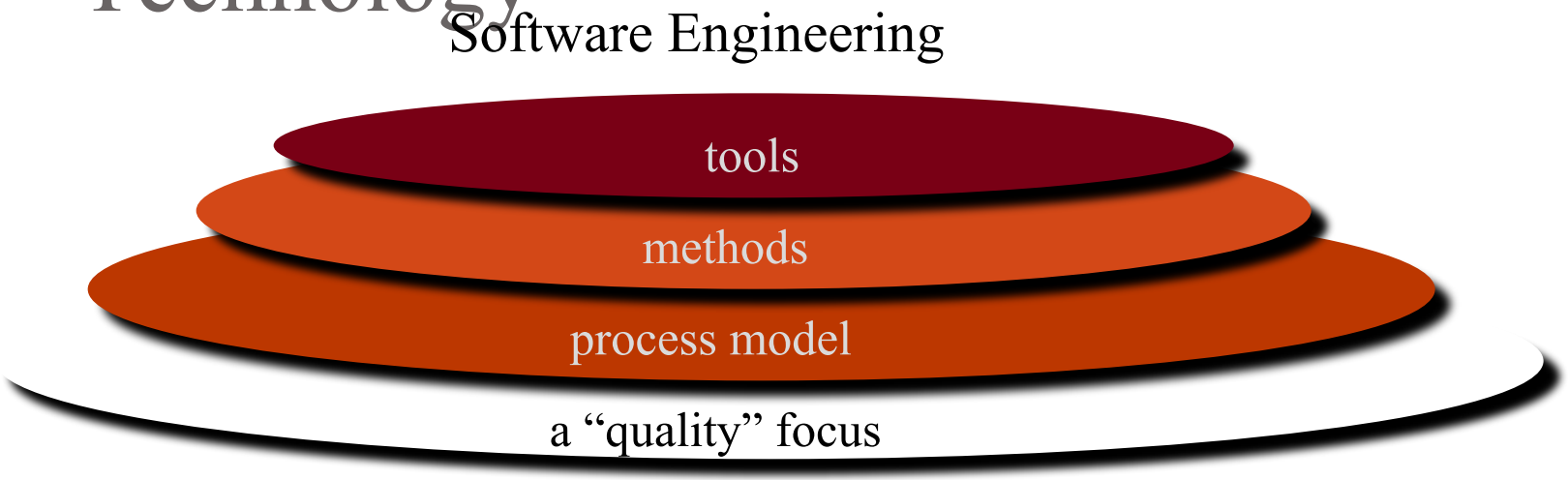
Contents

- Generic Process Model
- Prescriptive Process Model
- Waterfall Model
- Incremental Process (RAD) Model
- Evolutionary Process Model
- Agile Process Model

A Layered Technology



Process

- Glue that holds the technology layers together.
- Enables rational & timely development of computer software.
- Defines framework that must be established for effective delivery of software engineering technology.
- Forms the basis for management control of software projects.
- Establishes the context in which technical methods are applied, work products are produced, milestones are established, quality is ensured, change is properly managed.

Methods

- Provide the technical “how to’s “ for building software.
- Encompass a broad array of tasks that include communication, requirement analysis, design modeling, construction, testing, support.

Tools

- Provide automated or semi-automated support for the process and the methods.
- Integration of tools
- Information is shared

A Process Framework

- Process framework

Framework activities
- work tasks
- work products
- milestones & deliverables
- QA checkpoints

Umbrella Activities

Framework Activities

- Communication
- Planning
- Modeling
  - Analysis of requirements
  - Design
- Construction
  - Code generation
  - Testing
- Deployment

Umbrella Activities

- Software project management
- Formal technical reviews
- Software quality assurance
- Software configuration management
- Work product preparation and production
- Reusability management
- Measurement
- Risk management

The Process Model: Adaptability

- The framework activities will always be applied on every project but the tasks for each activity will vary based on:
  - The type of project
  - Characteristics of the project
  - Common sense judgment; concurrence of the project team

Process Patterns

- A *process pattern*
  - describes a process-related problem that is encountered during software engineering work,
  - identifies the environment in which the problem has been encountered, and
  - suggests one or more proven solutions to the problem.
- Stated in more general terms, a process pattern provides you with a *template* [Amb98]—a consistent method for describing problem solutions within the context of the software process.

Process Pattern Template

- ✔ **Pattern Name**
- ✔ **Forces :** The environment in which the pattern is encountered and the issues that make problem visible and may affect its solution.
- ✔ **Type** : Stage Pattern , Task Pattern , Phase Pattern
- ✔ **Initial Context :** Describes the condition under which the pattern applies
- ✔ Problem: The specific problem to be solved by the pattern
- ✔ **Solution:** Describes how to implement the patter successfully
- ✔ **Resulting Context**: Describes the conditions that will result once the pattern has been successfully implemented
- ✔ **Related Pattern:** List of all process pattern that are directly related.
- ✔ **Known uses and Examples:** Specific instances in which the pattern is applicable.

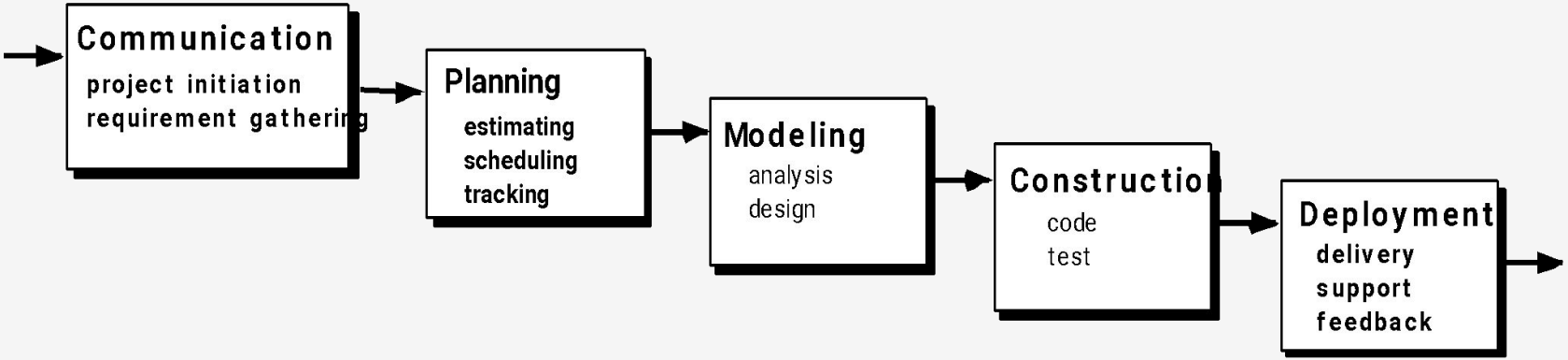
Process Pattern Example

Use of Process Pattern applicable when stakeholders have a general idea of what must be done but are unsure of specific software requirements.

- ✔Pattern Name : RequirementsUnclear
- ✔Intent :
- ✔Type : Phase
- ✔Initial Context :
- ✔Solution:
- ✔Resulting Context
- ✔Related Pattern
- ✔Known uses and Examples

Process Model

The Waterfall Model



- Systematic and sequential approach to software development.
- Classic life cycle model
- Model mandates that each phase will be executed after completion of the previous phase

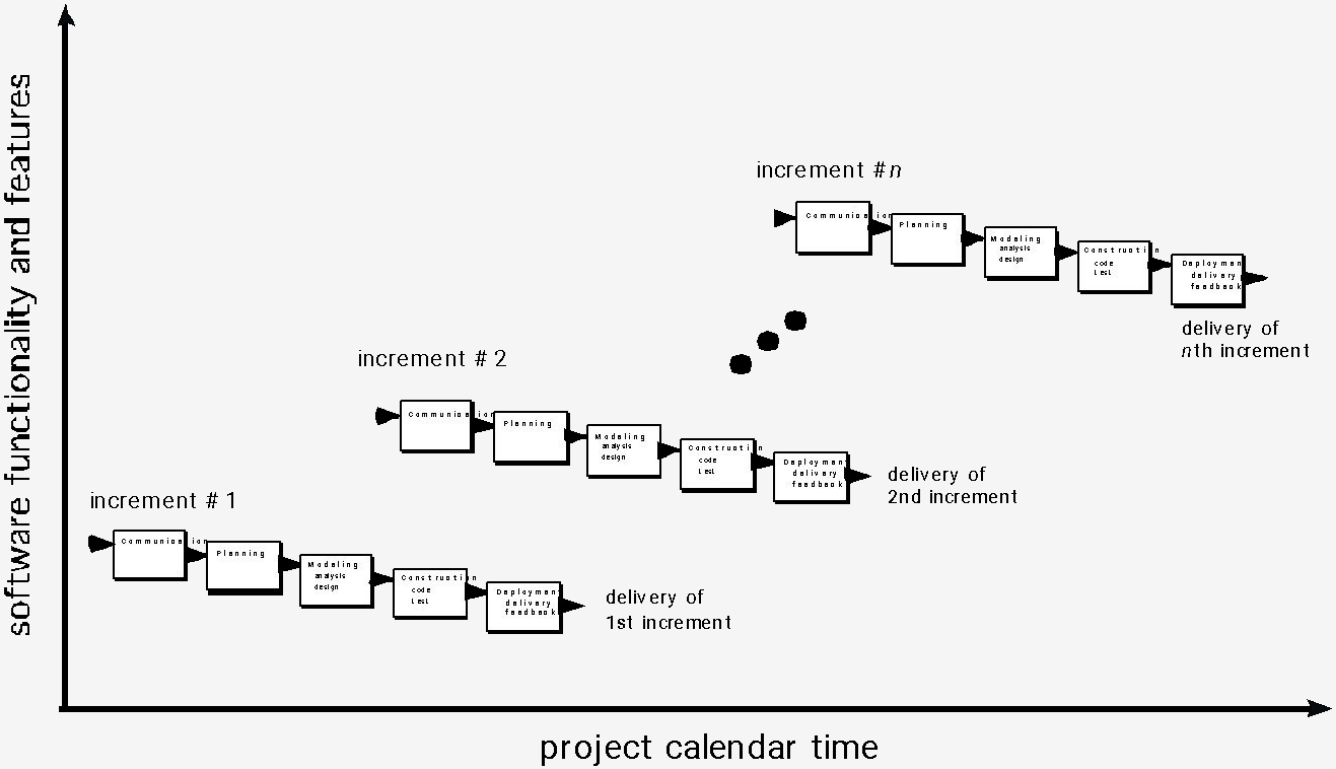
Advantages

- Simplicity
- Logical structuring of the different activities in a software project
- Model is perfect for projects where requirements are very well defined.

Disadvantages

- It is strict about moving only one step at a time. This is to ensure that the complete project is moving together.
- Customer has difficulty expressing requirements in their entirely.
- Has difficulty accommodating natural uncertainty that exists at the beginning of the cycle.
- Model does not allow capturing potential risk in the project.
- A working version of the software is not available until late in the process.

The Incremental Model



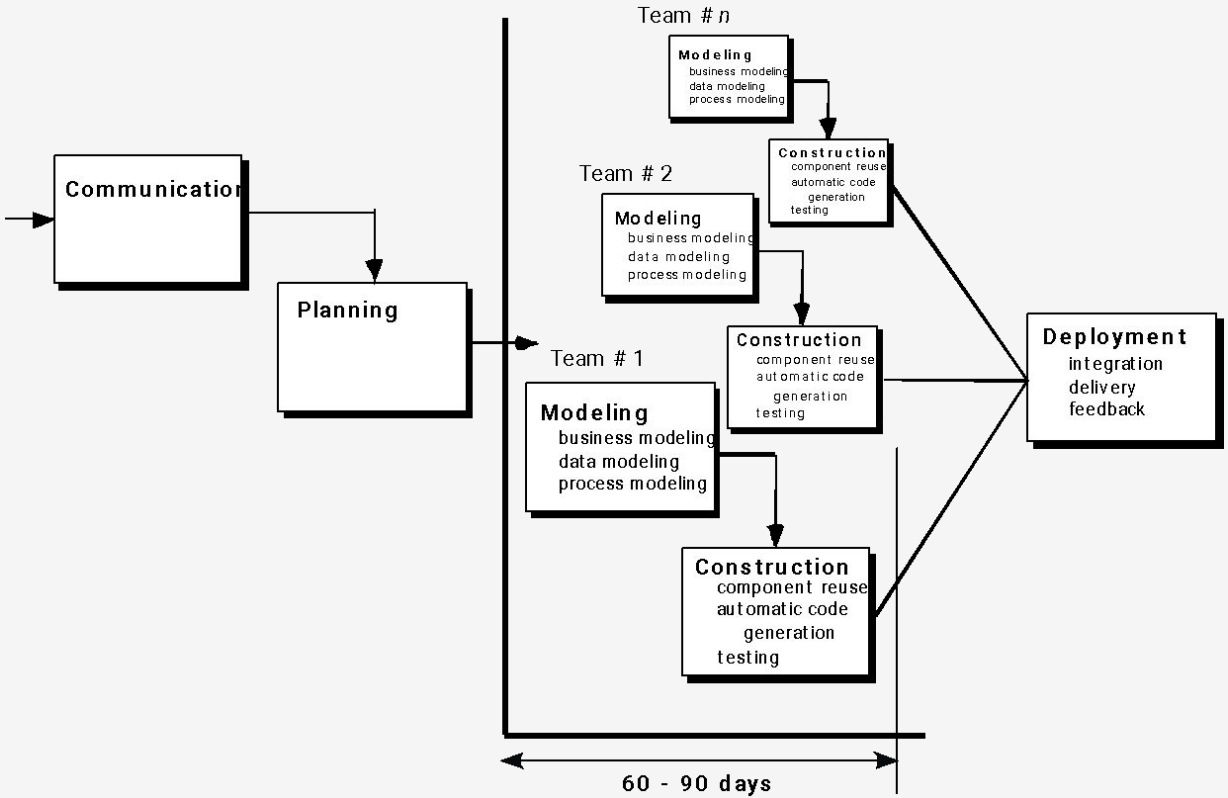
Theory

- Combines elements of the waterfall model applied in iterative manner.
- Applies linear sequences in a staggered fashion as calendar time progresses.
- Each linear sequence produces deliverable increments of the software. E.g. word processing software.
- Focuses on the delivery of an operational product with each increment.

Advantages

- Useful when staffing is unavailable.

The RAD Model



Theory

- Rapid Application Development
- It is recommended where there are tight deadlines and high pressure from customer
- Emphasizes on short development cycle
- Each major function can be addressed by a separate RAD team followed by the integration of the separately developed functionalities
- Necessitates the involvement of users throughout the development life cycle

Advantages & Disadvantages

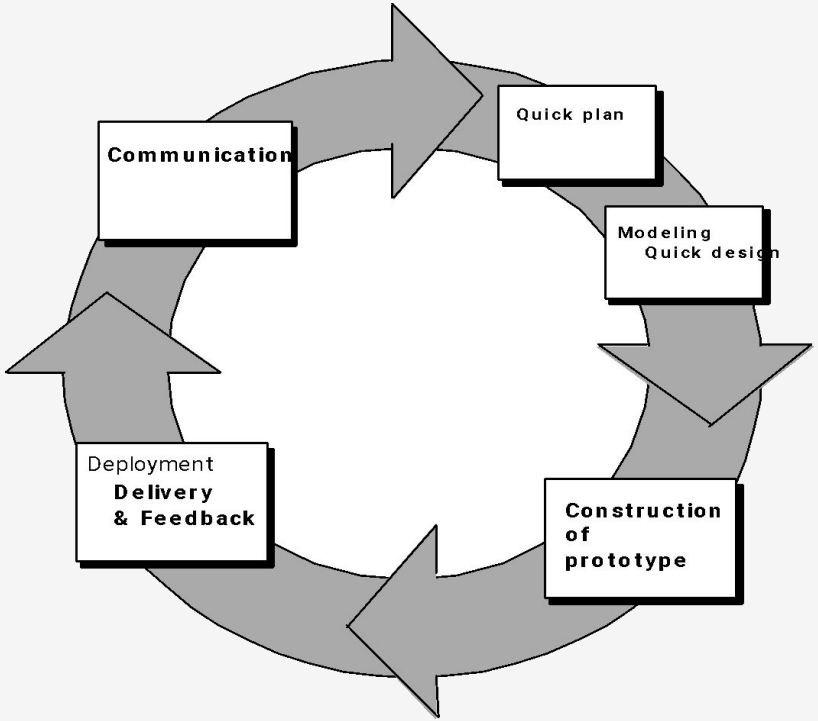
Advantages:

- Provides quick time to market.
- Fully functional system is expected within a short time of say 60 to 90 days.

Disadvantage :

- It requires sufficient human resources to create the right number of RAD teams.

Evolutionary Models: Prototyping



Theory

- Iterative approach to software development
- Useful when either the customer or the developer is unsure of the exact requirements of the software.
- Throw-way Model: Discard the model once all requirements are understood.
- Evolving Model: Refine the model every time when the requirements are clearer.

Advantages & Disadvantages

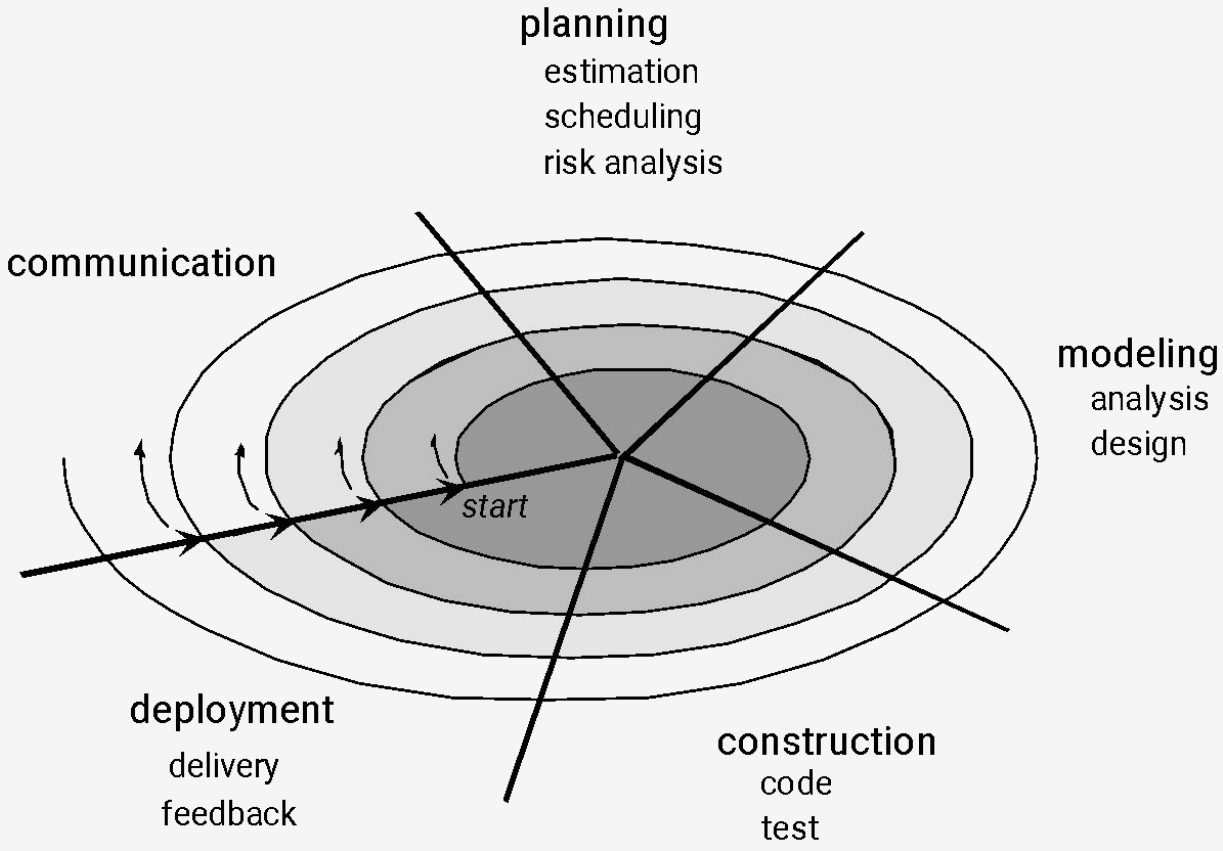
Advantage:

- Minimizing technical risks.

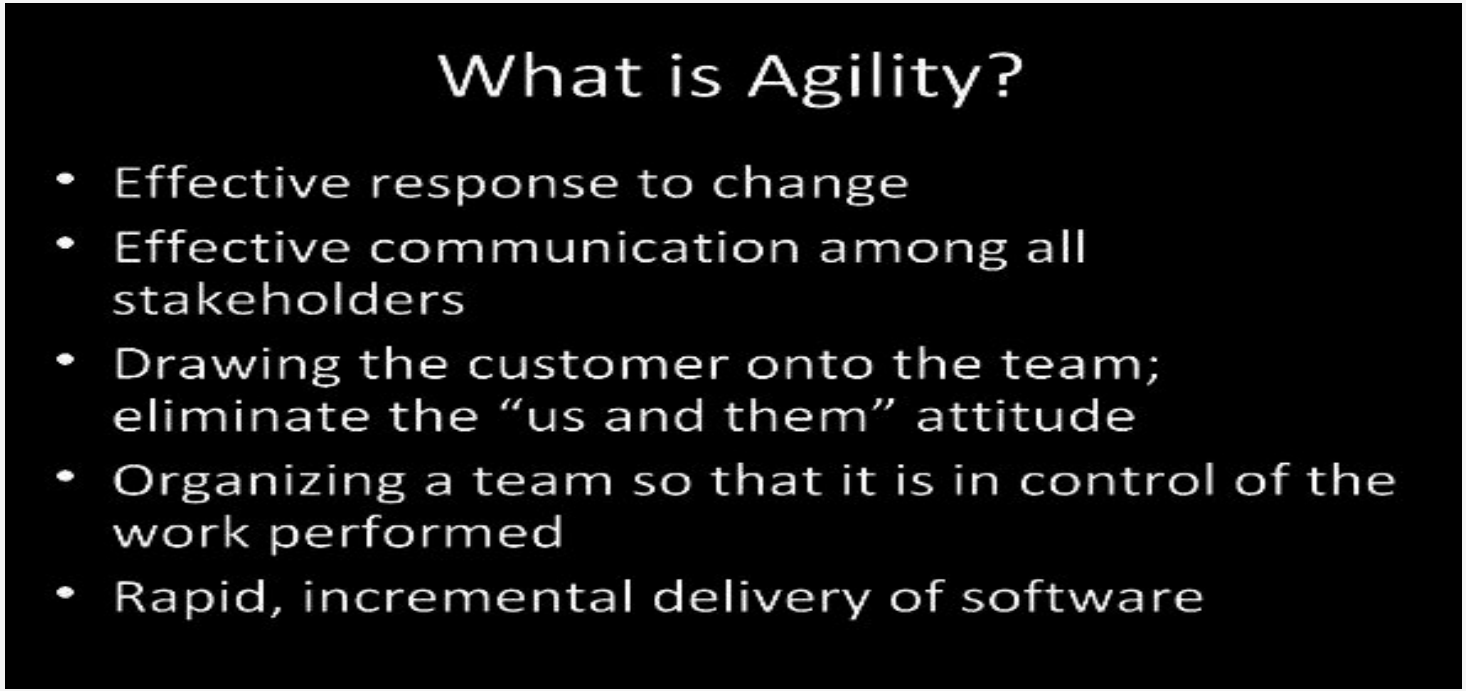
Disadvantage :

- It may lead to indiscipline of development

Evolutionary Models: The Spiral



Agile Process Model



Agile Process Model

- **Agile process model** refers to a software development approach based on iterative development.
- Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning.
- Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks.
- The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements.
- Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

References

- Roger Pressman, “Software Engineering: A Practitioner’s Approach”, Mcgraw Hill

Thank You