Software Engineering & Project Management

Prof. Yashwant Dongre

Yashwant.dongre@viit.ac.in

Department of Computer Engineering

BRACT'S, Vishwakarma Institute of Information Technology, Pune-48

(An Autonomous Institute affiliated to Savitribai Phule Pune University)
(NBA and NAAC accredited, ISO 9001:2015 certified)

Objective/s of this session

1.
2.
3.

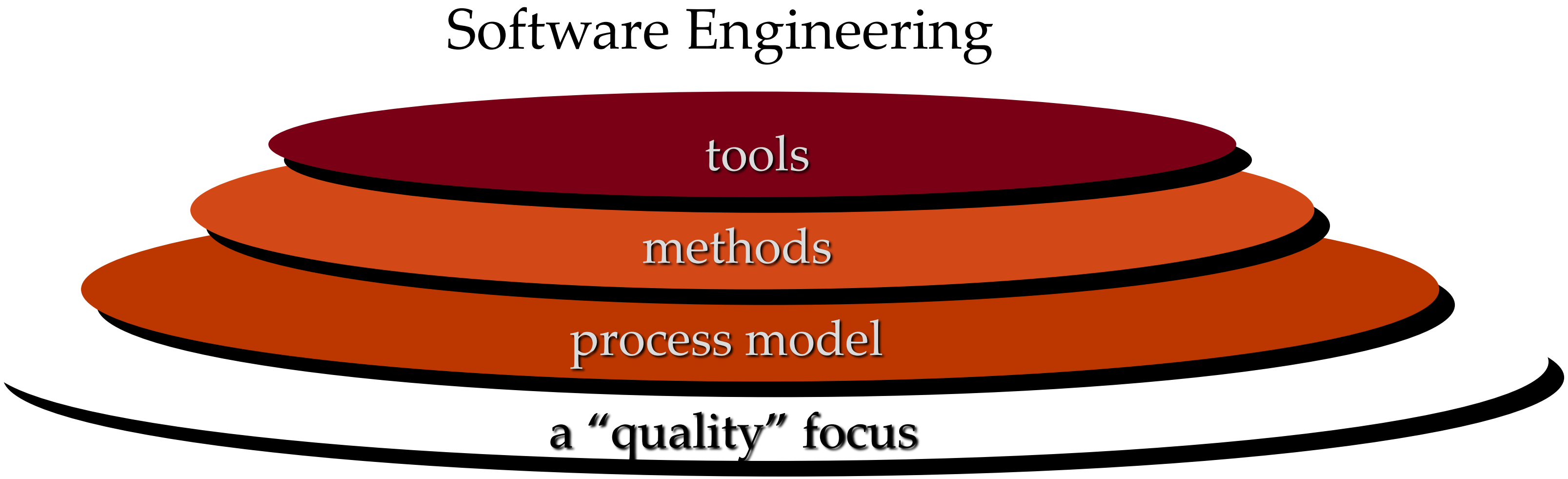
Learning Outcome/Course Outcome

1.
2.
3.

Contents

- Generic Process Model
- Prescriptive Process Model
- Waterfall Model
- Incremental Process (RAD) Model
- Evolutionary Process Model
- Unified Process Model
- Concurrent Model

A Layered Technology



Quality Focus

- TQM, Six Sigma gives continuous process improvement culture.
- Leads to the development of increasingly more effective approaches to software engineering

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

- Process patterns define a set of activities, actions, work tasks, work products and/or related behaviors
- A template is used to define a pattern
- Typical examples:
    - Customer communication (a process activity)
    - Analysis (an action)
    - Requirements gathering (a process task)
    - Reviewing a work product (a process task)
    - Design model (a work product)

Personal Software Process (PSP)

- Recommends five framework activities:
  - ➢Planning
  - ➢High-level design
  - ➢High-level design review
  - ➢Development
  - ➢Postmortem
- Stresses the need for each software engineer to identify errors early and as important, to understand the types of errors

Purpose

- To build computer software.
- Process may be haphazard, ad-hoc, may change on daily basis, may not be efficient, effective.
- Emphasizes personal measurement of both the work product that is produced and resultant quality of the work product.
- Makes practitioner’s responsible for project planning (estimating & scheduling)
- Empowers that practitioner to control the quality of all software work products are developed.

Disadvantage

- Not widely adopted throughout the industry.
- Have more to do with human nature and organizational inertia than strengths & weaknesses of PSP approach.
- Intellectually challenging & demands a level of commitment that is not always possible to obtain.
- Training is lengthy, training cost is high,
- The required level of measurement is culturally difficult for many software people.

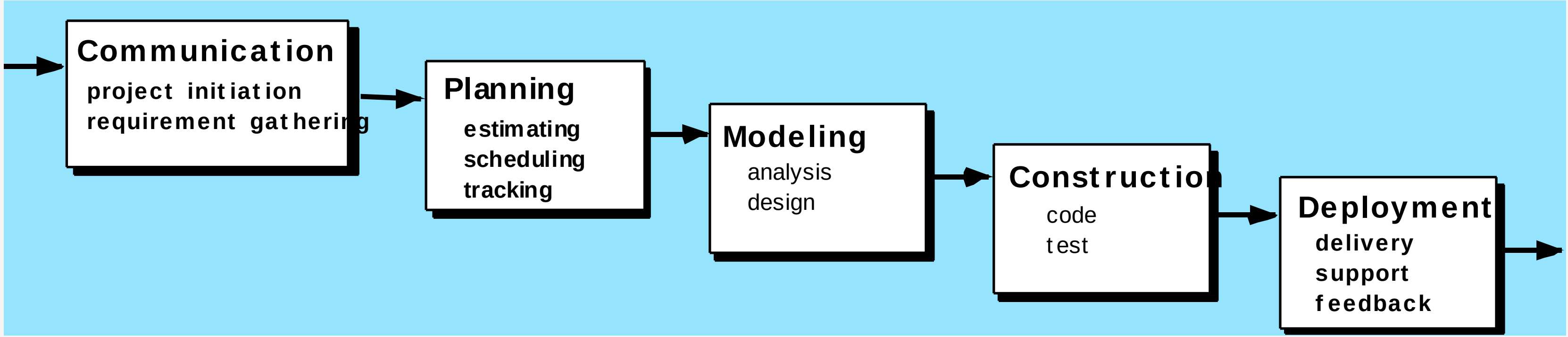
Team Software Process (TSP)

- Each project is “launched” using a “script” that defines the tasks to be accomplished
- Teams are self-directed
- Measurement is encouraged
- Measures are analyzed with the intent of improving the team process

Purpose

- Build a self-directed project team that organizes itself to produce high quality software.
- A self directed team has a consistent understanding of its overall goals & objectives.
- It defines roles & responsibilities for each team member; tracks quantitative project data; identify a team process that is appropriate for project & a strategy for implementing the process; defines local standards; assesses risk; tracks, manages, reports project status.

The Waterfall Model



Communication
project initiation
requirement gathering

Planning
estimating
scheduling
tracking

Modeling
analysis
design

Construction
code
test

Deployment
delivery
support
feedback

Theory

- 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

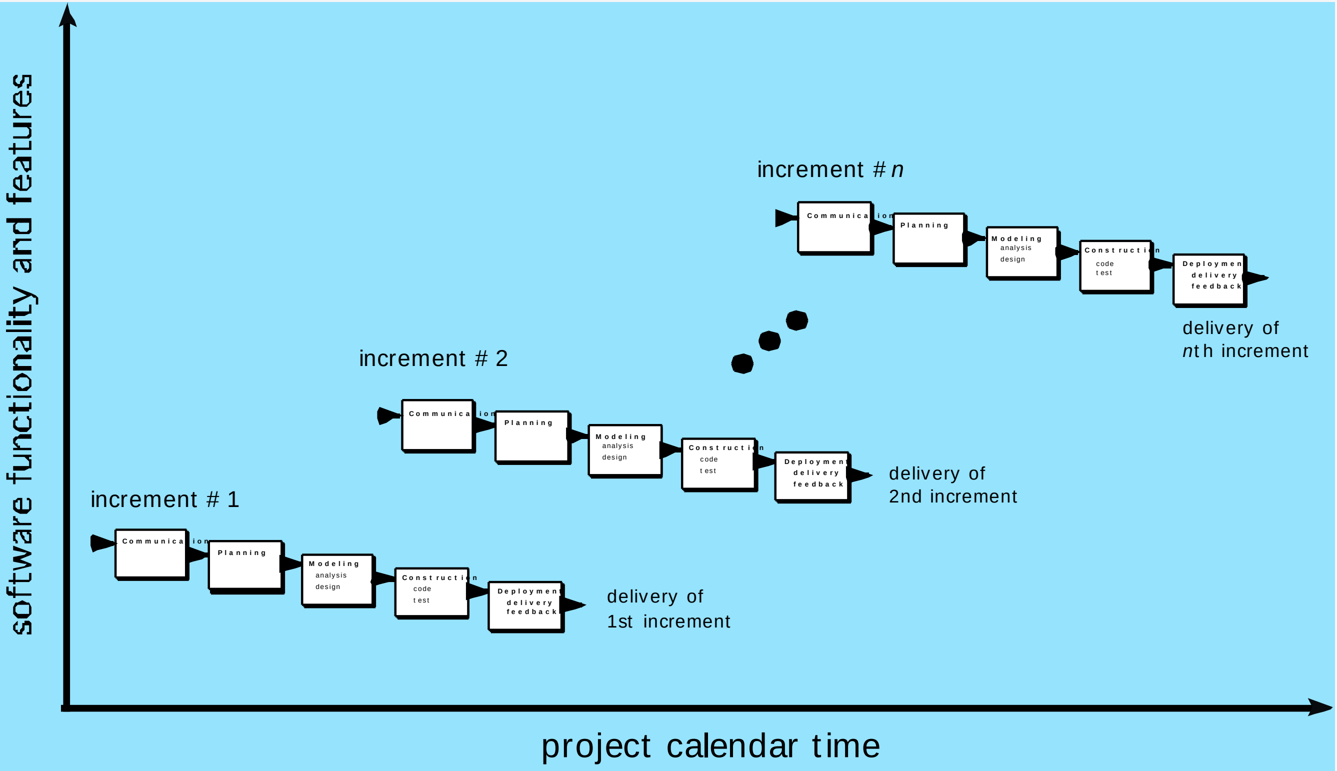
Advantages

- 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



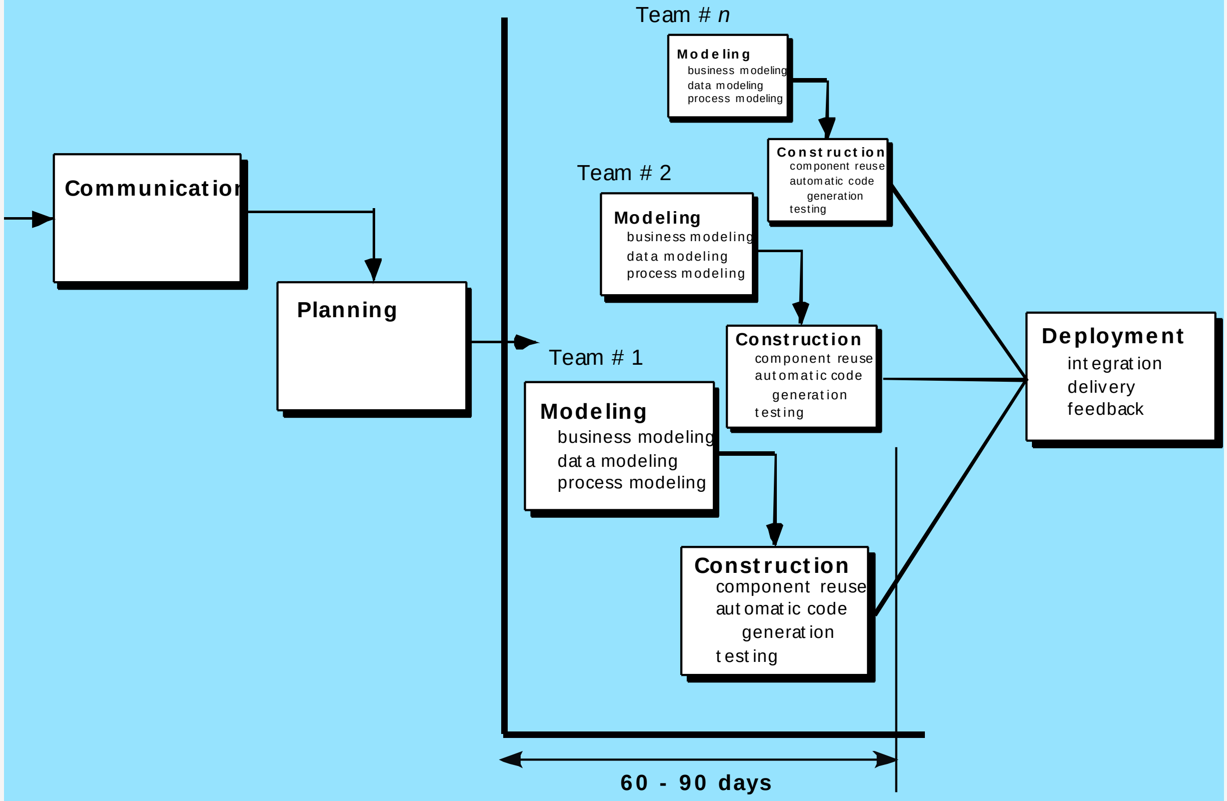
Theory

- 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

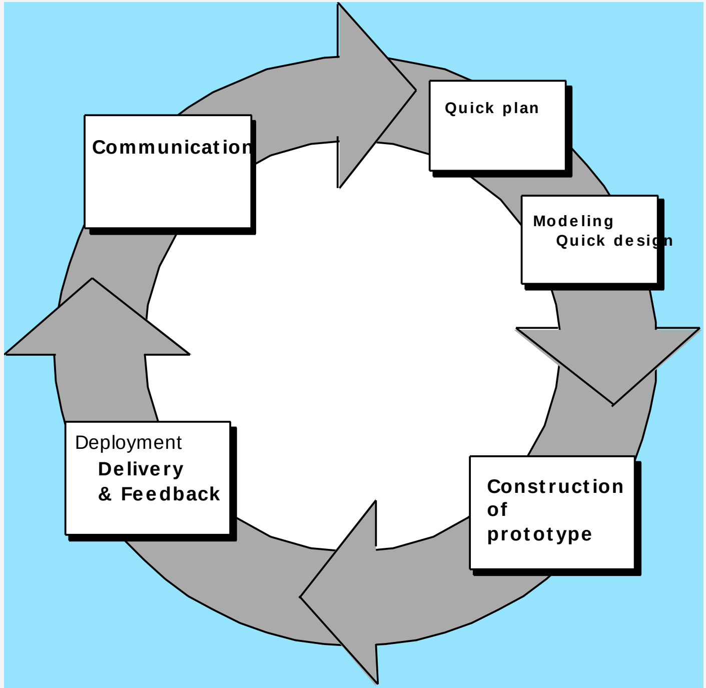
Advantages

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

Disadvantages

- 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.

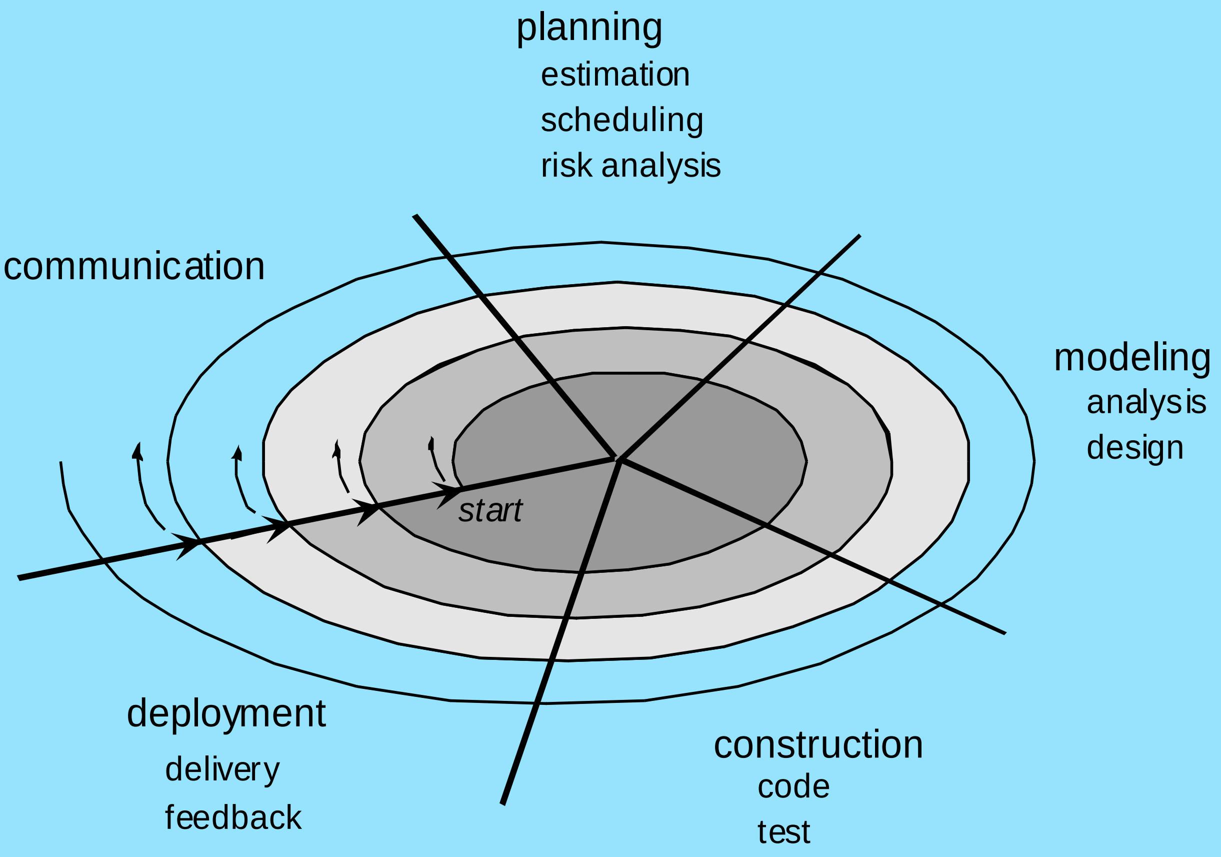
Advantages

- Minimizing technical risks.

Disadvantages

- It may lead to indiscipline of development

Evolutionary Models: The Spiral



Theory

- Meta Model for software development processes.
- Model couples the iterative nature of the prototyping model and the controlled, systematic aspect of the waterfall model.
- It is evolutionary software process model.

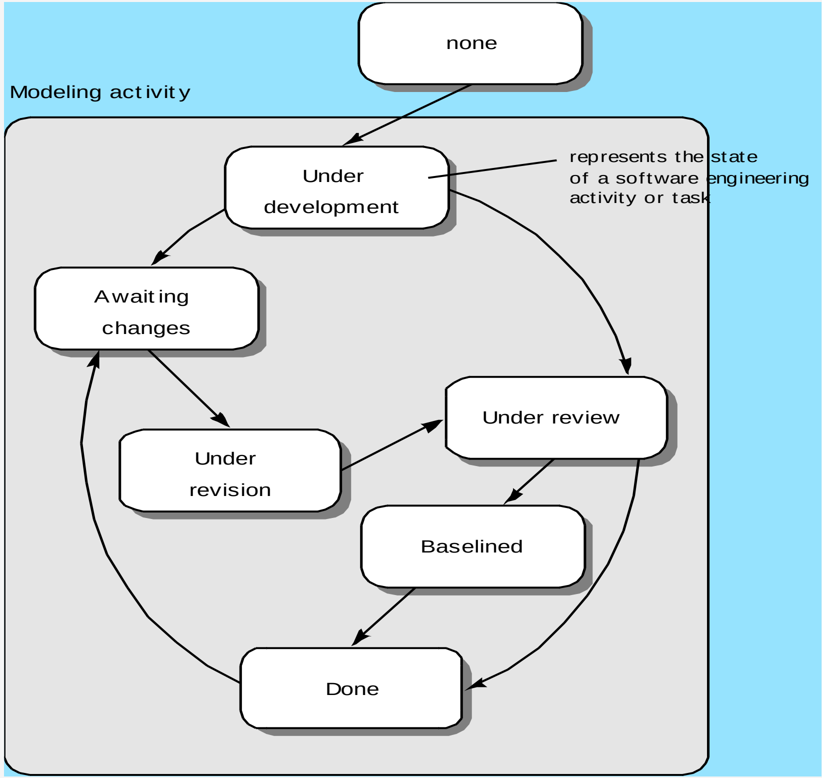
Advantages

- Introduces the element of risk analysis.
- It is more realistic because real world engineering requires considerable iteration.

Disadvantages

- It requires considerable expertise in terms of risk assessment and project management.

Evolutionary Models: Concurrent



Instructions/Guidelines/References

1.
2.
3.
4.

Wrap up and related outcomes

1.
2.
3.
4.

To be discussed next time

Part –I

- Engineering as a Profession.. How does it different from other profession?
- Roles and responsibilities of engineers
- Significance of ethics in engineering

Part-II

- Code of Ethical conduct for an engineer
- Ethics, Law and Moral form engineers perspective
- Case studies/examples of ethical reasoning …closely related to business ethics

Part-III

- E-mail Etiquettes
- Telephone Etiquettes
- Social Etiquettes

VISHWAKARMA INSTITUTES VI

Thank You