weeks
- Users have earlier use of some components of the system
- Changes due to new system are easier to manage piece by piece
- Less staff required
- Smaller capital outlay, rapid return on investment
- Client can make progress payment

□ Disadvantages

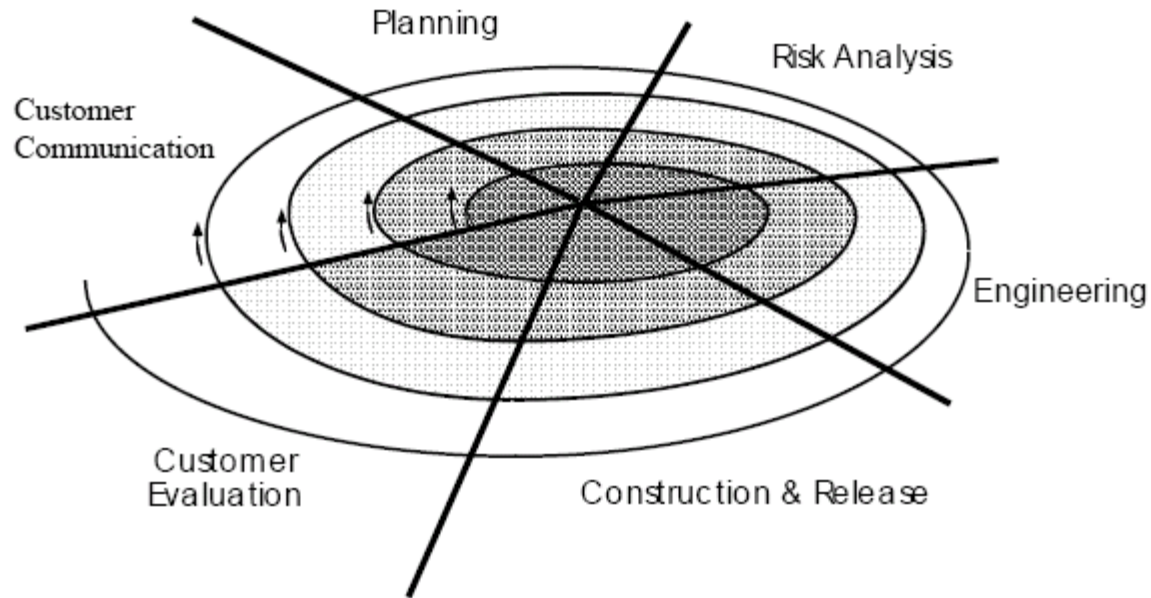
- Each build delivery requires total integration testing
- System must be incremental in nature
- Maintenance on delivered units may affect integration of new units
- May turn into Build and Fix

Spiral Model

- Original Model (1988)
- Four sectors



Spiral Model



Spiral Model

- The spiral model discussed in this section is a variation on the model proposed by Boehm.
- The spiral model originally proposed by Boehm, is an evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of linear sequential model.
- It provides the potential ,for rapid development
- The spiral model is intended for large, expensive and complicated projects.

- The steps in the spiral model can be generalized as follows:
- The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:
 - evaluating the first prototype in terms of its strengths, weaknesses, and risks;
 - defining the requirements of the second prototype;
 - planning and designing the second prototype;
 - constructing and testing the second prototype.

Spiral Model

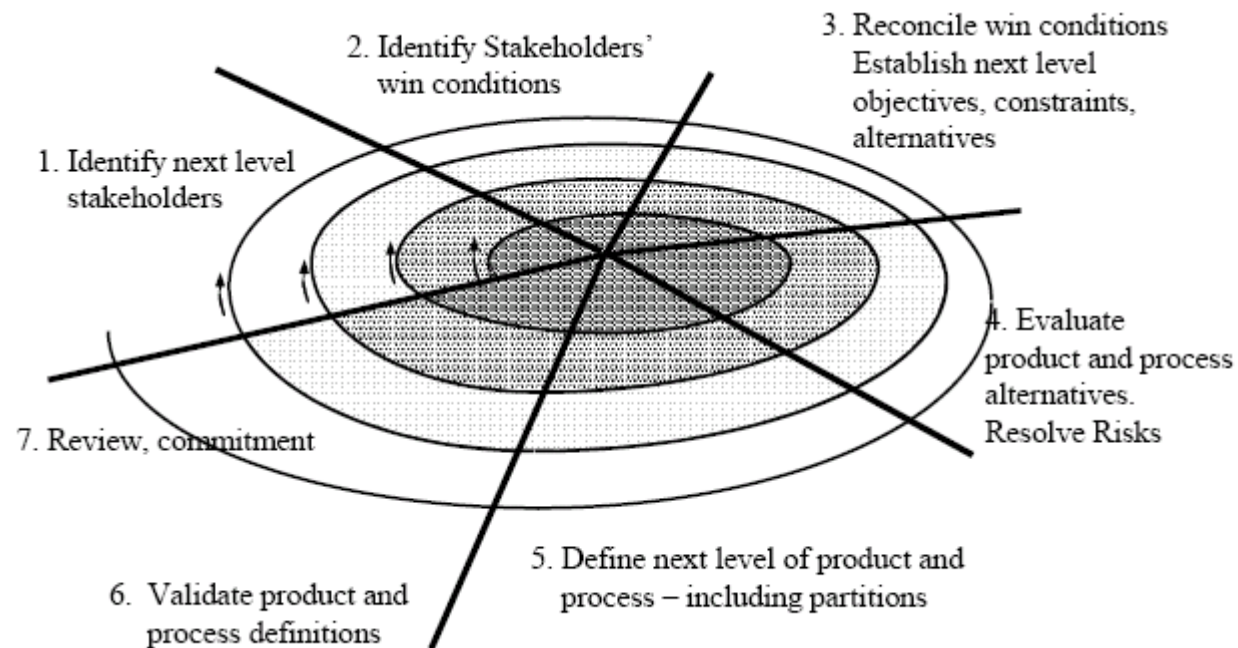
- SE team moves around the spiral beginning at the center.
- And began each cycle of the spiral by performing the next level of elaboration of the perspective system's
 - ▣ objectives,
 - ▣ constraints, and
 - ▣ Alternatives
- Cyclic approach
 - ▣ Incrementally growing a system's degree of definition and implementation while decreasing its degree of risk
- Spiral Model makes an assumption
 - ▣ Objectives, constraints, and alternatives are given or known

- ❑ It is not a cure-all / solution to all the problems.
- ❑ The spiral model is used most often in large projects.
For smaller projects, the concept of agile software development is becoming a viable alternative. The US military has adopted the spiral model for its Future Combat Systems program.

Win-Win Spiral Model

- ❑ Boehm & Bose 1994
- ❑ Initially called Next Generation Process Model
- ❑ Win-Win model answers these question
 - ▣ Where do the next-level objectives and constraints come from?
 - ▣ How do you know they are the right ones?
- ❑ Spiral model + Theory W
- ❑ Theory W approach - developed by Boehm & Ross 1989
 - ▣ Involves identifying the stakeholders and their win conditions
 - ▣ Uses negotiation processes to determine a mutually satisfactory set of objectives, constraints, and alternatives for the stakeholders

Win-Win Spiral Model



Win-Win Spiral – milestones

- Anchor Point milestones
 - ▣ Drive the spiral to progress toward completion
 - ▣ Offer a means to compare progress one spiral project and another
- They are
 - ▣ Life Cycle Objectives (LCO)
 - ▣ Life Cycle Architecture (LCA)
 - ▣ Initial Operational Capability (IOC)

Win-Win Spiral – milestones

- Life Cycle Objectives:
 - defines a set of objectives for each major software engineering activity.
 - For example as part of LCO, a set of objectives establishes the definition of top level system/product requirements.

Win-Win Spiral – milestones

□ Life Cycle Objectives

□ System and Software Architecture

- Architecture's feasibility in supporting the systems' objectives and requirements

□ Life Cycle Plan

- Identification of major stakeholders
- Identification of the process model(s)
- WWWWWHH Principle
 - Why is the system being developed?
 - What will be done by When?
 - Who is responsible for a function?
 - Where are they organizationally located?
 - How will the job be done, technically & managerially?
 - How much of each resource is necessary?

Win-Win Spiral – milestones

- **Life Cycle Architecture (LCA):** establishes the objectives that must be met as the system and software architecture is defined.
 - Definition of the system and software architecture
 - The specifics of COTS and reused software choice
 - The specifics of quality attribute levels
 - Identification of likely direction of architectural evolution
 - The stakeholders' concurrence that the LCA elements are compatible with their objectives for the system
- **Initial Operational Capability (IOC)** represents set of objectives associated with the following:
 - Software preparation
 - Documentation, data conversion, licenses
 - Site preparation prior to installations
 - Facilities, equipment, supplies
 - User, operator and maintainer preparation
 - Selection, teambuilding, training


Win-Win/Spiral Model

□ Advantages

- ▣ Risk Analysis forces study of alternatives before each stage
- ▣ Risk Analysis identifies costs of each stage based on development so far
- ▣ Project can be abandoned before it wastes more money

□ Disadvantages

- ▣ Applies only to in-house development
- ▣ Cost effective only for very large projects



Q&A