Lecture 1.3.1
Evolutionary Process Models

Introduction to Software Engineering

Different Process Models

- Waterfall Model (Linear Sequential Model)
- Incremental Process Model
- Prototyping Model
- The Spiral Model
- Rapid Application Development Model
- Agile Model

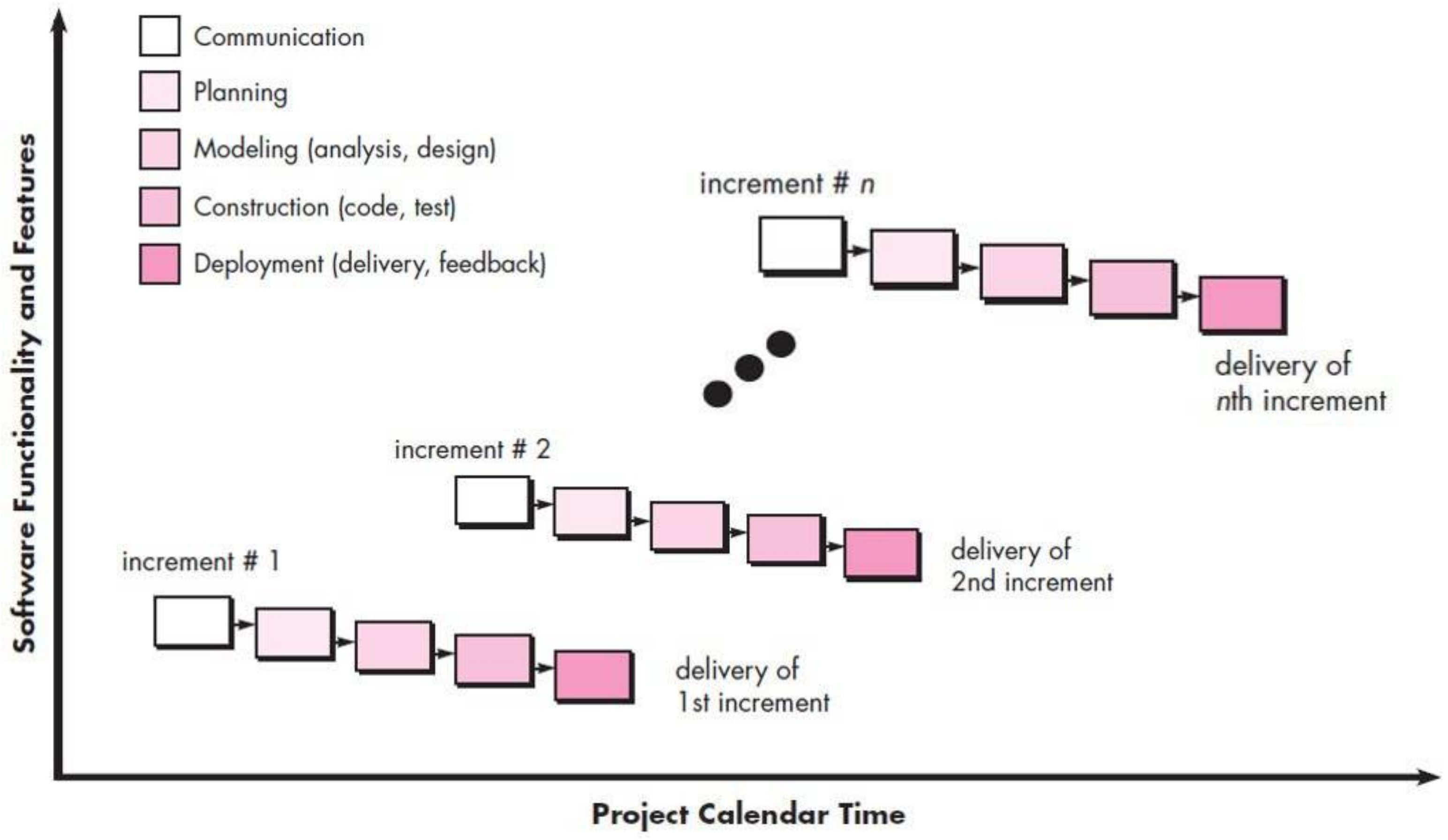
Evolutionary Process Models

- When a set of core product or system requirements is well understood but the details of product or system extensions have yet to be defined.
- In this situation there is a need of process model which specially designed to accommodate product that evolve with time.
- Evolutionary Process Models are specially meant for that which produce an increasingly more complete version of the software with each iteration.
- Evolutionary Models is a combination of Incremental & iterative.
- Evolutionary models are
  - Prototyping Model
  - Spiral Model
  - The Incremental Model.

Advantages & Disadvantages

- **Advantages:**
- In evolutionary model, a user gets a chance to experiment partially developed system.
- It reduces the error because the core modules get tested thoroughly.
- Ensure a greater level of customer satisfaction and comfort
- **Disadvantages:**
- Sometimes it is hard to divide the problem into several versions that would be acceptable to the customer which can be incrementally implemented and delivered.

Incremental Process Model


Communication
Planning
Modeling (analysis, design)
Construction (code, test)
Deployment (delivery, feedback)
increment # n
delivery of nth increment
increment # 2
delivery of 2nd increment
increment # 1
delivery of 1st increment
Software Functionality and Features
Project Calendar Time

Incremental Process Model cont.

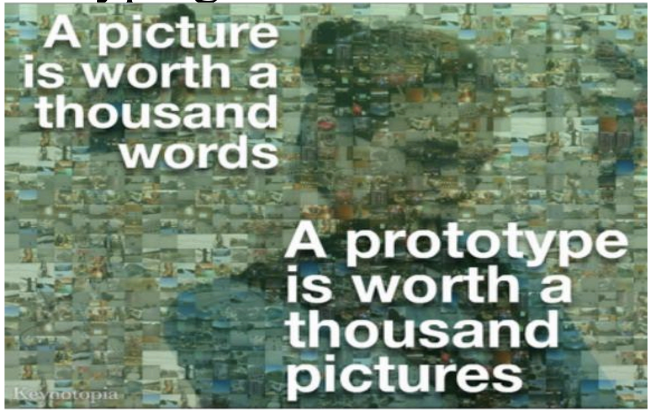
- The incremental model combines elements of linear and parallel process flows.
- This model applies linear sequence in a iterative manner.
- Initially core working product is delivered.
- Each linear sequence produces deliverable "increments" of the software.
- For example, word-processing software developed using the incremental model
  - It might deliver basic file management, editing and document production functions in the first increment
  - more sophisticated editing in the second increment; spelling and
  - grammar checking in the third increment; and advanced page layout
  - capability in the fourth increment.

Incremental Process Model cont.

- When to Use ?
  - When the requirements of the complete system are clearly defined and understood but staffing is unavailable for a complete implementation by the business deadline.
- Advantages
  - Generates working software quickly and early during the software life cycle.
  - It is easier to test and debug during a smaller iteration.
  - Customer can respond to each built.
  - Lowers initial delivery cost.
  - Easier to manage risk because risky pieces are identified and handled during iteration.

Prototyping model

- Prototyping model is appropriate when
  - Customers have general objectives of software but do not have detailed requirements for functions & features.
  - Developers are not sure about efficiency of an algorithm & technical feasibilities.
- It serves as a mechanism for identifying software requirements.
- Prototype can be serve as "the first system .
- Both stakeholders and software engineers like prototyping model
  - Users get feel for the actual system
  - Developers get to build something immediately

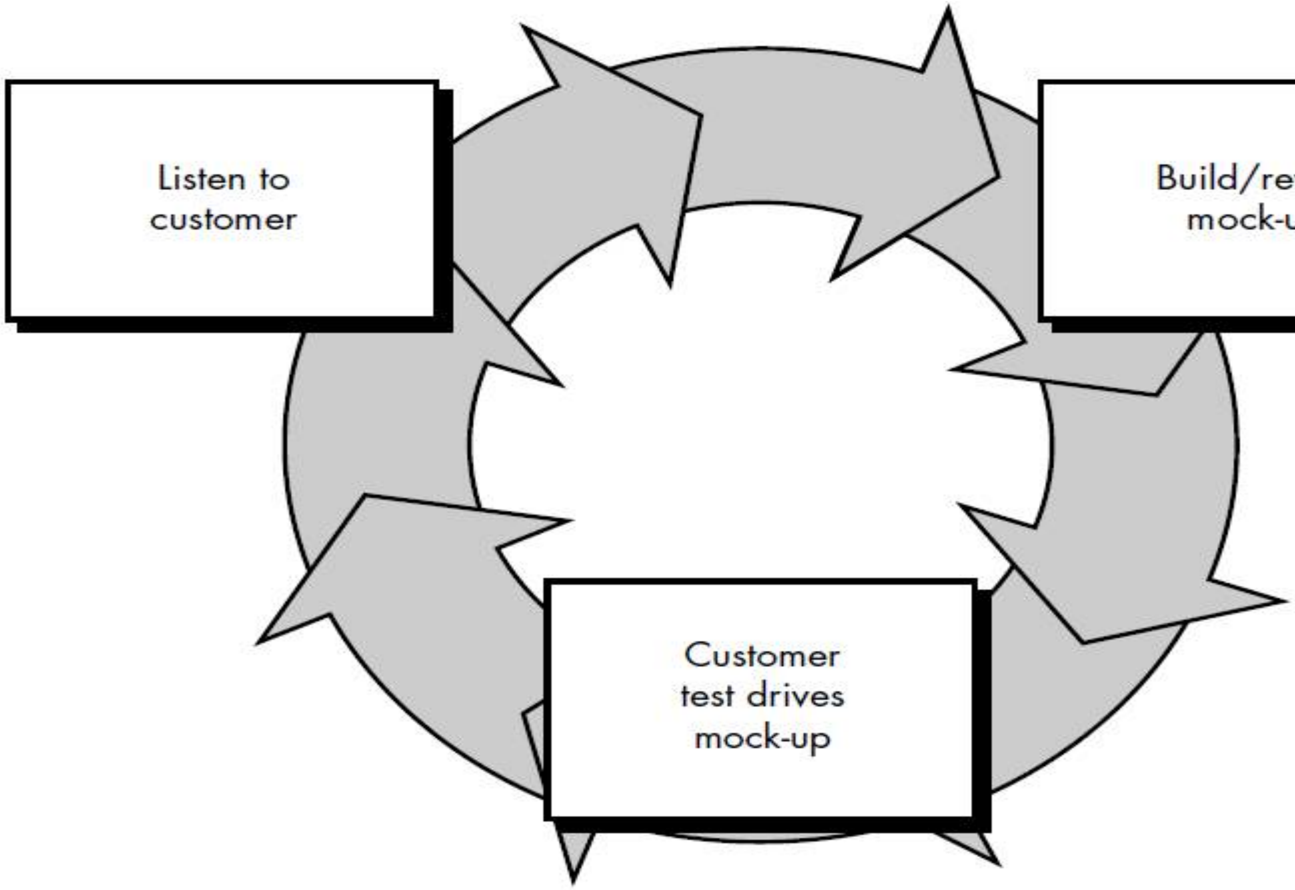

A picture is worth a thousand words
A prototype is worth a thousand pictures
Keynotopia

THE PROTOTYPING MODEL

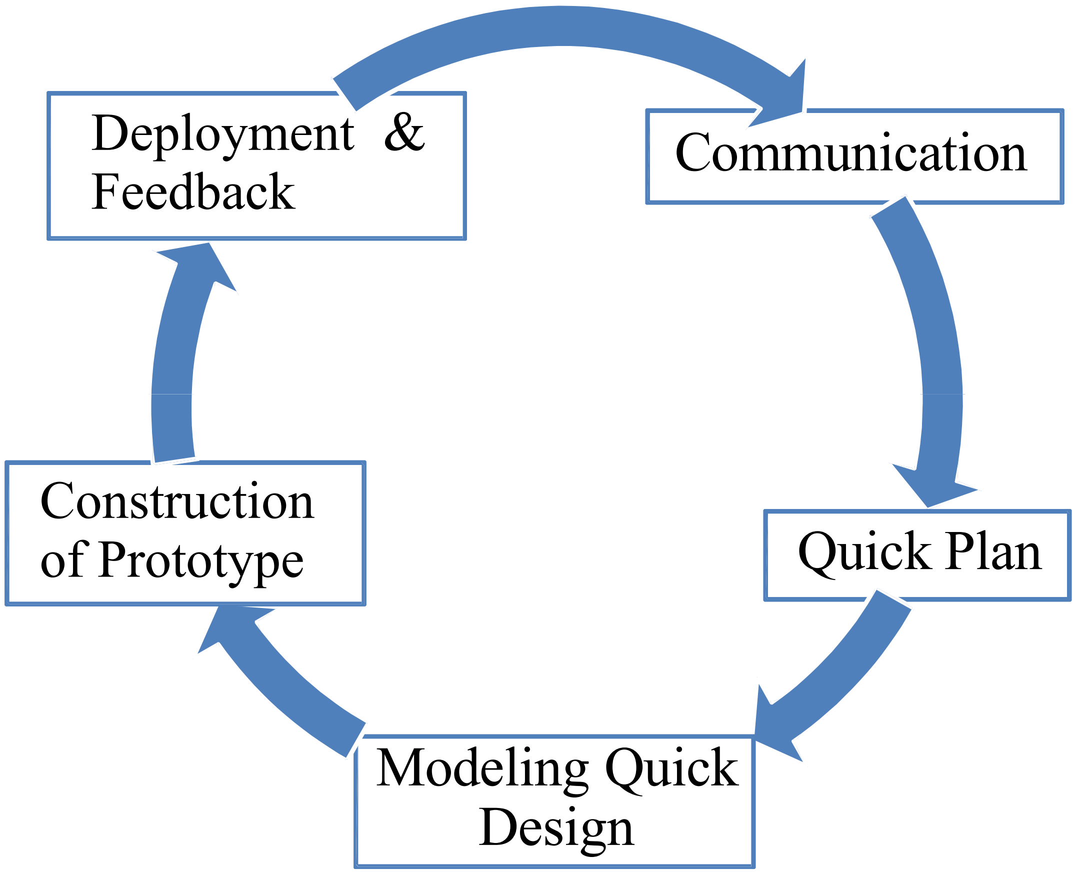
- **Need:**

1. Customer defines a set of general objectives for software but is not able to identify detailed input, processing, or output requirements
2. The developer may be unsure of
   a) Efficiency of an algorithm,
   b) Adaptability of an operating system
   c) Form that human/machine interaction should take.

Prototyping Model



Prototyping model cont.



Prototyping model cont.

- It works as follow
  - Communicate with stockholders & define objective of Software
  - Identify requirements & design quick plan
  - Model a quick design (focuses on visible part of software)
  - Construct Prototype & deploy
  - Stakeholders evaluate this prototype and provides feedback
  - Iteration occurs and prototype is tuned based on feedback
- Problem Areas
  - Customer demand that "a few fixes" be applied to make the prototype a working product, due to that software quality suffers as a result
  - Developer often makes implementation in order to get a prototype working quickly; without considering other factors in mind like OS, Programming language, etc.

Prototyping model cont.

- Advantages
  - Users are actively involved in the development
  - Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed
  - Errors can be detected much earlier

Often not used, as it is feared that **development costs may become large**.
However, in some situations, the cost of software development without prototyping may be more than with prototyping.

Prototype : When to use?

- Suitable for projects where requirements are hard to determine
- Confidence in the stated requirements is low.
  - Use of waterfall model in such projects leads to requirement changes and associated rework while the development is going on.
- Requirements frozen after experience with the prototype are likely to be more stable.
- Excellent technique for reducing some types of risks associated with a project.

Prototyping can be problematic

Following are reasons:

1. Software quality or long-term maintainability concerns
2. The developer often makes implementation compromises

- Customer and developer must both **agree** that the prototype is built to **serve as a mechanism for defining requirements.**