UCS1405 - SOFTWARE ENGINEERING

Life Cycle Models

Dr.K.Madheswari
Associate Professor
CSE

Learning Objective

- To give an introduction to process models.
- To learn the different types of processmodels.
- To analyze different models and in turn perform comparative study of the models

Overview

- Prescriptive Models
  - Waterfall Model
  - Incremental Model
  - RAD
  - Evolutionary models
- Specialized process models

Prescriptive Models

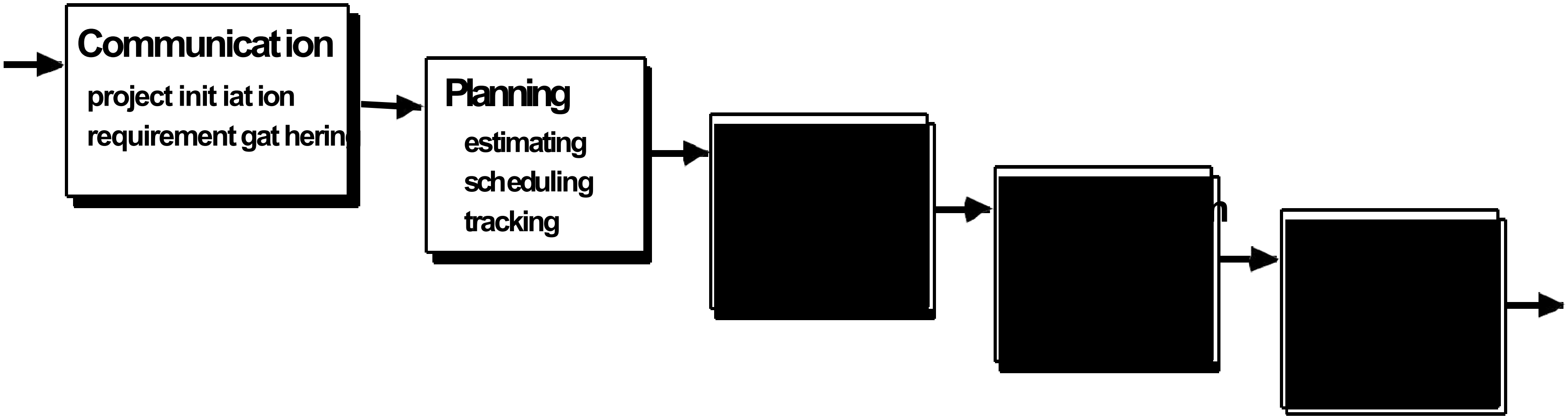
- Prescriptive process models advocate an orderly approach to software engineering

That leads to a few questions …

- If prescriptive process models strive for structure and order, are they inappropriate for a software world that thrives on change?
- Yet, if we reject traditional process models (and the order they imply) and replace them with something less structured, do we make it impossible to achieve coordination and coherence in software work?

- A prescriptive process model populates a process framework with explicit task sets for software engineering actions.

The Waterfall Model



Communicat ion

project init iat ion

requirement gat herin

Planning

estimating

scheduling

tracking

[redacted]

[redacted]

[redacted]

The Waterfall Model

- Also known as **Classical life cycle or linear sequential model**.
- The oldest and the most widely used software engineering paradigm.
- A systematic, sequential approach to software development that begins at the system level and progresses through planning, modeling, construction and deployment, culminating in on-going support of the completed software.

Problems in the Waterfall Model

- Real projects rarely follow the sequential flow that the model proposes. Although the linear model can accommodate iteration, it does so indirectly. As a result, changes can cause confusion as the project team proceeds.
- It is often difficult for the customer to state all requirements explicitly.
- The customer must have patience. A working version of the program(s) will not be available until late in the project time-span. A major blunder, if undetected until the working program is reviewed, can be disastrous.
- leads to "blocking states" - some project team members must wait for other members of the team to complete dependent tasks.
- The time spent waiting can exceed the time spent on productive work!

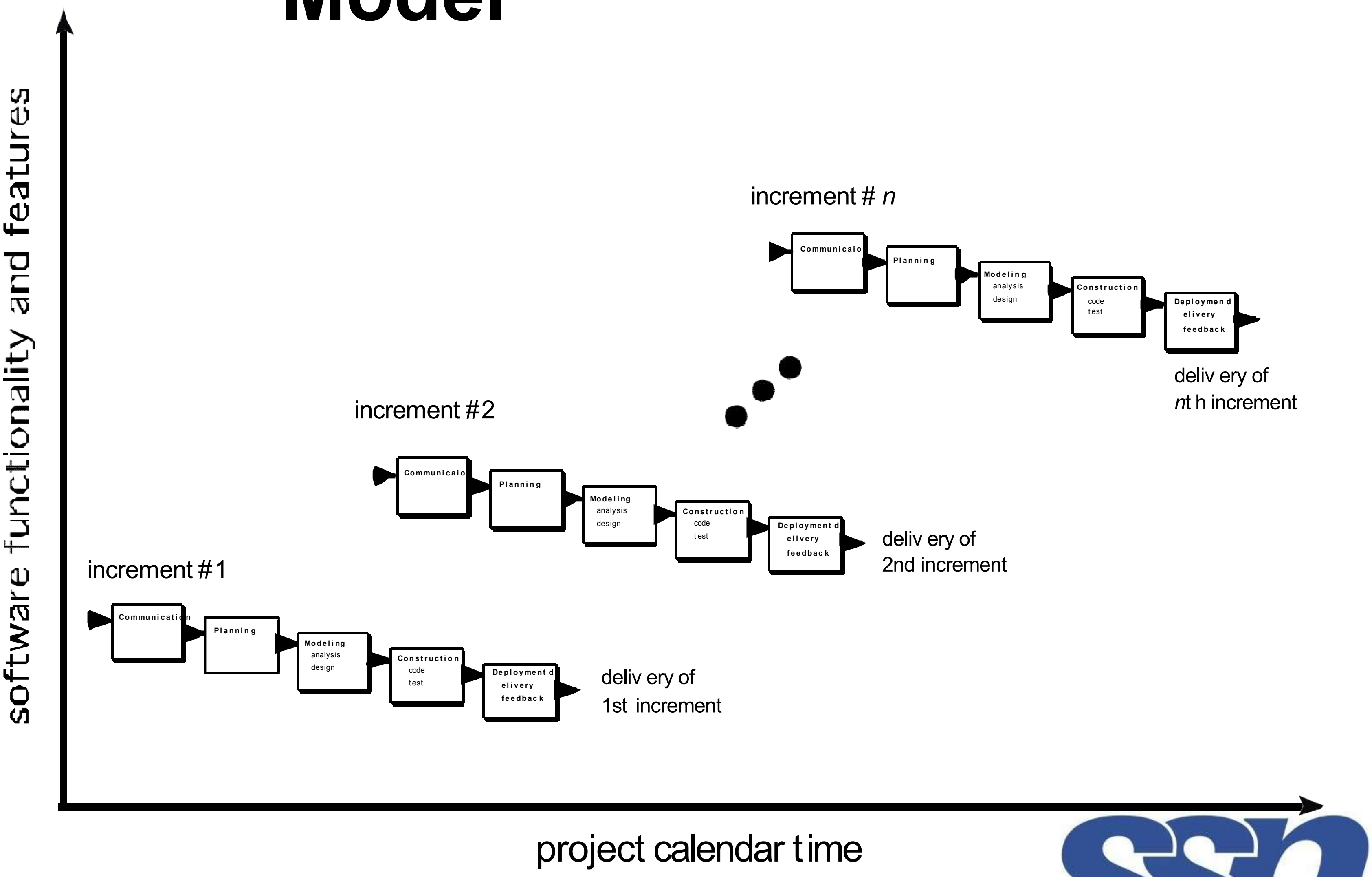
Where to use the waterfall model?

- In situations where requirements are fixed and work is to proceed to completion in a linear manner.
- Large scale projects where requirements are well defined

Key points

- It provides a template into which methods for analysis, design, coding, testing, and support can be placed.
- The classic life cycle remains a widely used procedural model for software engineering.
- While it does have weaknesses, it is significantly better than a haphazard approach to software development.

The Incremental Model



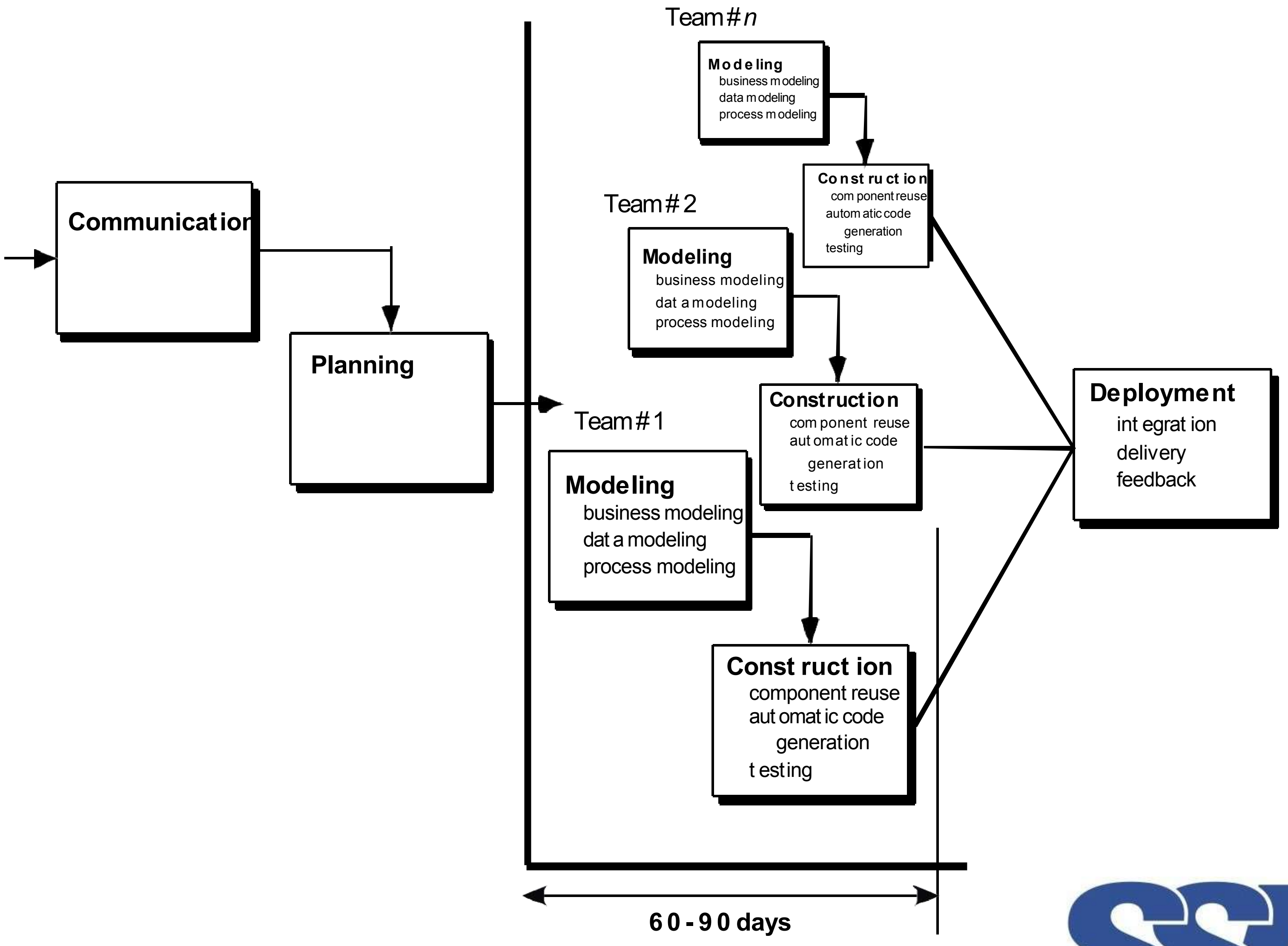
The Incremental Model

- It combines the elements of waterfall model applied in an iterative fashion.
- Each linear sequence produces deliverable “increments” of the software.
- The first increment is often a core product.

Where to use the Incremental model?

- In situations when staffing is unavailable for a complete implementation by the business deadline that has been established for the project.

The RAD Model



The RAD Model

- Rapid application development (RAD) is an incremental software development process model that emphasizes a short development cycle.
- The RAD model is a “high-speed” adaptation of the linear sequential model in which rapid development is achieved by using component-based construction.
- If requirements are well understood and project scope is constrained, the RAD process enables a development team to create a “fully functional system” within very short time periods (e.g., 60 to 90 days).
- Used primarily for information systems applications.

The RAD Model (cont..)

Business modeling:

•The information flow among business functions is modeled in a
way that answers the following questions:

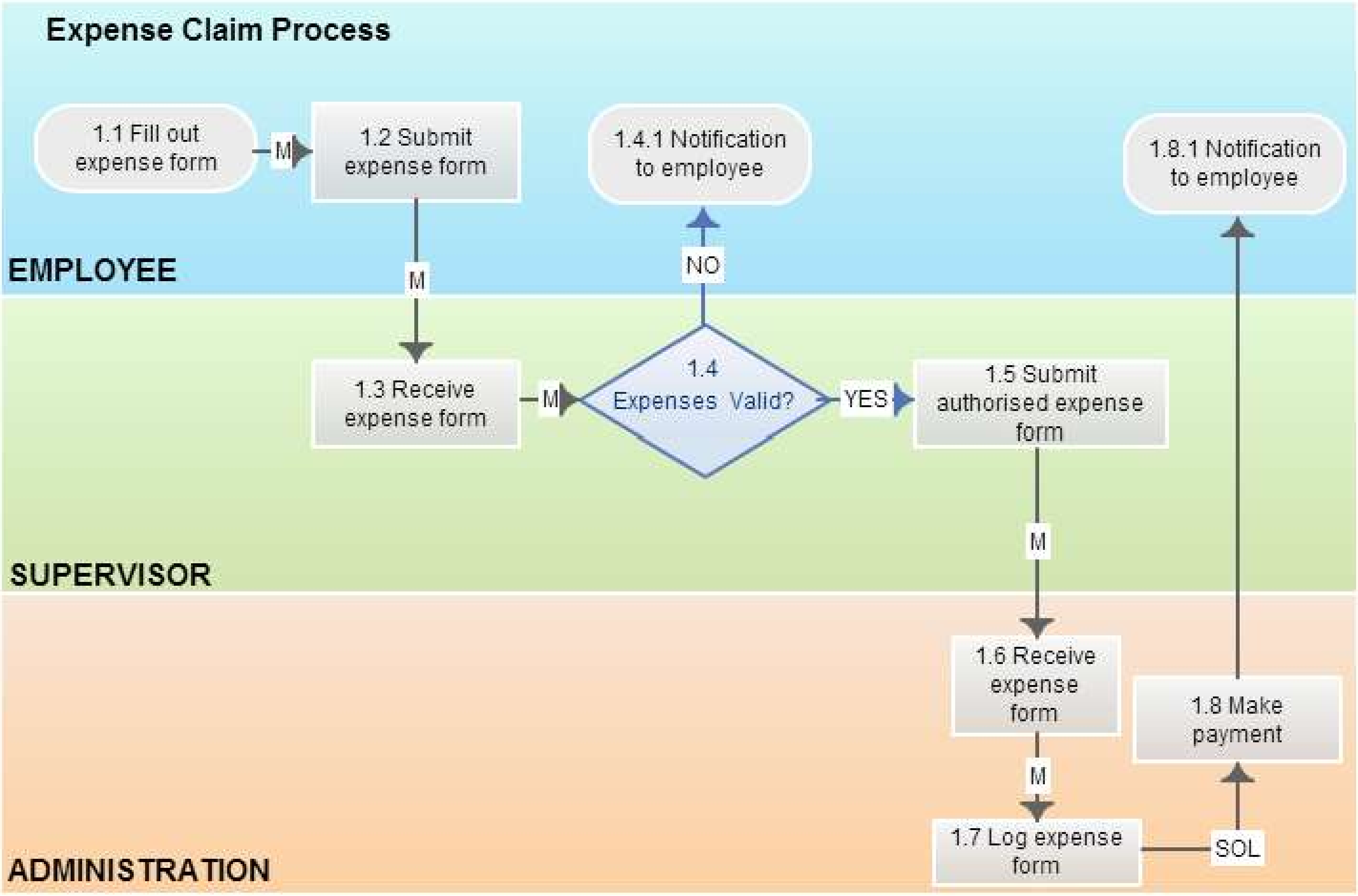
- What information drives the business process?
- What information is generated?
- Who generates it?
- Where does the information go?
- Who processes it?

Data modeling:

•The information flow defined as part of the business modeling phase is refined into a set of data objects that are needed to support the business.

•The characteristics (called *attributes*) of each object are identified and the relationships between these objects defined.

Business modeling



M - Manual
Sol - Facilitated by Solution

Process modeling:

- The **data objects** defined in the data modeling phase are transformed to achieve the information flow necessary to implement a business function.
- Processing descriptions are created for adding, modifying, deleting, or retrieving a data object.

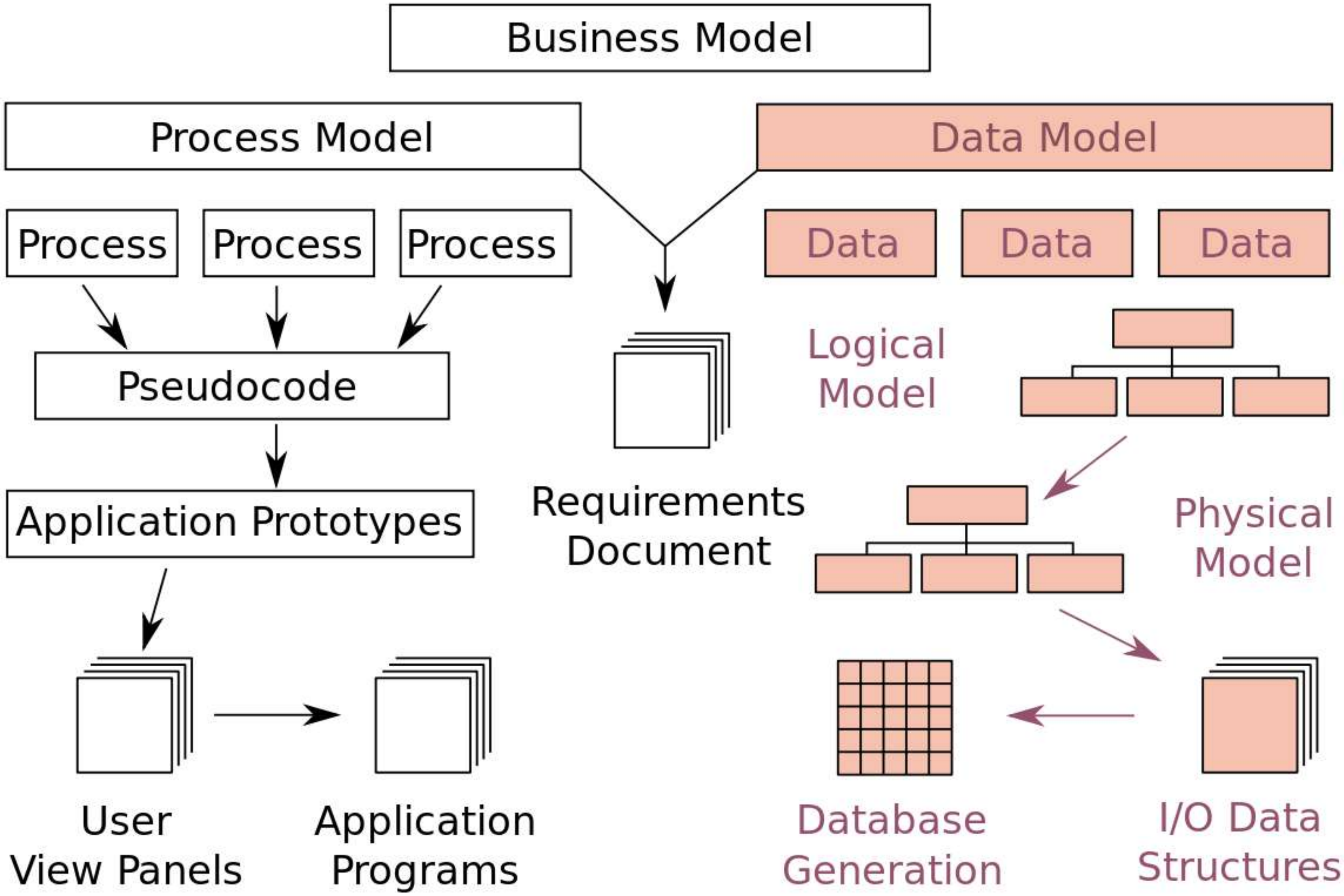
Construction:

- RAD assumes the use of fourth generation techniques
- Rather than creating software using conventional third generation programming languages the RAD process works to reuse existing program components (when possible) or create reusable components (when necessary).
- In all cases, automated tools are used to facilitate construction of the software.

Testing and turnover:

- Since the RAD process emphasizes reuse, many of the program components have already been tested.
- This reduces overall testing time.

Business Model Integration



Business Model

Process Model

Data Model

Process

Process

Process

Pseudocode

Application Prototypes

User View Panels

Application Programs

Requirements Document

Data

Data

Data

Logical Model

Physical Model

Database Generation

I/O Data Structures

•For large but scalable projects, RAD requires sufficient human resources to create the right number of RAD teams.

•RAD requires developers and customers who are committed to the rapid-fire activities necessary to get a system complete in a much abbreviated time frame. If commitment is lacking from either constituency, RAD projects will fail.

•Not all types of applications are appropriate for RAD. If a system cannot be properly modularized, building the components necessary for RAD will be problematic.

•If high performance is an issue and performance is to be achieved through tuning the interfaces to system components, the RAD approach may not work.

•RAD is not appropriate when technical risks are high. This occurs when a new application makes heavy use of new technology or when the new software requires a high degree of interoperability with existing computer programs.

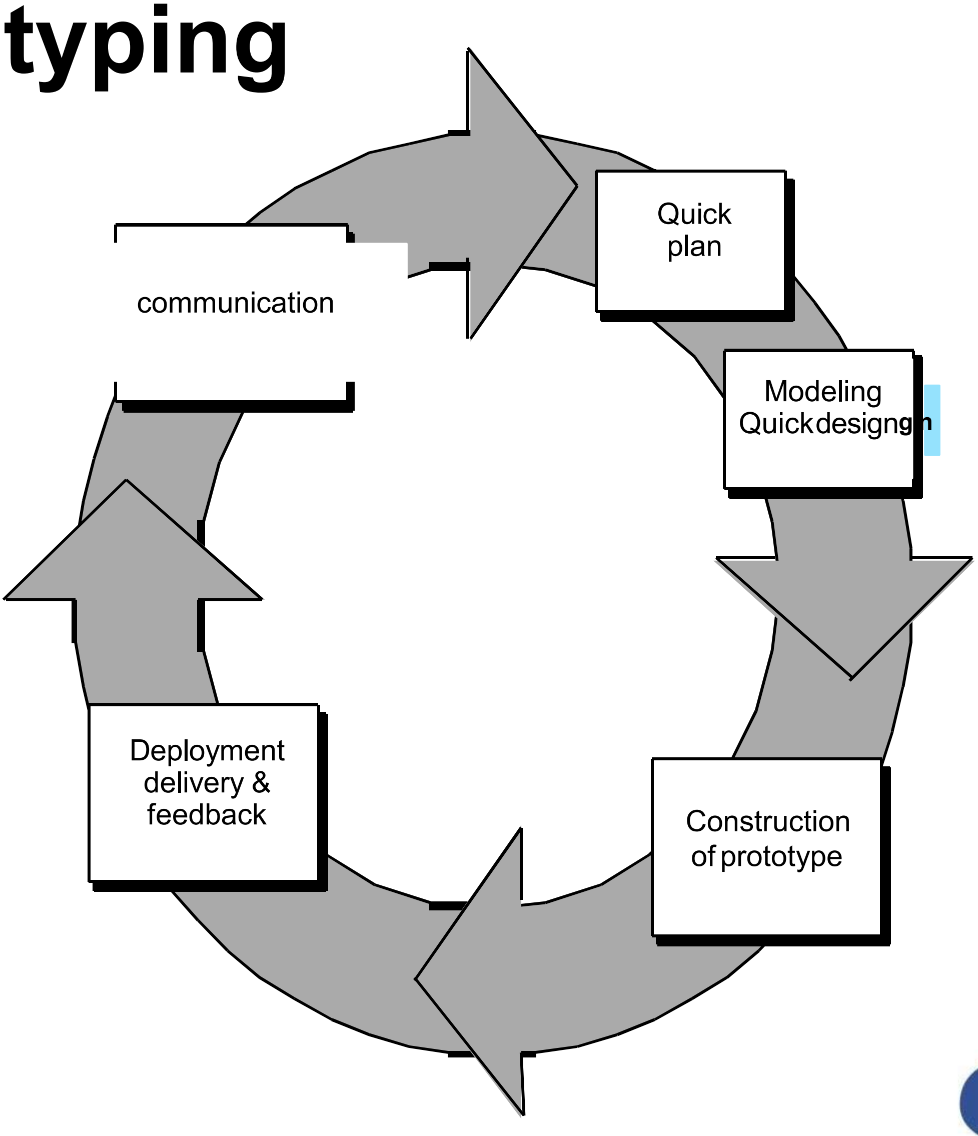
Where to use the RAD model?

Business product with unclear requirements

Need for prototyping

- The customer defines a set of general objectives for software but does not identify detailed input, processing, or output requirements.
- In other cases, the developer may be unsure of
  - The efficiency of an algorithm,
  - The adaptability of an operating system, or
  - The form that human/machine interaction should take.
- A *prototyping paradigm* may offer the best approach.
- It serves as a mechanism for identifying software requirements.
- It serves as a first system.

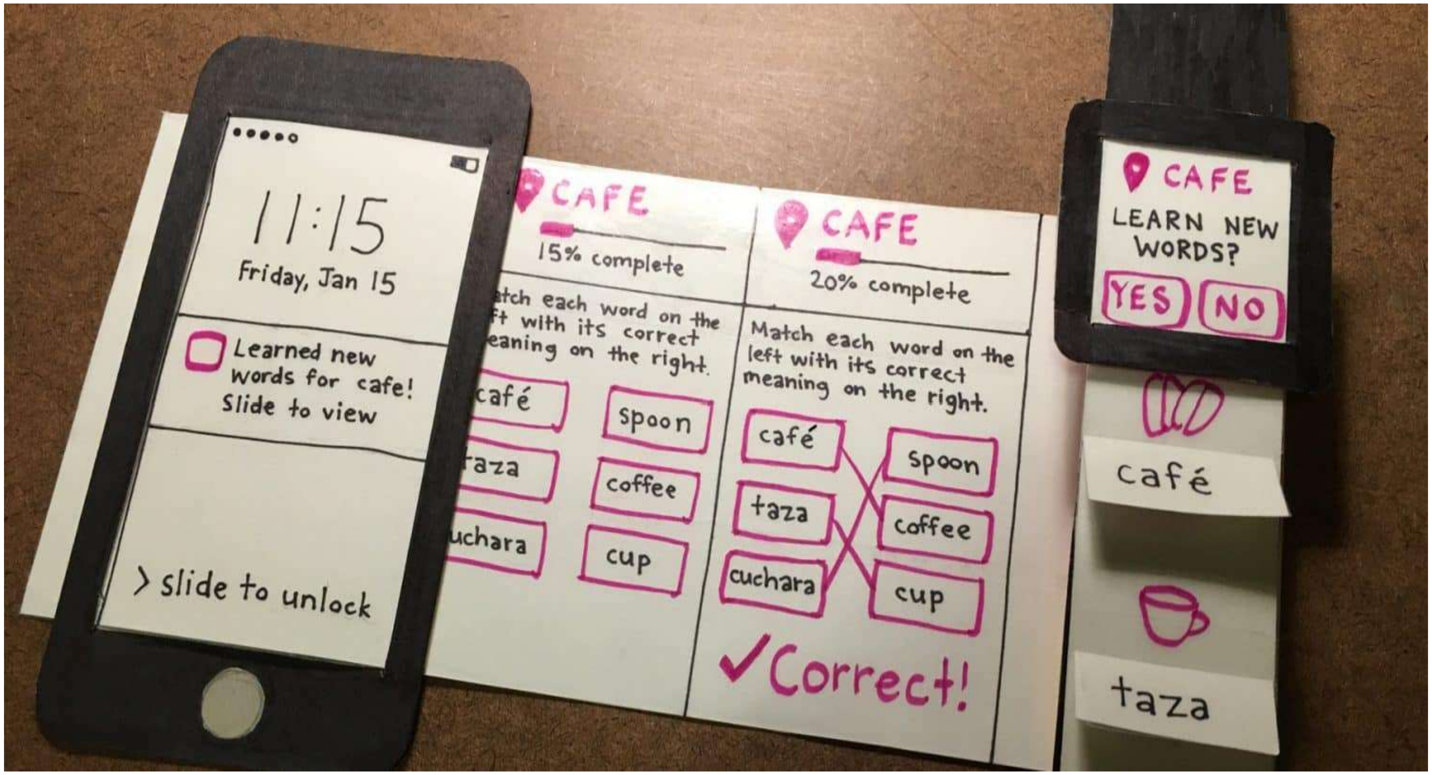
Evolutionary Models: Prototyping



The prototyping paradigm begins with

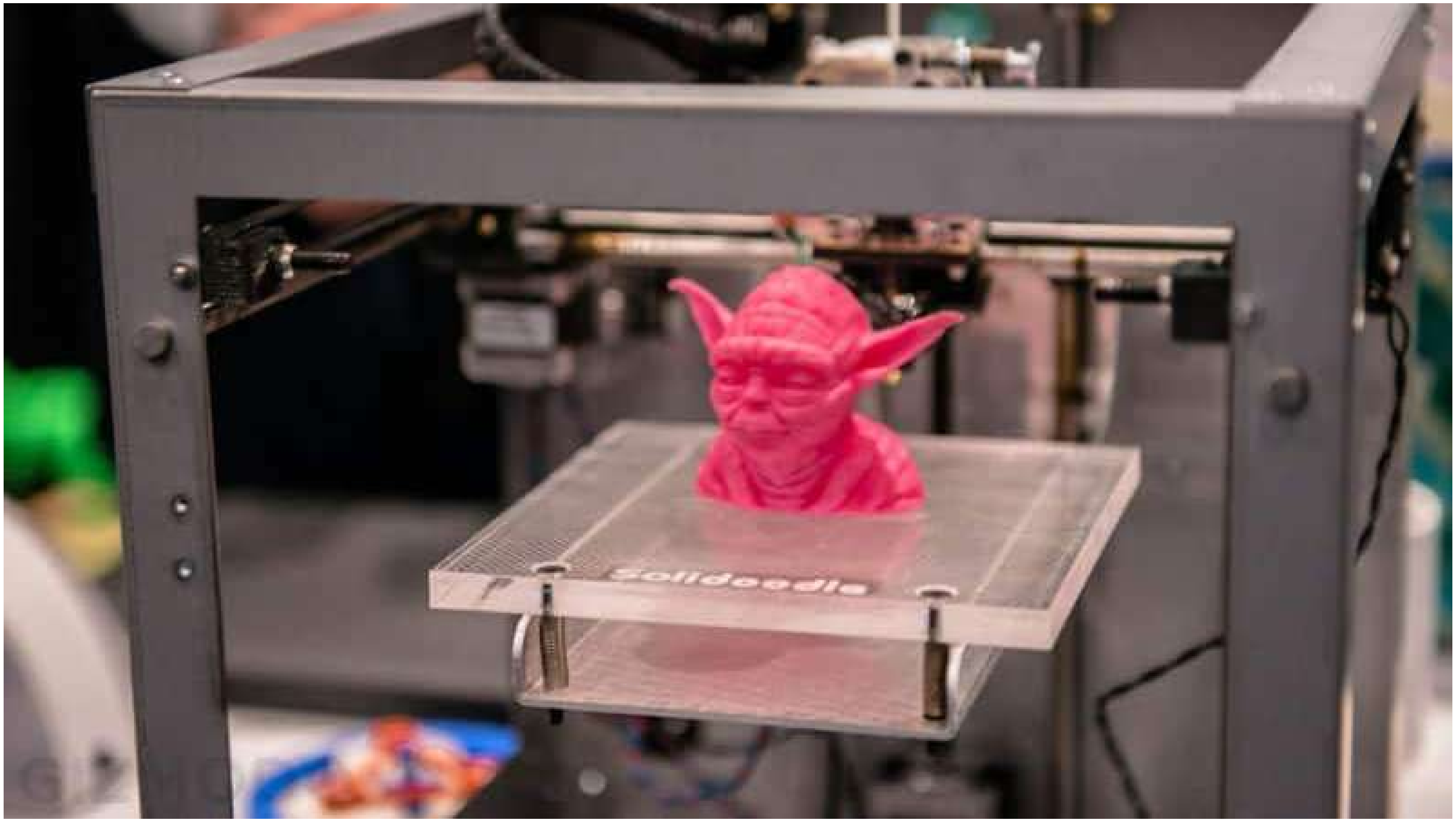
- **Requirements gathering**:
  - Developer and customer meet and define the overall objectives for the Software,
  - Identify whatever requirements are known,
  - outline areas where further definition is mandatory.
- **Quick design:**.
  - The quick design focuses on a representation of those aspects of the software that will be visible to the customer/user (e.g., input approaches and output formats).
- The quick design leads to the construction of a prototype. The **prototype** is evaluated by the customer/user and used to refine requirements for the software to be developed.
- Iteration occurs as the prototype is tuned to satisfy the needs of the customer, while at the same time enabling the developer to better understand what needs to be done.

Example for prototyping-paper Prototype

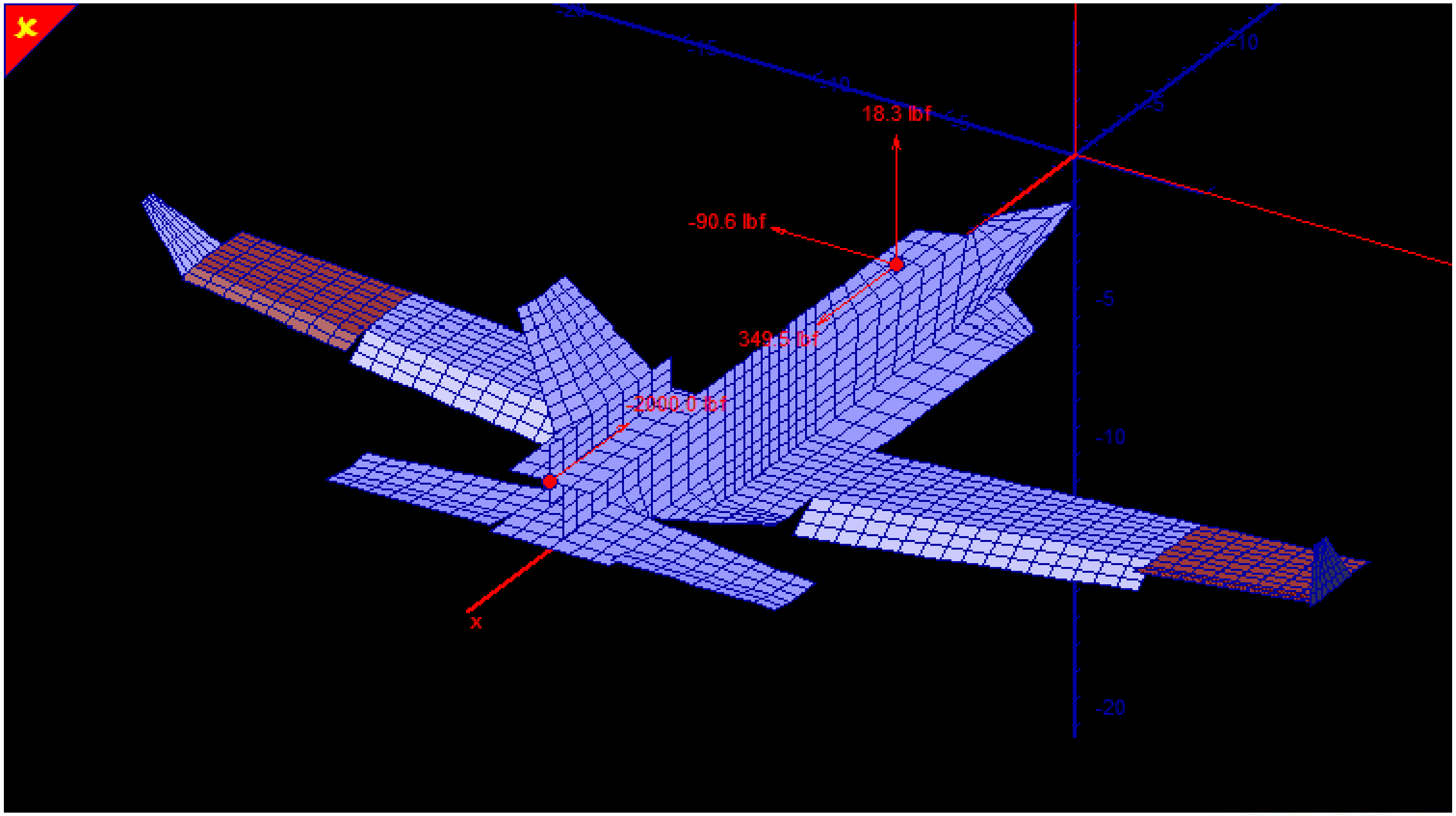


11:15
Friday, Jan 15
Learned new words for cafe! Slide to view
> slide to unlock
CAFE
15% complete
Match each word on the left with its correct meaning on the right.
café
taza
cuchara
spoon
coffee
cup
CAFE
20% complete
Match each word on the left with its correct meaning on the right.
café
taza
cuchara
spoon
coffee
cup
✓Correct!
CAFE
LEARN NEW WORDS?
YES
NO
café
taza

Example for prototyping- 3D printing



Example for prototyping- Digital Prototype



Example for prototyping- Scale Model

Problems in prototyping

- The customers sees what appears to be a working version of the software, unaware that the prototype is held together “ with chewing gum and baling wire” .
- The developer often makes implementation compromises in order to get a prototype working quickly.

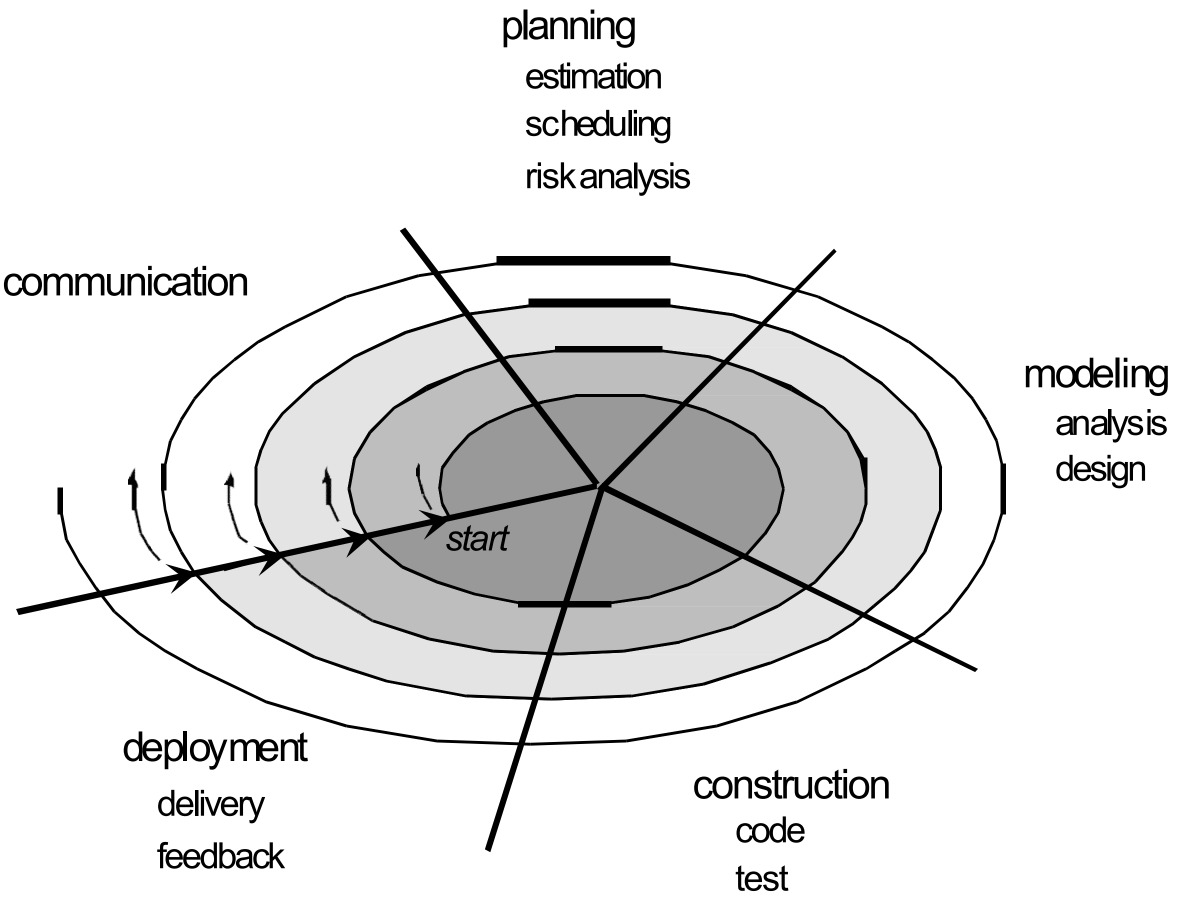
Where to use the prototyping?

Business product with unclear requirements

Example: Ecommerce

This model can be successfully used for developing user interfaces, high technology software-intensive systems, and systems with complex algorithms and interfaces.

Evolutionary Models: The Spiral



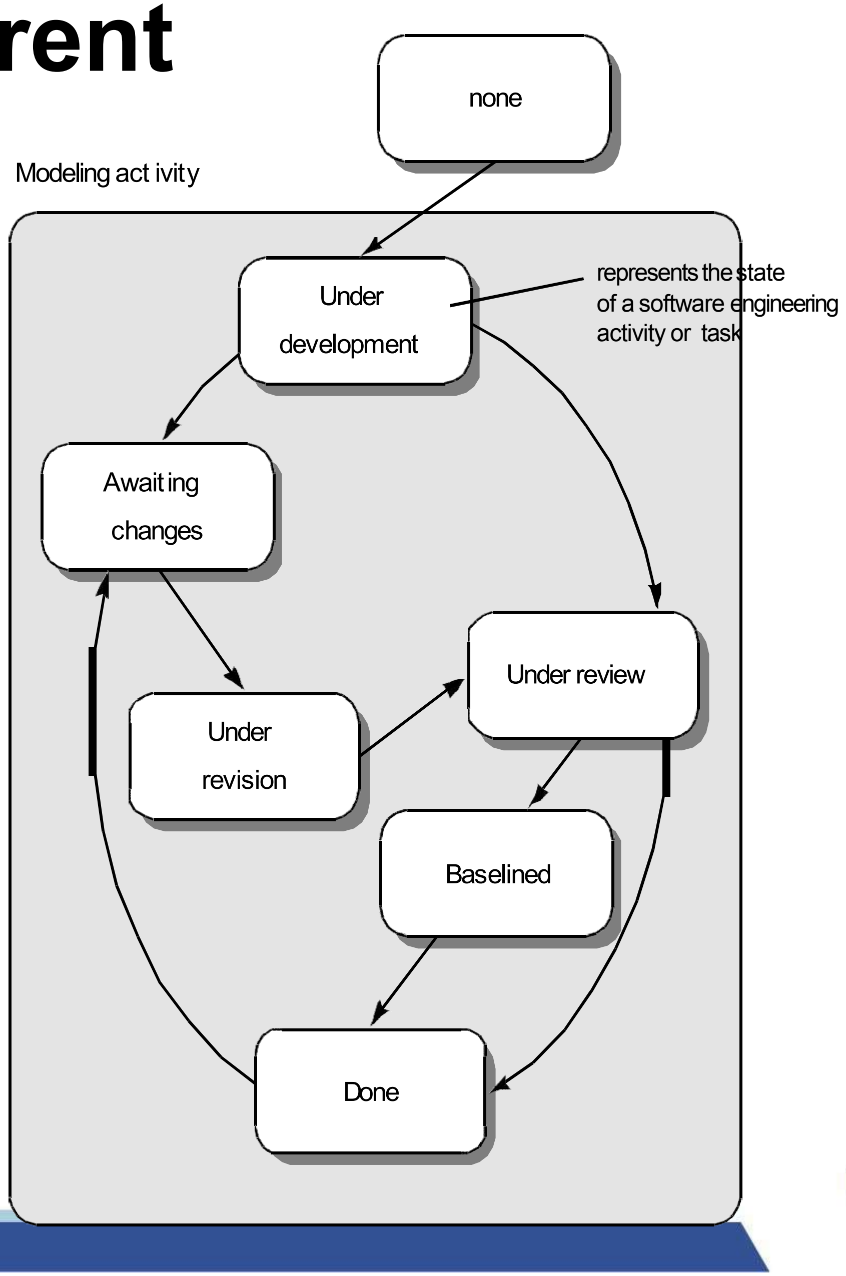
- Originally proposed by Boehm, couples iterative nature of prototyping and the systematic aspects of waterfall model
- Rapid development of versions.
- Software is developed in series of incremental releases.
- Each iteration produces a more complete product
- Better management through risk analysis.
- During early iterations, the incremental release might be a paper model or prototype.
- During later iterations, increasingly more complete versions of the engineered system are produced.
- Anchor point milestones- a combination of work products and conditions that are attained along the path of the spiral.
- The first circuit around the spiral might result in the development of a product specification.
- Subsequent passes around the spiral might be used to develop a prototype and then progressively more sophisticated versions of the software.

- May be difficult to convince customers that evolution is controllable
- Demands risk assessment expertise : major risk will cause problems if not identified
- Relatively new and not widely used, so cannot determine performance

Where to use the Spiral model?

R&D/large scale projects where privacy/security issues are involved

Evolutionary Models: Concurrent



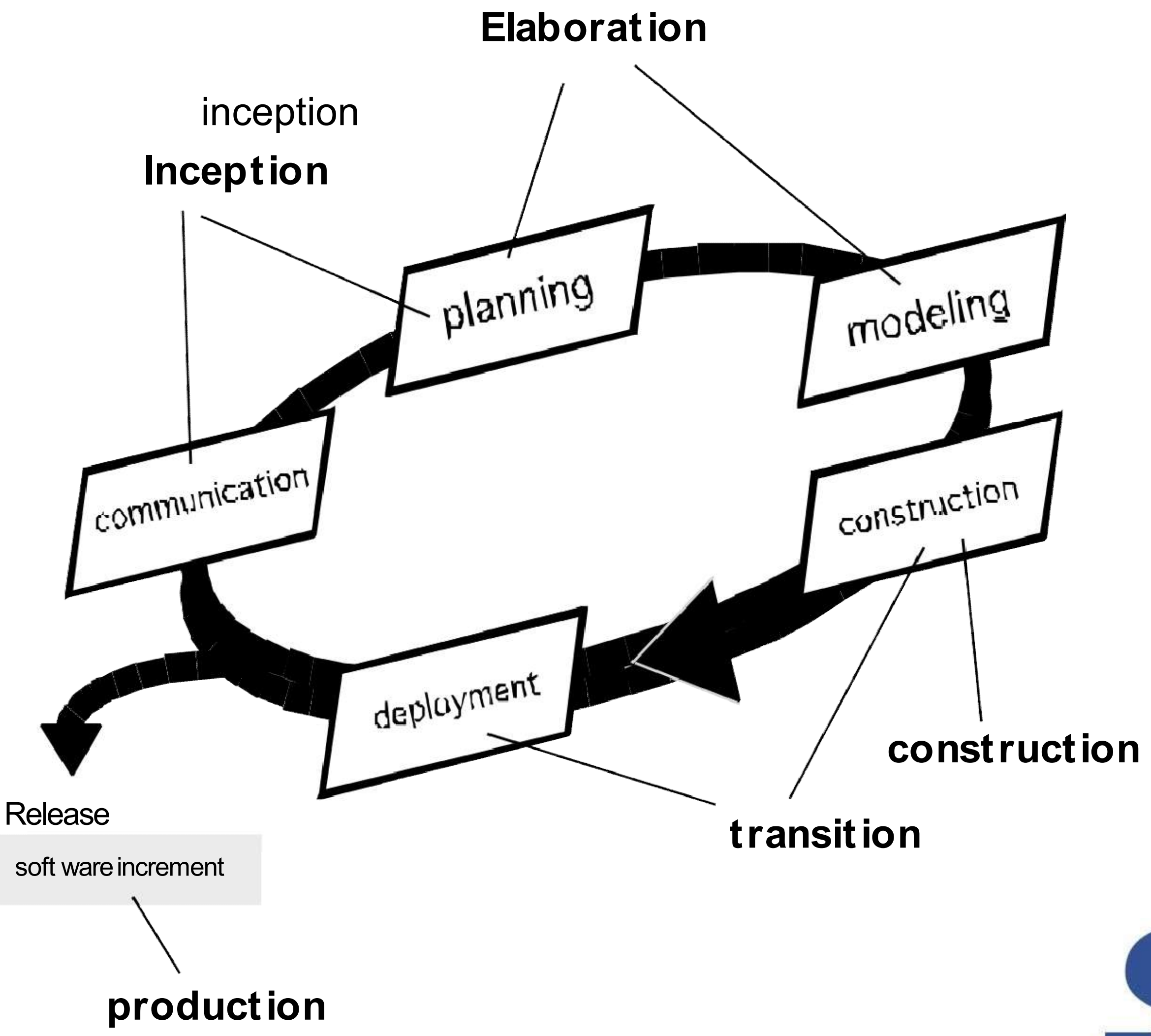
- The concurrent process model can be represented schematically as a **series** of framework activities, software engineering actions and tasks, and their associated states.
- For example, the modeling activity defined for the spiral model is accomplished by invoking the following tasks:
  - prototyping and/or analysis modeling,
  - requirements specification, and design
- For example,
  - early in a project the *customer communication* activity has completed its first iteration and exists in the **awaiting changes** state.
  - The modeling activity (which existed in the **none** state while initial customer communication was completed) now makes a transition into the **under development** state.
  - If, however, the customer indicates that changes in requirements must be made, the modeling activity moves from the **under development** state into the **awaiting changes** state.

Where to use the Concurrent process model?

Specialized Process Models

- Component based development—the process to apply when reuse is a development objective
- Formal methods—emphasizes the mathematical specification of requirements
- AOSD—provides a process and methodological approach for defining, specifying, designing, and constructing *aspects*
- Unified Process—a "use-case driven, architecture-centric, iterative and incremental" software process closely aligned with the Unified Modeling Language (UML)

The Unified Process (UP)



UP -Work Products

Inception phase

- Vision document
- Initial use-case model
- Initial project glossary
- Initial business case
- Initial risk assessment .
- Project plan,
  - phases and iterations.
- Business model,
  - if necessary.
- One or more prototypes

Elaboration phase

- Use-case model
- Supplementary requirements
  - including non-functional
- Analysis model
- Software architecture
  - Description.
- Executable architectural
  - prototype.
- Preliminary design model
- Revised risk list
- Project plan including
  - iteration plan
  - adaptedworkflows
  - milestones
  - technical work products
- Preliminary user manual

Construction phase

- Design model
- Software components
- Integrated software
  - increment
- Test plan and procedure
- Test cases
- Support documentation
  - user manuals
  - installation manuals
  - description of current
    - increment

Transition phase

- Delivered software increment
- Beta test reports
- General user feedback

Summary

- Prescriptive Models
  - Waterfall Model
  - Incremental Model
  - RAD
  - Evolutionary models
- Specialized process models

Check your understanding

- When do we use “formal methods’?
- What process model should be used when requirements are fixed?

Reference

- Roger S Pressman, “Software Engineering – A Practitioner ’s Approach”, McGraw-Hill International Edition, Seventh Edition, 2010.

Thank you