

Process Models

Process Model

Definition:

- A framework for the activities, actions, and tasks that are required to build high-quality software
- Represented as a set of work phases that is applied to design and build a software product
- Also called **Software Development Life Cycle** (SDLC)

Importance:

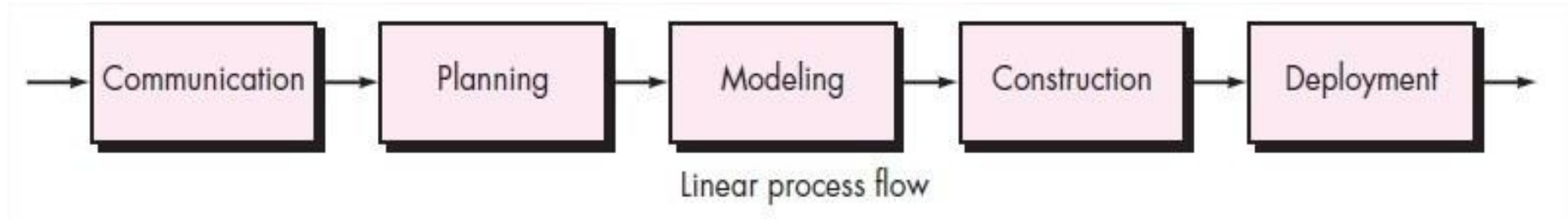
- To make the best use of human resources.
- Deliver the updates in regular manner.
- Make sure it's budget oriented

Framework Activities

- Communication
- Planning
- Modeling
 - Analysis of Requirements
 - Design
- Construction
 - Code Generation
 - Testing
- Deployment

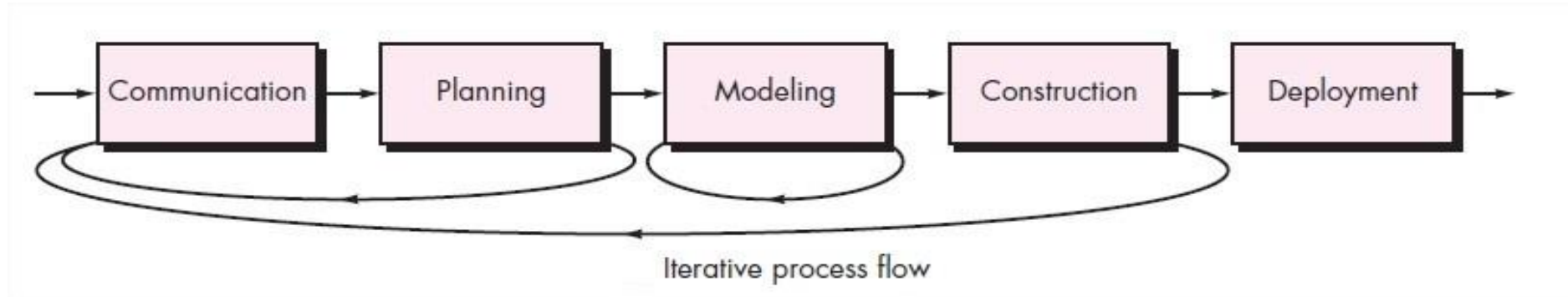
Process Flow

Linear Process Flow:



- executes each of the five activities in sequence

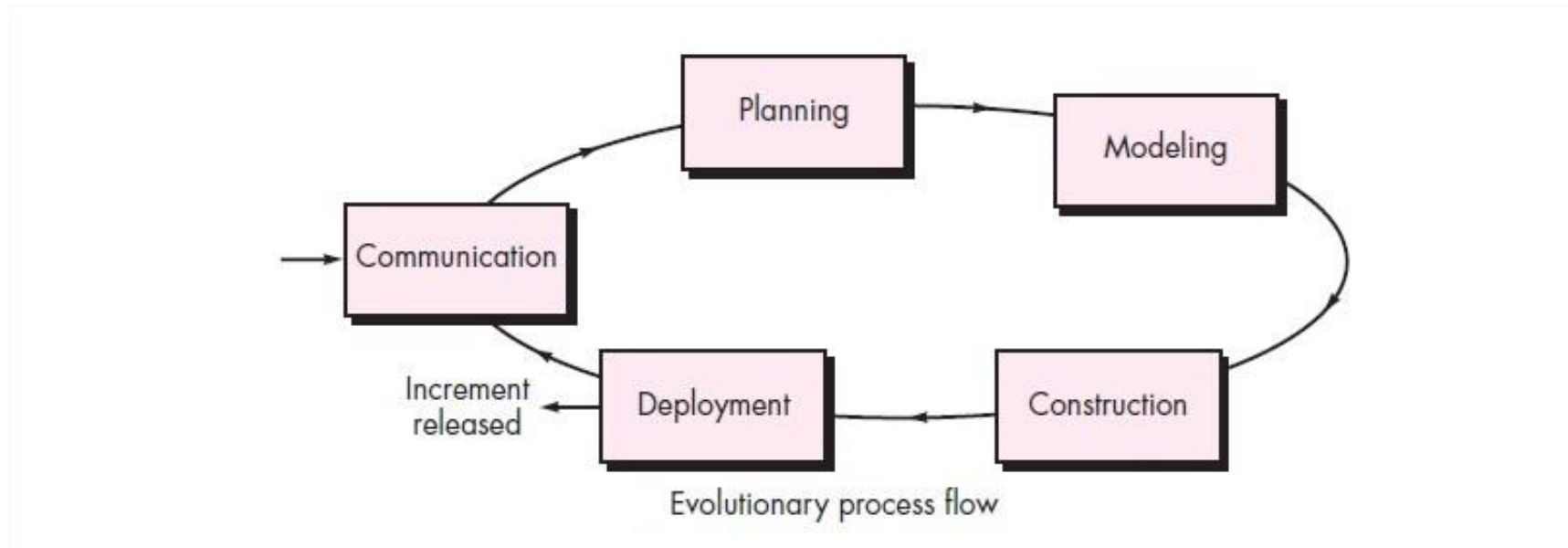
Iterative Process Flow:



- repeats one or more of the activities before proceeding to the next

Process Flow (Contd.)

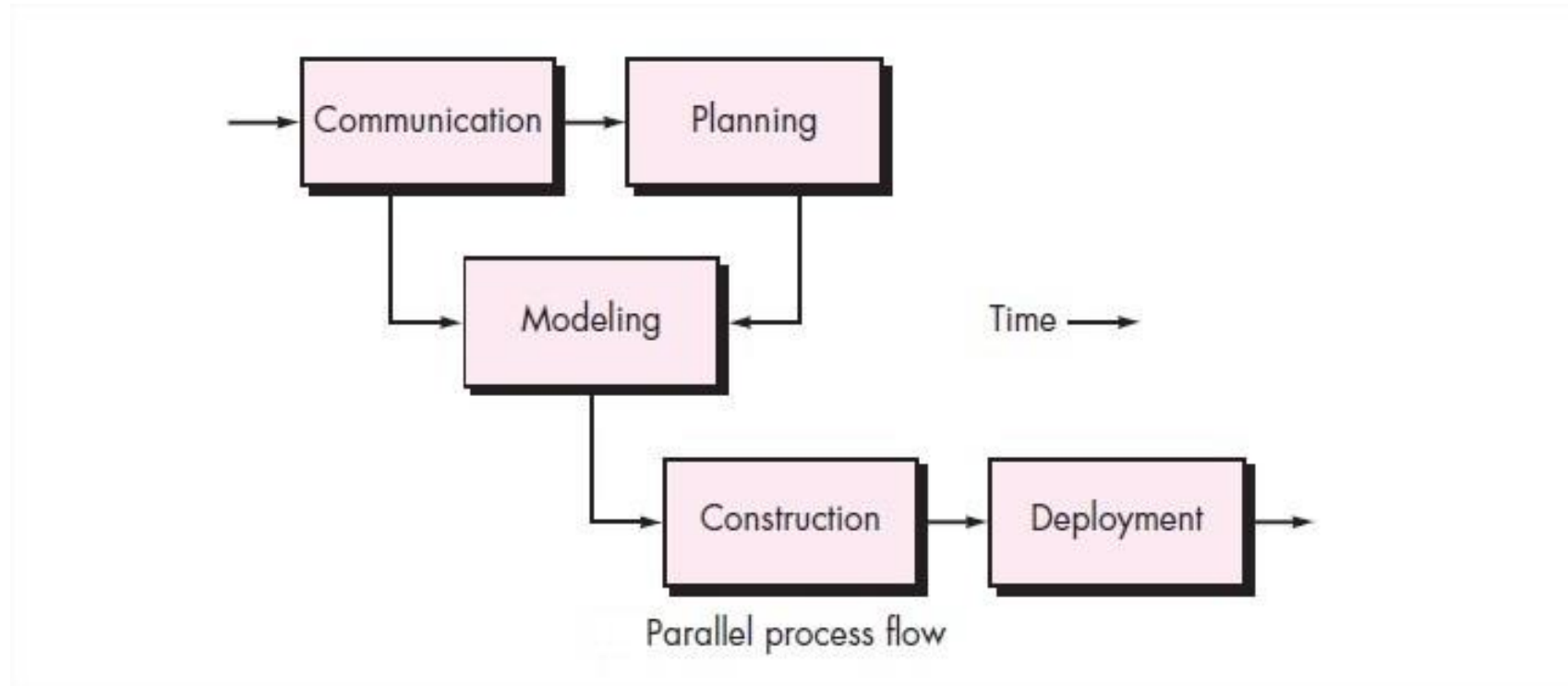
Evolutionary Process Flow:



- Executes the activities in a circular manner
- Each circuit through the five activities leads to a more complete version of the software.

Process Flow (Contd.)

Parallel Process Flow:



- Executes one or more activities in parallel with other activities
- For example, modeling for one aspect of the software might be executed in parallel with construction of another aspect of the software

Task Set

Defines the actual work to be done to accomplish the objectives of a software engineering action.

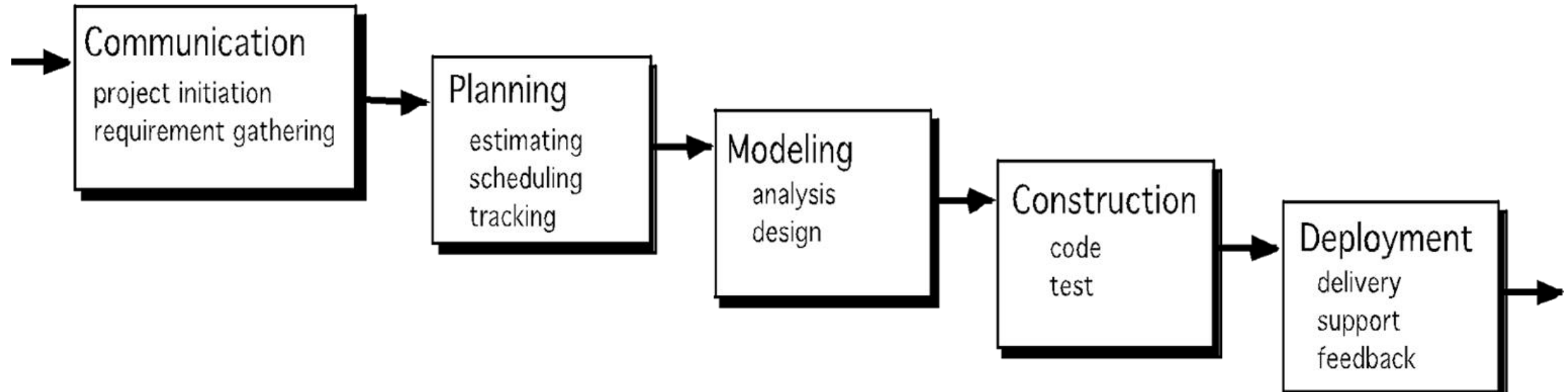
Identifying a Task Set:

For a small, relatively simple project, the task set for requirements gathering might look like this:

1. Make a list of stakeholders for the project
2. Invite all stakeholders to an informal meeting
3. Ask each stakeholders to make a list of features and functions required
4. Discuss requirements and build a final list
5. Prioritize requirements
6. Note areas of uncertainty

Waterfall Model

- ❑ The waterfall model is the classic process model – it is widely known, understood and used
- ❑ In some respect, waterfall is the "common sense" approach
- ❑ **When requirements are well defined and reasonably stable, it leads to a linear fashion**
- ❑ Sequential approach to software development



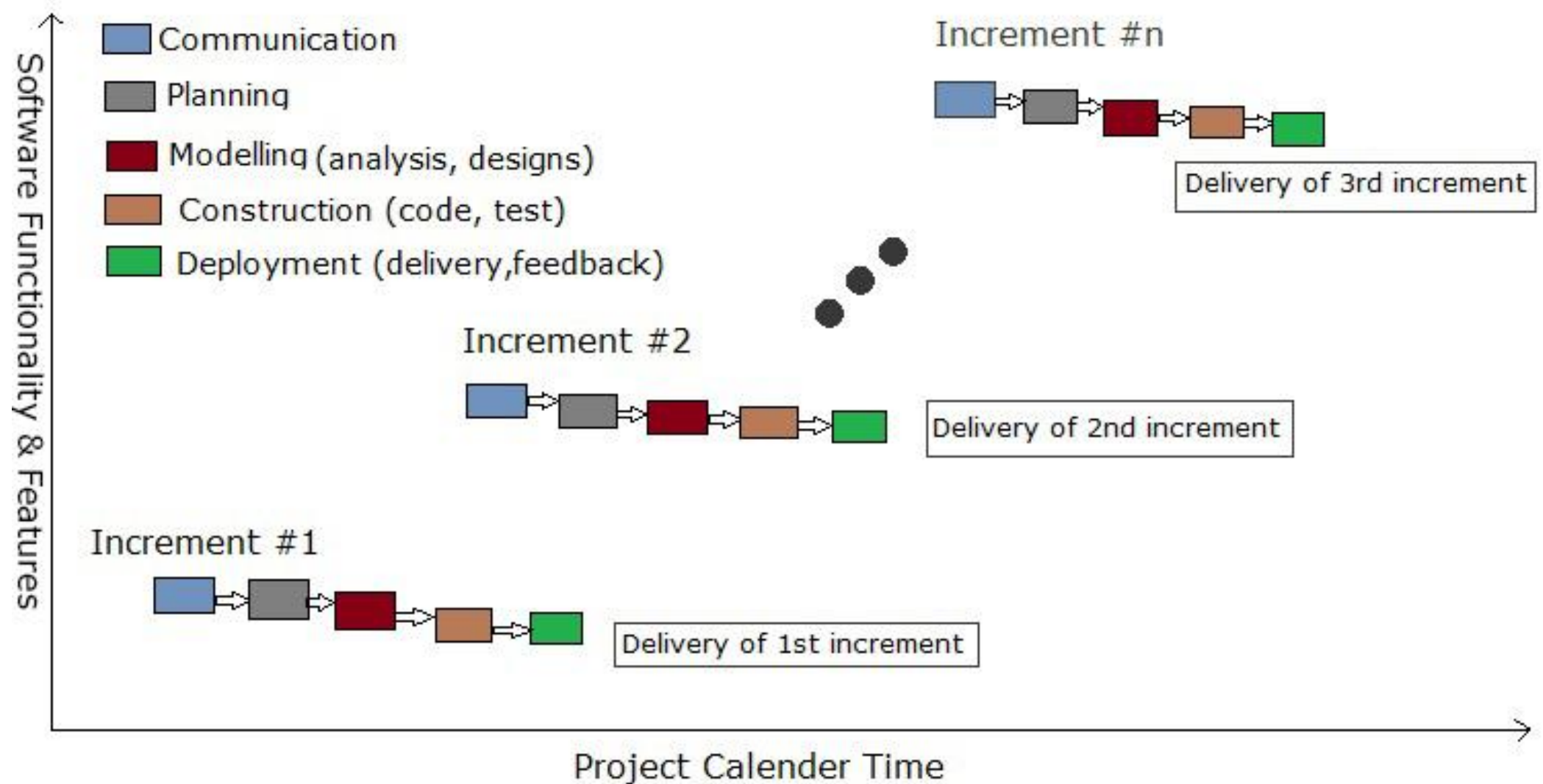
Waterfall Model: Advantages

- ❑ Easy to understand and implement
- ❑ Widely used and known (in theory!)
- ❑ **Fits other engineering process models: civil, mechanical etc.**
- ❑ **Reinforces good habits: define-before-design, design-before-code**
- ❑ Identifies deliverables and milestones
- ❑ **Document driven: People leave, documents don't**
- ❑ **Works well on short products and weak teams**

Waterfall Model: Disadvantages

- ☐ Doesn't reflect iterative nature of exploratory development
- ☐ Sometimes unrealistic to expect accurate requirements early in a project
- ☐ Software is delivered late, delays discovery of serious errors
- ☐ Difficult and expensive to change decisions

Incremental Model



Incremental Model

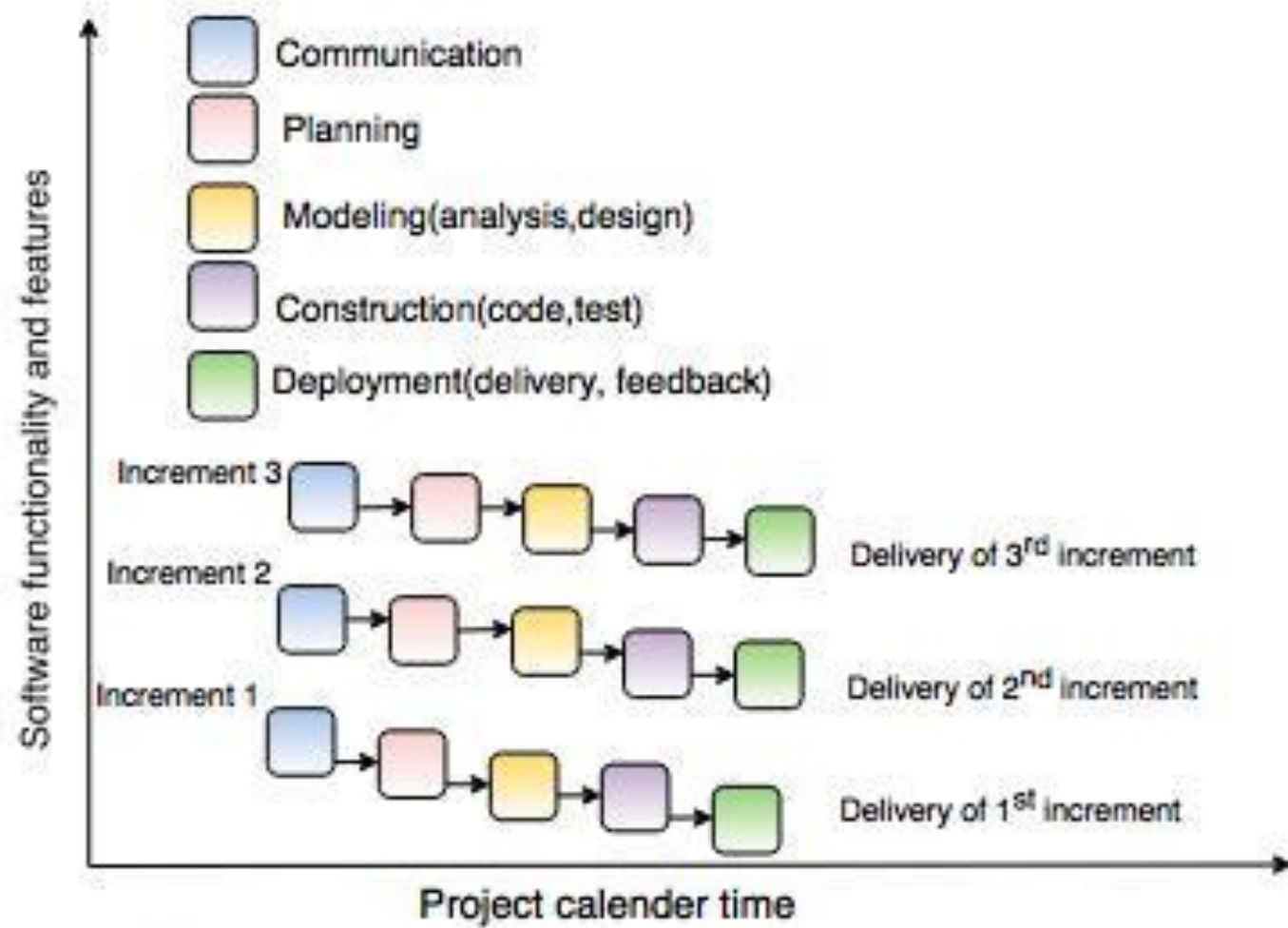


Fig. - Incremental Process Model

Incremental Model

- ☐ System development is broken down into many mini development projects
- ☐ Partial systems are successively built to produce a final total system
- ☐ Highest priority requirement is tackled first

When to use:

- ☐ Major requirements of the system are clearly understood
- ☐ When demand for an early release of a product arises
- ☐ When software engineering team are not very well skilled or trained

Incremental Model: Advantages

- ☐ Customer feedback after delivery of each component
- ☐ Initial version is released very quickly
- ☐ Risk of requirements changes is reduced
- ☐ Errors are easy to be identified
- ☐ Give quick results

Incremental Model: Disadvantages

- ☐ It requires a good planning and designing
- ☐ Actual cost may exceed the estimated cost

A Case Study of Incremental Model

Suppose we want to develop a web-based social network with the following functionalities:

- The user should signup the system.
- The user should log into the system and can send or accept the friend request.

Solution:

We convert the requirements into several components:

[Component: 1] Sign up and log in

[Component: 2] Send Friend request

[Component: 3] Accept friend request

Rapid Application Development (RAD)

Model

- ☐ Extension of Incremental Process Models
- ☐ **Emphasizes an extremely short development cycle (60 to 90 days)**
- ☐ Gathering requirements using workshops or focus groups
- ☐ Uses a component-based construction approach
- ☐ The application should be modularized and addressed by separate RAD teams

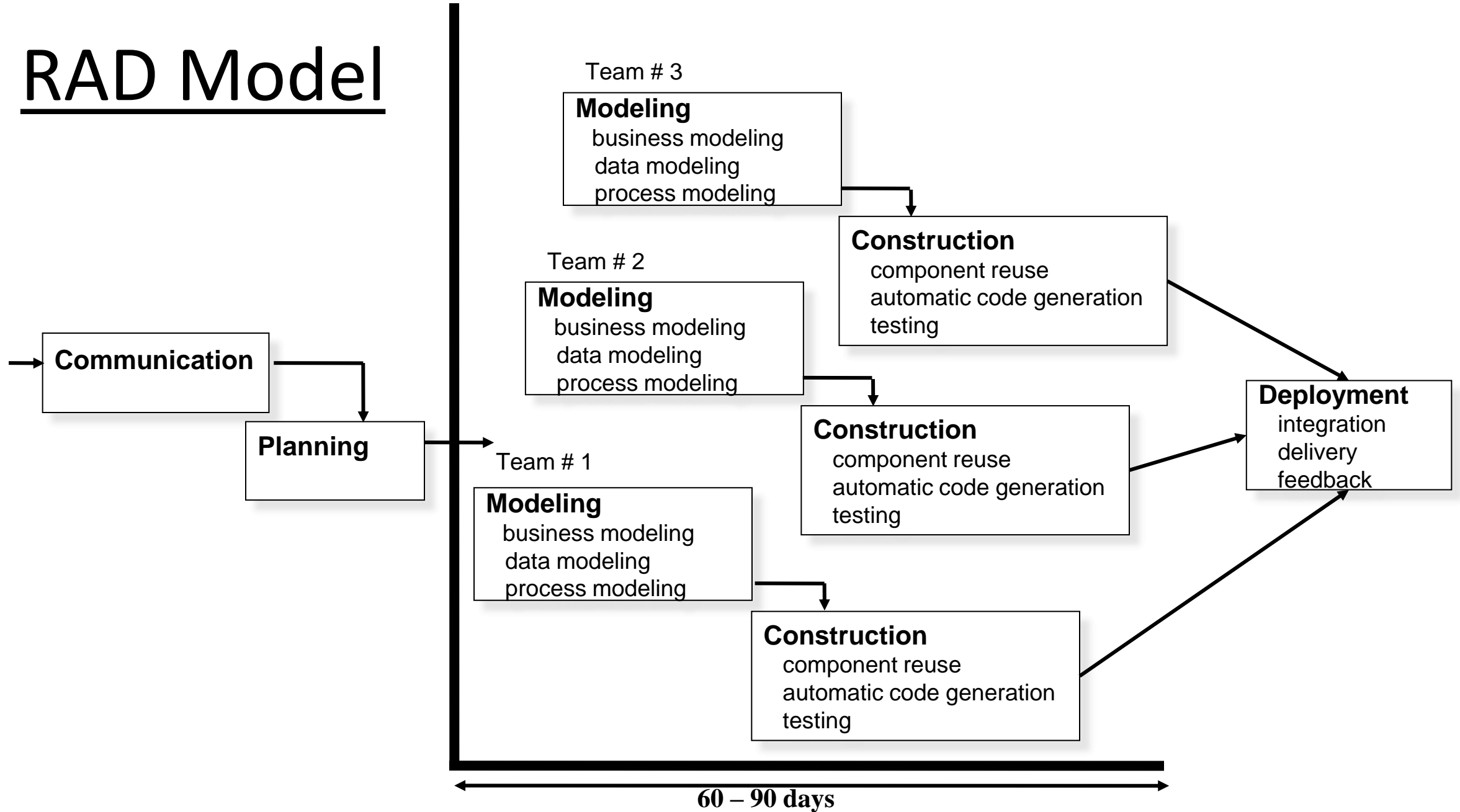
The key objectives of RAD are:

- ☐ High Speed
- ☐ High Quality
- ☐ Low Cost

RAD Model: Phases

- **Business modeling:** The information flow is identified between various business functions.
- **Data modeling:** Information gathered from business modeling is used to define data objects that are needed for the business. (e.g. Entity Relationship, UML)
- **Process modeling:** Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective. (done by different graphical method e.g. flowchart, data flow diagram)
- **Application generation:** Automated tools are used to convert process models into code and the actual system.
- **Testing and turnover:** Test new components and all the interfaces.

RAD Model



RAD Model: When to Use?

- When a system needs to be produced in a short span of time (2-3 months)
- When the requirements are known
- When the user will be involved all through the life cycle
- When technical risk is less
- When a budget is high enough to afford designers for modeling along with the cost of automated tools for code generation

RAD Model: Advantages

- ☐ Reduce development time
- ☐ Increases reusability of components
- ☐ Flexible and adaptable to changes
- ☐ Due to code generators and code reuse, there is a reduction of manual coding
- ☐ With less people, productivity can be increased in short time
- ☐ All functions are modularized so it is easy to work with

RAD Model: Disadvantages

- ❑ Need strong team and individual performances for identifying business requirements
- ❑ It requires highly skilled developers/designers
- ❑ Inapplicable to cheaper projects as the cost of modeling and automate code generation is very high
- ❑ If there is lack of commitment, RAD will fail
- ❑ RAD is based on Object Oriented approach and if it is difficult to modularize the project the RAD may not work well.

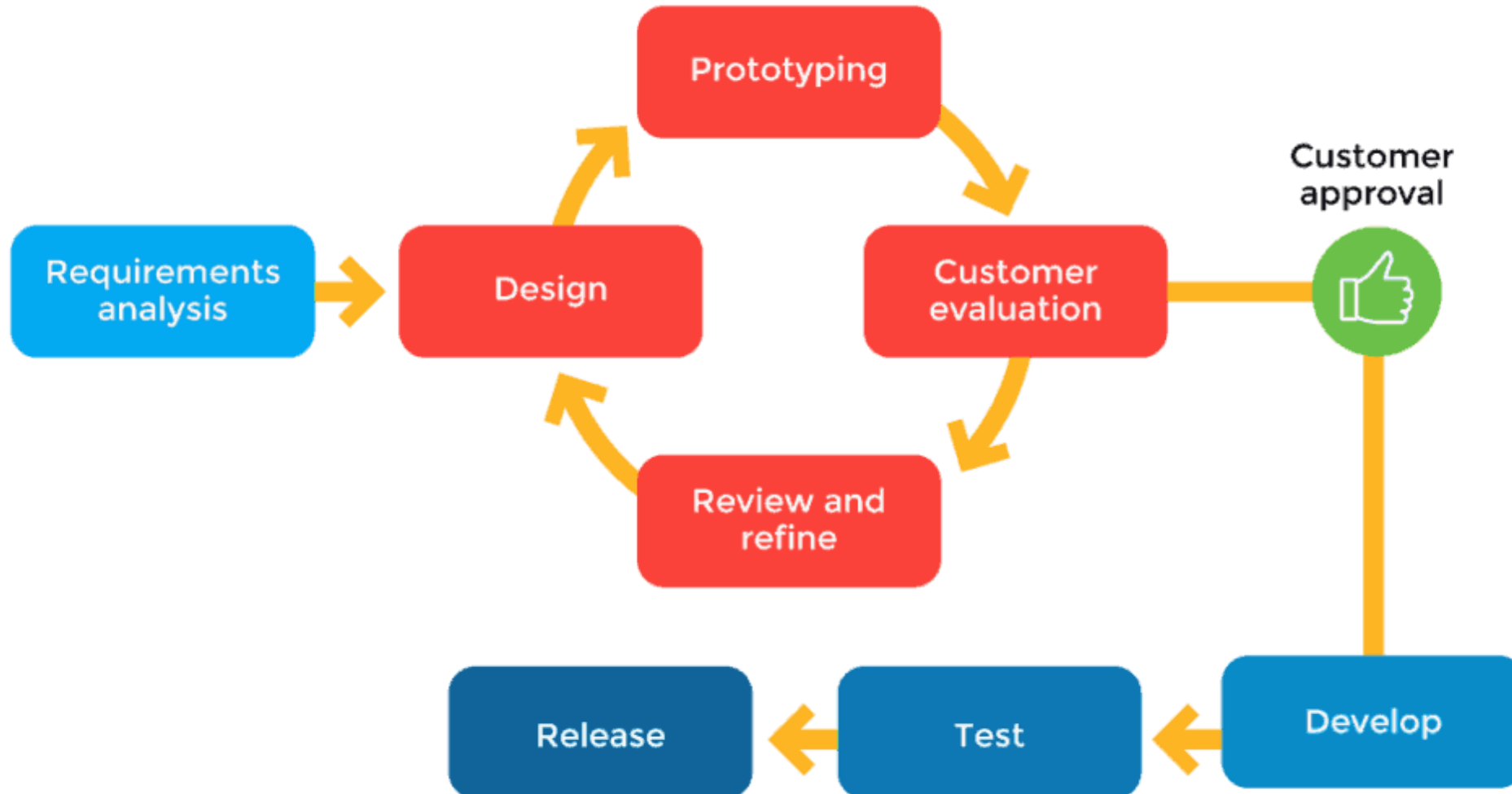
Prototyping Model

- ❑ It's an evolutionary process model
- ❑ Trial-and-error process that takes place between the developers and the users.
- ❑ A prototype is developed based on the currently known requirements.
- ❑ By using this prototype, the client can get an “actual feel” of the system
- ❑ Users thoroughly evaluate the first prototype and note its strengths and weaknesses, what needs to be added and what should to be removed. The developer collects and analyzes the remarks from the

Prototyping Model

- ❑ The first prototype is modified, based on the comments supplied by the users, and a second prototype of the new system is constructed.
- ❑ The second prototype is evaluated in the same manner as was the first prototype.
- ❑ These steps are iterated as many times until the users are satisfied that the prototype represents the final product desired.

Prototyping Model



Prototyping Model: When to Use?

- ☐ Customers don't have a clear idea about what he wants to build
- ☐ Requirements are unstable
- ☐ Desired system needs a lot of interactions between the customers and developers

Prototyping Model: Advantages

- ❑ Customer satisfaction is high as they see some output quickly
- ❑ Errors can be detected much earlier
- ❑ Missing functionality can be identified easily
- ❑ The developed prototype can be reused by the developer for more complicated projects in the future.

Prototyping Model: Disadvantages

- ☐ Prototyping is a slow and time taking process
- ☐ The cost of developing a prototype is a total waste as the prototype is ultimately thrown away
- ☐ Poor documentation because the requirements of the customers are changing
- ☐ The client may lose interest in the final product when he or she is not happy with the initial prototype

Spiral Model

- Combination of sequential and prototype model
- Each spiral can be termed as a loop and each loop is a separate development process in a spiral model
- **Both the customers and developers have a little knowledge about the final output**
- Formation of Spiral shapes happens until customer is fully satisfied
- Customer are highly involved throughout all the phases

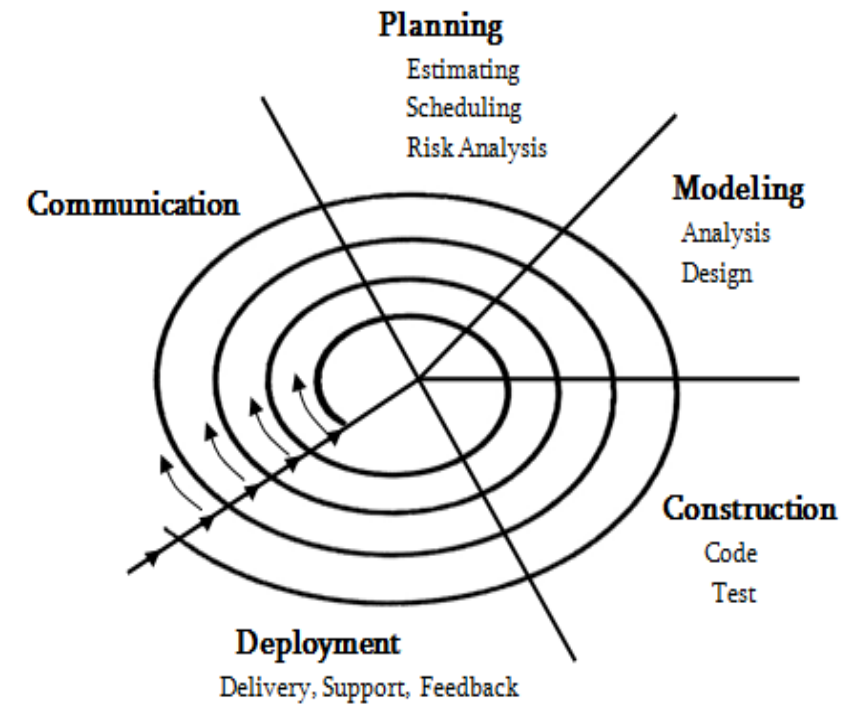


Figure : Spiral Model

Spiral Model

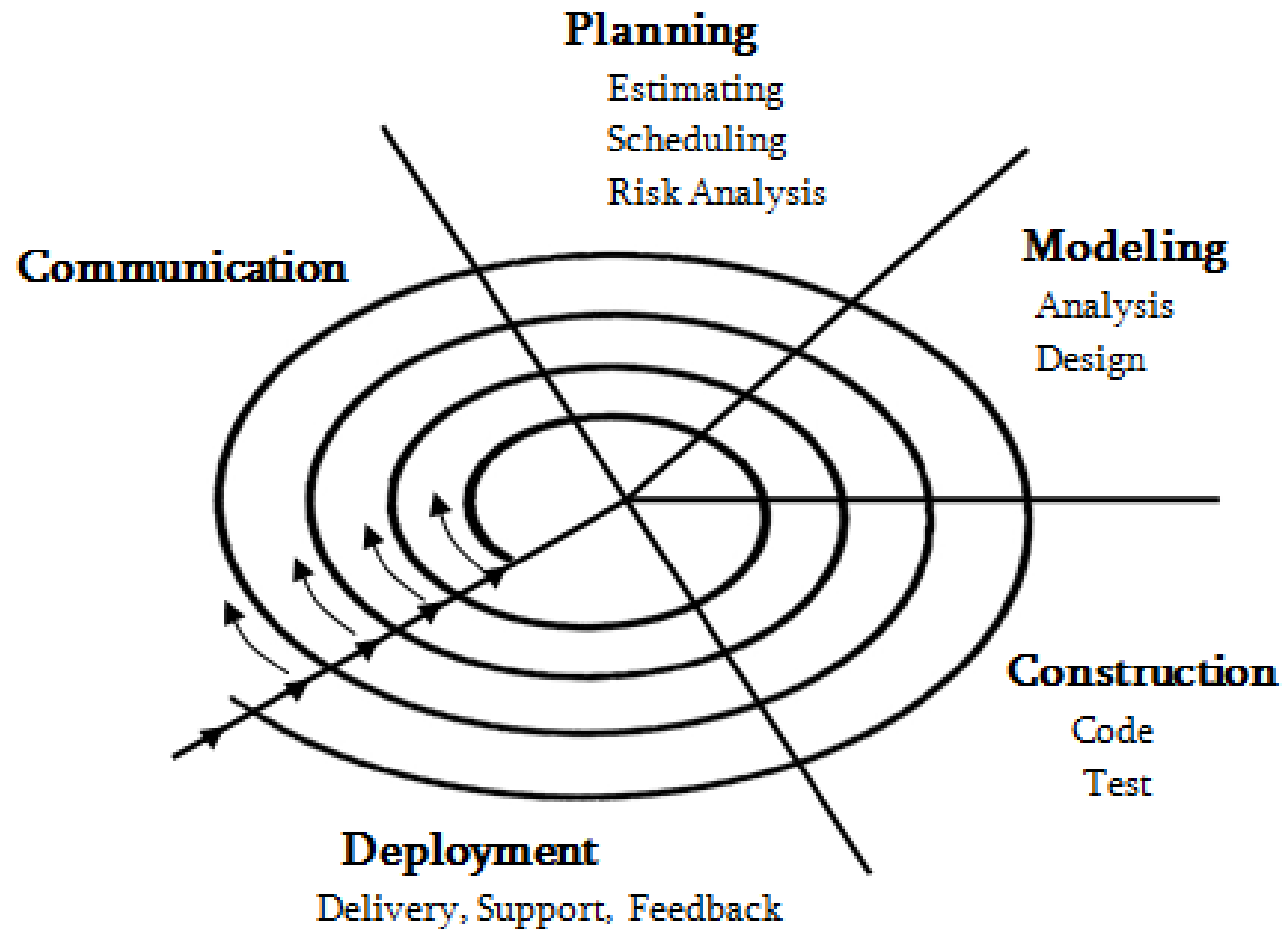


Figure : Spiral Model

Spiral Model: When to Use?

- Both the customers and developers have a little knowledge about the final output
- **Applicable for medium-sized and medium-budget**
- Long-term project commitment because of potential changes

Spiral Model: Advantages

- Low risk as customers are always available
- Changes made to the requirements after development has started can be easily adopted and incorporated.
- Customer satisfaction as the customer will be able to see and evaluate their product in every phase

Spiral Model: Disadvantages

- Time management becomes difficult
- Not suitable for small projects
- Process is complex
- Spiral may go on indefinitely

Concurrent Model

- Also called Concurrent engineering
- Often, it is required to tell the current state of overall project or sub-modules
E.g. **Module A** is currently in Under-development phase.
Module B is currently in Designing Phase.
- The concurrent process model defines a series of events that will trigger transitions from state to state for each of the software engineering activities.
- Provides an accurate picture of the current state of a project.

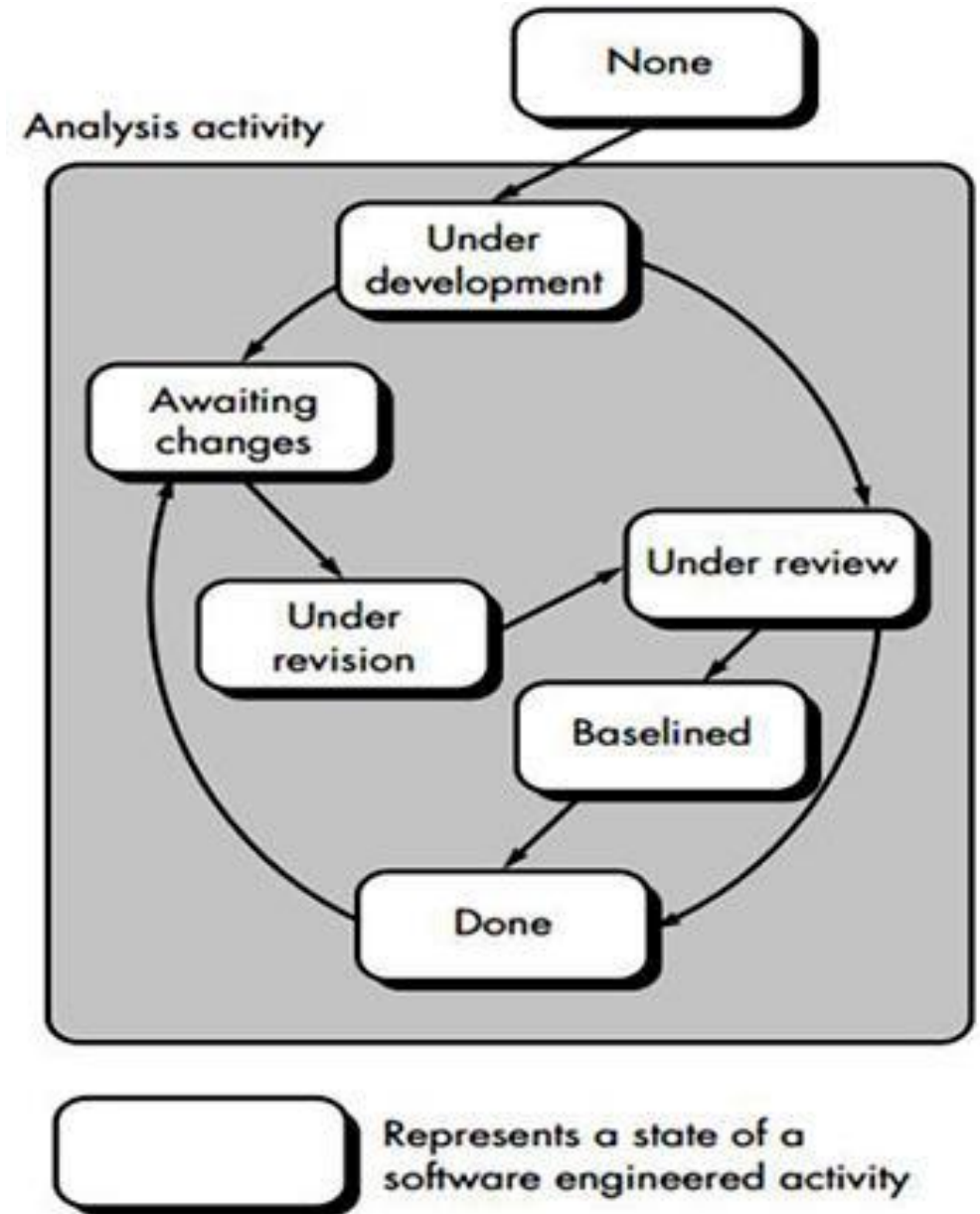
Concurrent Model (Contd.)

The communication activity has completed in the first iteration and exits in the awaiting changes state.

The modeling activity completed its initial communication and then go to the underdevelopment state.

If the customer specifies the change in the requirement, then the modeling activity moves from the under development state into the awaiting change state.

The concurrent process model activities moving from one state to another state



Concurrent Model: Advantages

- This model is applicable to all types of software development processes.
- It is easy for understanding and use.
- It gives immediate feedback from testing.
- It provides an accurate picture of the current state of a project.

Concurrent Model: Disadvantages

- It needs better communication between the team members. This may not be achieved all the time.
- It requires to remember the status of the different activities.