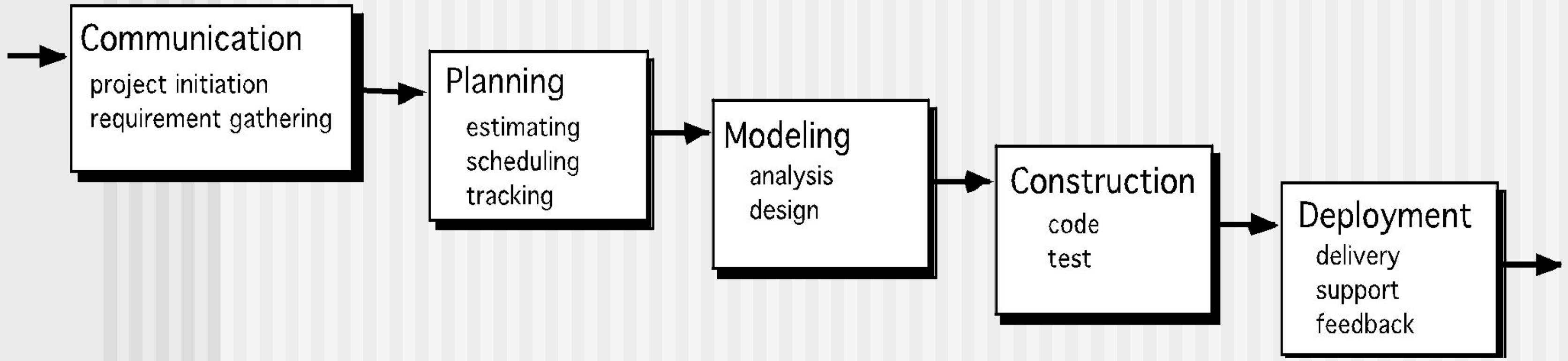
Chapter 2

- **Process Models**

Prescriptive Models

- Prescriptive process models advocate an orderly approach to software engineering i.e Process flow determines how activities, actions and tasks are arranged with respect to sequence and time.
- Linear Process Flow - Executes each activity in sequence.
- Iterative Process Flow - Repeats one or more activities before starting next.
- Evolutionary process flow - Carry out activities in a circular way.
- Parallel process flow - Execute one or more activities in parallel with each other

The Waterfall Model



It is the oldest paradigm for SE. When requirements are well defined and reasonably stable, it leads to a linear fashion.

(problems: 1. rarely linear, iteration needed. 2. hard to state all requirements explicitly. Blocking state. 3. code will not be released until very late.)

The classic life cycle suggests a systematic, sequential approach to software development.

Waterfall….

- Linear sequence of stages/phases
  - Requirements -> Analysis –> D –> Code –> Test –> Deploy
- A phase starts only when the previous has completed; no feedback!
- The phases partition the project, each addressing a separate concern

Waterfall…

- Linear ordering implies each phase should have some output
- The output must be validated/certified
- Outputs of earlier phases: work products
- Common outputs of a waterfall: SRS, project plan, design docs, test plan and reports, final code, supporting docs

Waterfall Advantages

- Natural approach for problem solving
- Conceptually simple, cleanly divides the problem into distinct independent phases
- Easy to administer in a contractual setup – each phase is a milestone

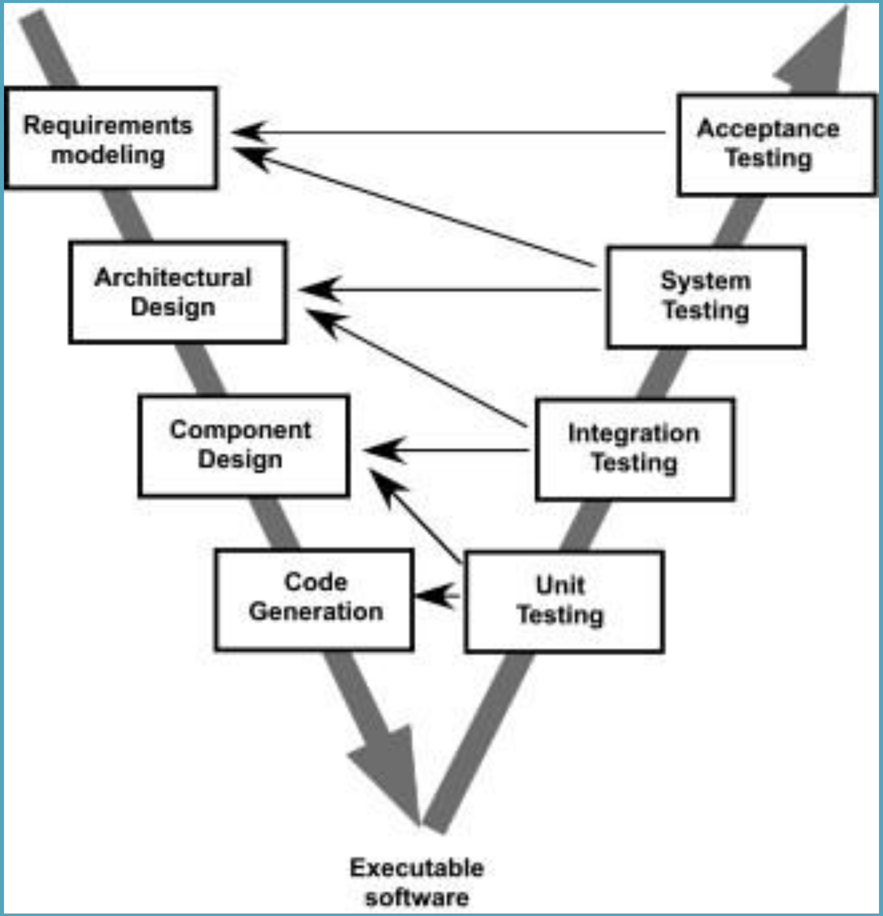
Waterfall Usage

- Well suited for projects where requirements can be understood easily *and unchanging requirements* and technology decisions are easy *(small company website development).*
- Has been used widely For standard/familiar type of projects
- Well suited to the out sourcing model
  - *Simple small or mid-sized projects with clearly defined and unchanging requirements Projects with the need for stricter control, predictable budget and timelines (e.g., governmental projects).*
  - *Projects where a well-known technology stack and tools are used.*

Waterfall disadvantages

- Assumes that requirements can be specified and frozen early
- May fix hardware and other technologies too early
- Follows the “big bang” approach – all or nothing delivery; too risky
- Very document oriented, requiring docs at the end of each phase

The V-Model



A variation of waterfall model depicts the relationship of quality assurance actions to the actions associated with communication, modeling and early code construction activates.

Team first moves down the left side of the V to refine the problem requirements. Once code is generated, the team moves up the right side of the V, performing a series of tests that validate each of the models created as the team moved down the left side.

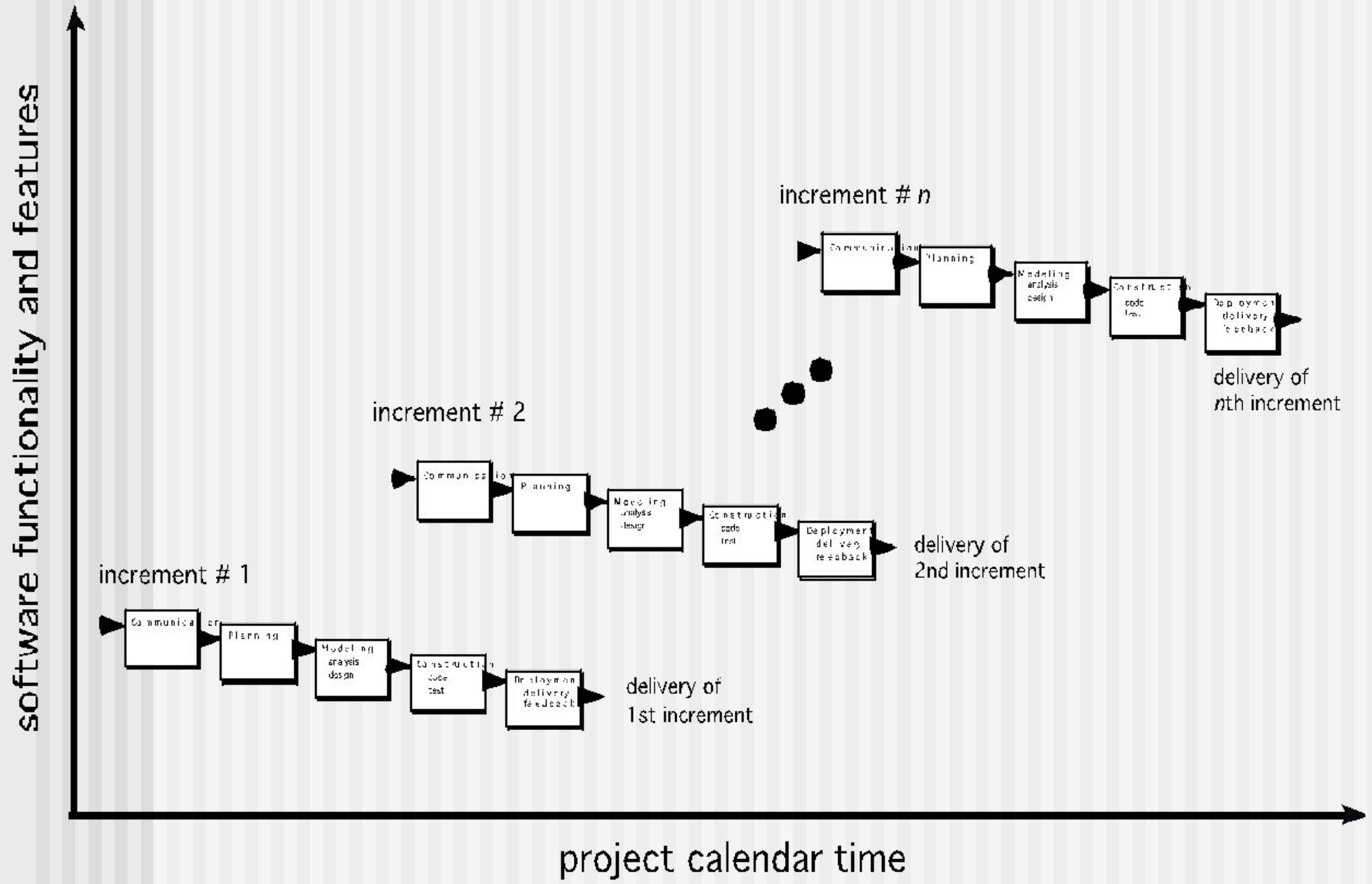
V-model….

- V-model is another linear model with each stage having a corresponding testing activity.
- implies quality control, it makes the V-model one of the most expensive and time-consuming models.
- As in the Waterfall case, all requirements are gathered at the start and cannot be changed.

Use:

- *Projects where failures and downtimes are unacceptable (e.g., medical software, aviation fleet management software).*

The Incremental Model


software functionality and features
increment # n
Communication
Planning
Modeling analysis design
Construction code test
Deployment delivery feedback
delivery of nth increment
increment # 2
delivery of 2nd increment
increment # 1
delivery of 1st increment
project calendar time

The Incremental Model

- When initial requirements are reasonably well defined, and the overall scope of the development effort prevents a purely linear process. A need to expand a limited set of new functions to a later system release.
- It combines elements of linear and parallel process flows. Each linear sequence produces deliverable increments of the software.
- The first increment is often a core product with many supplementary features. Users use it and evaluate it with more modifications to better meet the needs.

Incremental….

- suited for large projects, less expensive to the change of requirements as they support customer interactions with each increment.
- Initial versions of the software are produced early, which facilitates customer evaluation and feedback.
- It doesn’t fit into small projects, or projects that waterfall are best suited for; A structured process with a detailed, and accurate description of the system.

Iterative Development

- Counters the “all or nothing” drawback of the waterfall model
- Combines benefit of prototyping and waterfall
- Develop and deliver software in increments
- Each increment is complete in itself
- Can be viewed as a sequence of waterfalls
- Feedback from one iteration is used in the future iterations

Iterative Development

- Most Software Products follow it
- Used commonly in customized development also
  - Businesses want quick response for sw
  - Cannot afford the risk of all-or-nothing
- Newer approaches like XP, Agile,… all rely on iterative development

Iterative Development

- Benefits
  - Get-as-you-pay
  - feedback for improvement

- Drawbacks
  - Architecture/design may not be optimal
  - Amount of refactoring may increase
  - Total cost may increase

Iterative Development

- Applicability
  - where response time is important,
  - risk of long projects cannot be taken,
  - all req not known
- Execution
  - Each iteration is a mini waterfall – decide the specs, then plan the iteration
  - Length of iteration driven by amount of new functionality to be added in an iteration

Evolutionary Models

- Software system evolves over time as requirements often change as development proceeds. However, a limited version must be delivered to meet competitive pressure.
- Usually a set of core product or system requirements is well understood, but the details and extension have yet to be defined.
- so need a process model that has been explicitly designed to accommodate a product that evolved over time.
- It is iterative that enables you to develop increasingly more complete version of the software.
- Two types : Prototyping and Spiral models.

Evolutionary Models: Prototyping

- When to use: Customer defines a set of general objectives but does not identify detailed requirements for functions and features.
- Or Developer may be unsure of the efficiency of an algorithm, the form that human computer interaction should take.

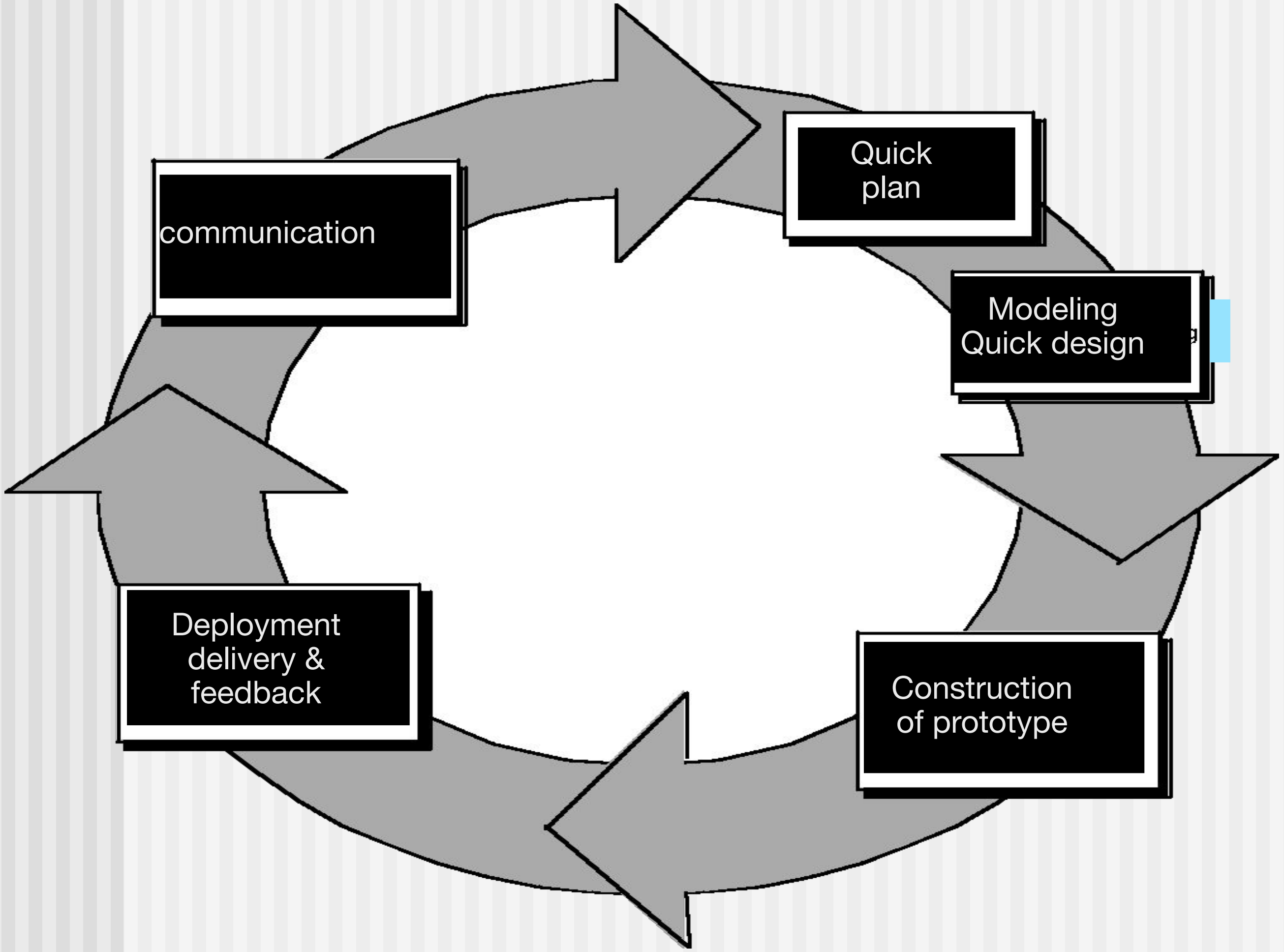
Evolutionary Models: Prototyping : What Step

- Begins with communication by meeting with stakeholders to define the objective, identify whatever requirements are known, outline areas where further definition is mandatory.
- A quick plan for prototyping and modeling (quick design) occur. Quick design focuses on a representation of those aspects the software that will be visible to end users. (interface and output).
- Design leads to the construction of a prototype which will be deployed and evaluated. Stakeholder's comments will be used to refine requirements.

Evolutionary Models: Prototyping-use

- Both stakeholders and software engineers like the prototyping paradigm.
- Users get a feel for the actual system, and developers get to build something immediately.
- However, engineers may make compromises in order to get a prototype working quickly.
- The less-than-ideal choice may be adopted forever after you get used to it.

Evolutionary Models: Prototyping



Prototyping…

- Advantages
  - Requirement will be more stable and more likely to satisfy user needs
  - Early opportunity to explore scale/performance issues
  - Ability to modify or cancel the project early
  - Enhanced user engagement
- Disadvantages:
  - Potential hit on cost and schedule and Risk

Prototyping…

- Applicability:
  - When req are hard to elicit
  - When confidence in reqs is low
  - Where reqs are not well understood
  - When design is driven by user needs

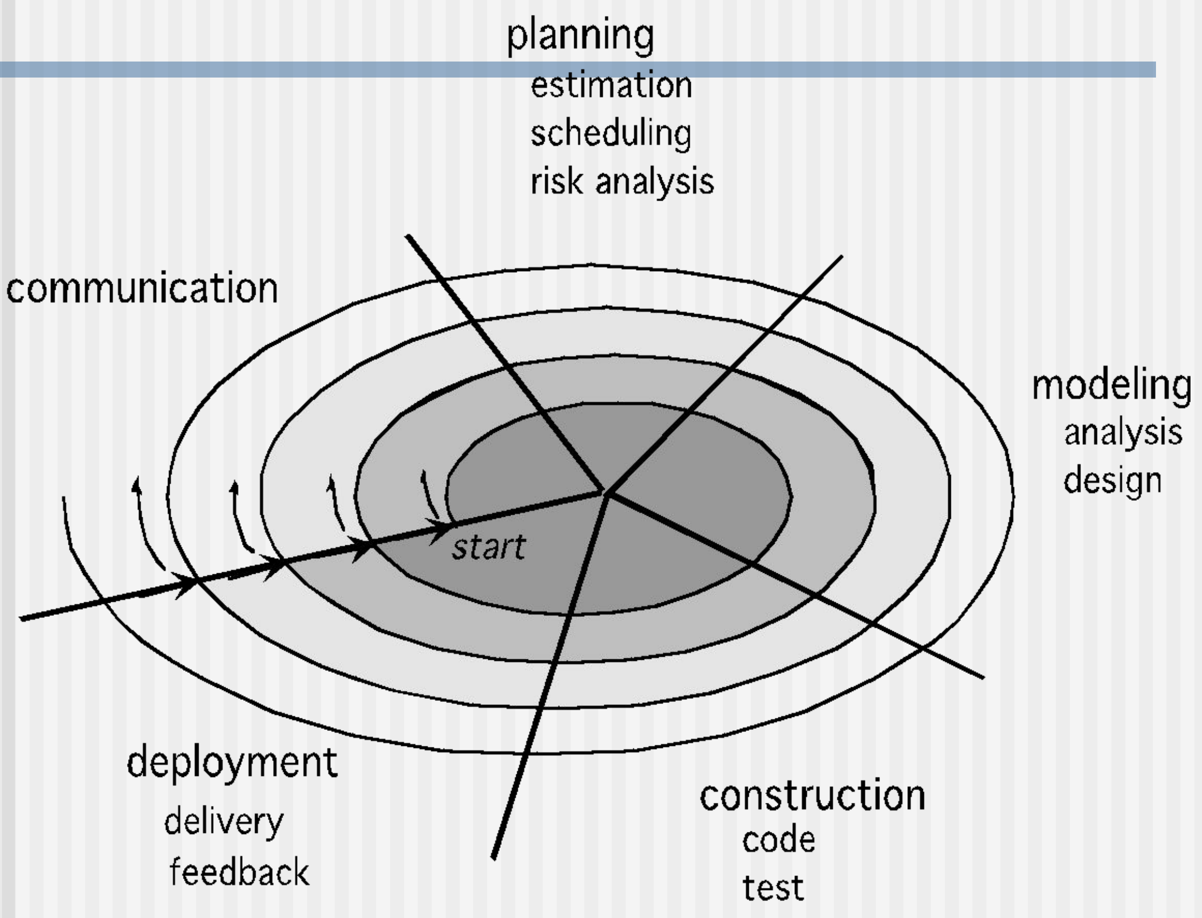
Evolutionary Models: The Spiral

- It couples the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model and is a risk-driven process model generator that is used to guide multi-stakeholder concurrent engineering of software intensive systems.
- Two main distinguishing features: one is cyclic approach for incrementally growing a system's degree of definition and implementation while decreasing its degree of risk. The other is a set of anchor point milestones for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions.
- A series of evolutionary releases are delivered. During the early iterations, the release might be a model or prototype. During later iterations, increasingly more complete version of the engineered system are produced.

Evolutionary Models: The Spiral

- The first circuit in the clockwise direction might result in the product specification; subsequent passes around the spiral might be used to develop a prototype and then progressively more sophisticated versions of the software. Each pass results in adjustments to the project plan. Cost and schedule are adjusted based on feedback. Also, the number of iterations will be adjusted by project manager.
- Good to develop large-scale system as software evolves as the process progresses and risk should be understood and properly reacted to. Prototyping is used to reduce risk.
- However, it may be difficult to convince customers that it is controllable as it demands considerable risk assessment expertise.

Evolutionary Models: The Spiral



Spiral….

- spiral model puts focus on risk assessment.
- need to engage people with a strong background in risk evaluation.
- A typical Spiral iteration , important activities - planning, risk analysis, prototypes creation, and evaluation of the previously delivered part.
- Repeated spiral cycles seriously extend project timeframes.
- intensive customer involvement appears

Spiral…

Use :

- *Projects with unclear business needs or too ambitious/innovative requirements.*
- *Projects that are large and complicated.*
- *Research and development (R&D) activity or the introduction of a new service or a product*

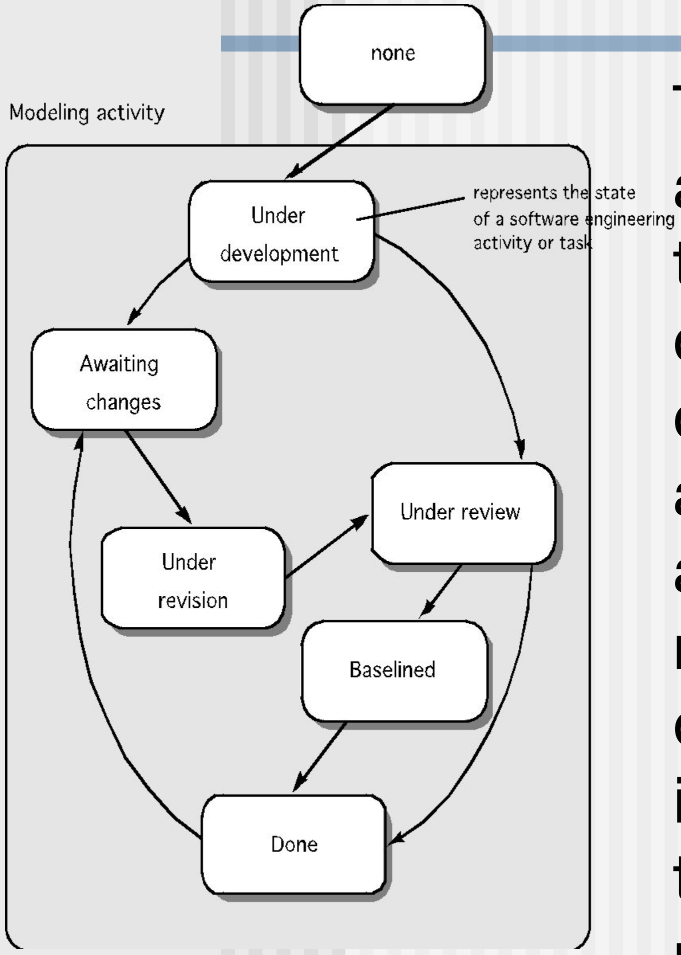
Three Concerns on Evolutionary Processes

- First concern is that prototyping poses a problem to project planning because of the uncertain number of cycles required to construct the product.
- Second, it does not establish the maximum speed of the evolution. If the evolution occur too fast, without a period of relaxation, it is certain that the process will fall into chaos. On the other hand if the speed is too slow then productivity could be affected.
- Third, software processes should be focused on flexibility and extensibility rather than on high quality. We should prioritize the speed of the development over zero defects. Extending the development in order to reach high quality could result in a late delivery of the product when the opportunity role has disappeared.

Concurrent Model

- Allow a software team to represent iterative and concurrent elements of any of the process models.
- For example, the modeling activity defined for the spiral model is accomplished by invoking one or more of the following actions: prototyping, analysis and design.

Concurrent Model



The Figure shows modeling may be in any one of the states at any given time. For example, after communication activity has completed its first iteration and in the awaiting changes state. The modeling activity was in inactive state, now makes a transition into the under development state. If customer indicates changes in requirements, the modeling activity moves from the under development state into the awaiting changes state.

Concurrent….

- Concurrent modeling is applicable to all types of software development and provides an accurate picture of the current state of a project, i.e. it defines a process network. Each activity, action or task on the network exists simultaneously with other activities, actions or tasks.
- Events generated at one point trigger transitions among the states.

Summary – waterfall

Strength	Weakness	Types of Projects
Simple Easy to execute Intuitive and logical Easy contractually	All or nothing – too risky Req frozen early May chose outdated hardware/tech Disallows changes No feedback from users Encourages req bloating	Well understood problems, short duration projects, automation of existing manual systems

Summary – Evolutionary

Strength	Weakness	Types of Projects
Helps req elicitation Reduces risk Better and more stable final system	heavy Possibly higher cost and schedule Encourages req bloating Disallows later change	Systems with novice users; or areas with req uncertainity. Heavy reporting based systems can benefit from UI proto

Summary – Incremental

Strength	**Weakness**	**Types of Projects**
Regular deliveries, leading to biz benefit Can accommodate changes naturally Allows user feedback Avoids req bloating Naturally prioritizes req Allows reasonable exit points Reduces risks	Overhead of planning each iteration Total cost may increase System arch and design may suffer Rework may increase	For businesses where time is imp; risk of long projects cannot be taken; req not known and evolve with time

Exercises

1. Explain waterfall model with figure, also write down its strength, weakness and types of projects in which it is useful
2. Explain V model with figure, also write down its strength, weakness and types of projects in which it is useful
3. Explain Spiral model with figure, also write down its strength, weakness and types of projects in which it is useful
4. Explain Incremental model with figure, also write down its strength, weakness and types of projects in which it is useful
5. Explain Evolutionary model with figure, also write down its strength, weakness and types of projects in which it is useful
6. Explain Concurrent model with figure, also write down its strength, weakness and types of projects in which it is useful
7. What is prescriptive model? Explain in brief different types of prescriptive model.