Chapter3: Process Models.

Building software is an iterative social learning process.

- Roadmap than one follows to build a software is software process.

The details of process depends on the type of project
eg: avionics, webbased systems etc.

def: work product: program, documents & data produced as a result of a process.

eg: Educational institute as an example of process, product and quality (affiliations, accreditations etc.)

def: Software process: A framework for activities, ~~tasks~~ actions and tasks that are required to build a high quality software.

SE involves process, tools and people

Process is measured by quality, timeliness and long-term

Prescriptive models. what do they prescribe?

- set of framework activities (process elements) (communication, modeling, construction and deployment) — generic framework activities for all process models
- SE actions
- tasks
- work products
- quality assurance
- change control mechanisms
- workflow: how process activities interact with one another

1. Waterfall model: Suggest a systematic, sequential approach to software development

Works well when requirements are well understood and workflow is in a linear fashion.

Criticism:
- Projects rarely follow sequential flow
- Requirements are not stable
- Customers see the working product very late in the project time-span.

2. Incremental Process Models

Incremental Model applies iterative implementation of elements of waterfall model.

There are multiple sequences (iterations) of waterfall model and each (iteration) sequence produce deliverable increments.

the first iteration delivers core product that customer can used and evaluate.

Advantages

- ✱ If customer insists on impossible deadline then deliver only first increment by that deadline
- ✱ Increments may be planned to avoid (manage) technical risks

3. The RAD model : Rapid Application Development

It has a rapid waterfall development cycle ~~toi~~ using (60-90 days)

Component-based construction approach

Requires well-understood requirements

After communication and planning activities, Modeling and construction are performed by multiple teams in parallel. Construction emphasises use of existing components and automated code generation.

Implemented modules from multiple teams ~~or~~ is integrated and deployed. Modeling, construction & deployed typically takes 60-90 days.

Drawbacks:

- • large projects require more human resources for forming parallel ~~teams~~ RAD teams.
- • Requires committment from developers as well as customers during development
- • The ~~produ~~ project should be modular so that team can work in parallel
- • Performance can be limited due to system components are already built (ready made)
- • Cannot apply for projects with high technical risks
- • Project cannot wait for a new technology to arrive

4. Evolutionary Process models :

It is iterative in nature. Complex versions of the product are developed during each iteration.

when product evolves over a period of time but situation demands a working product early, then use this process model

use when (Current requirements are known but future requirements are very uncertain)

~~Types of evolutionary process models.~~

~~a. Prototyping.~~

Types of evolutionary process models:

a. Prototyping :
b. ~~Spn~~ Spiral model :
c. Concurrent development model :

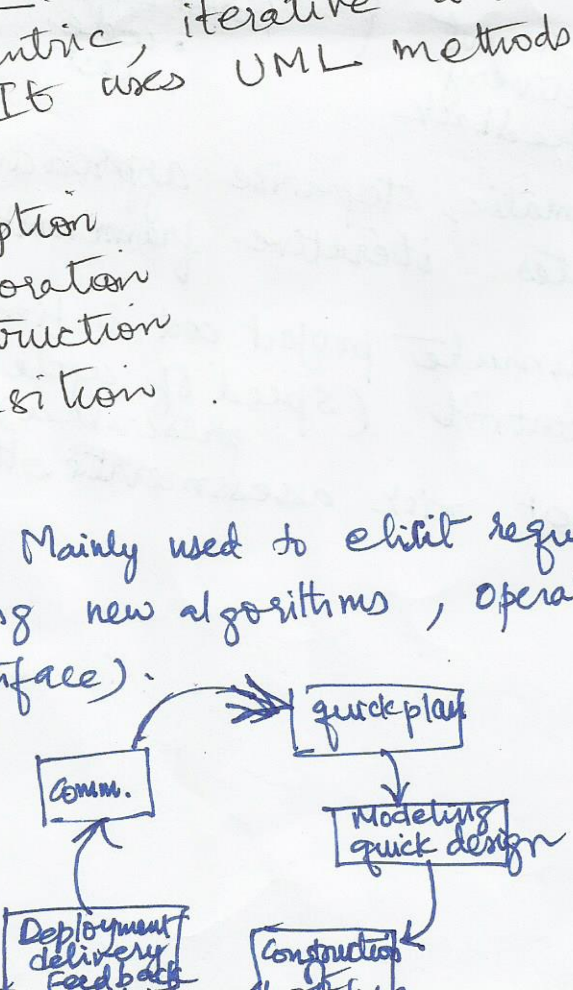
Unified Process : It is an "use-case driven, architecture centric, iterative and incremental" in nature. It uses UML methods and tools

Phases :

- Inception
- Elaboration
- Construction
- Transition.

Prototyping : Mainly used to elicit requirements, understand effects of using new algorithms, operating system or HCI (User Interface).

Prototypes are not functional systems



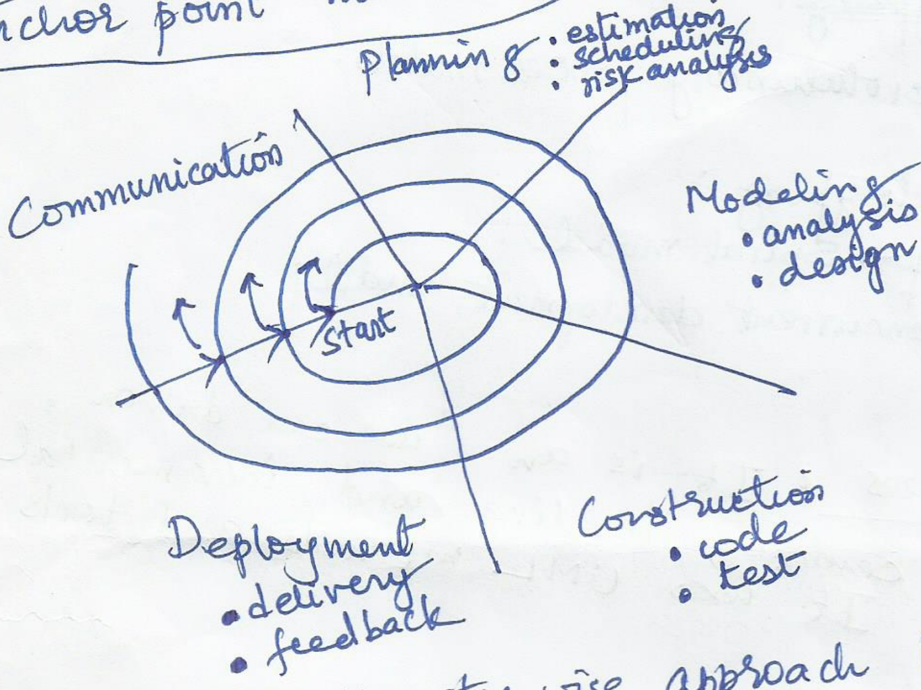
- Ideally prototype must be thrown away
- In reality prototypes may evolve into functional systems

Spiral Model

Proposed by Barry Boehm in 1988. It is a

It is an evolutionary S.P model. It is a risk-driven process model. It is incremental (grows systems degree of definition and implementation) and iterative in nature.

It is cyclic (with risk assessment) and has anchor point milestones (work products and conditions)

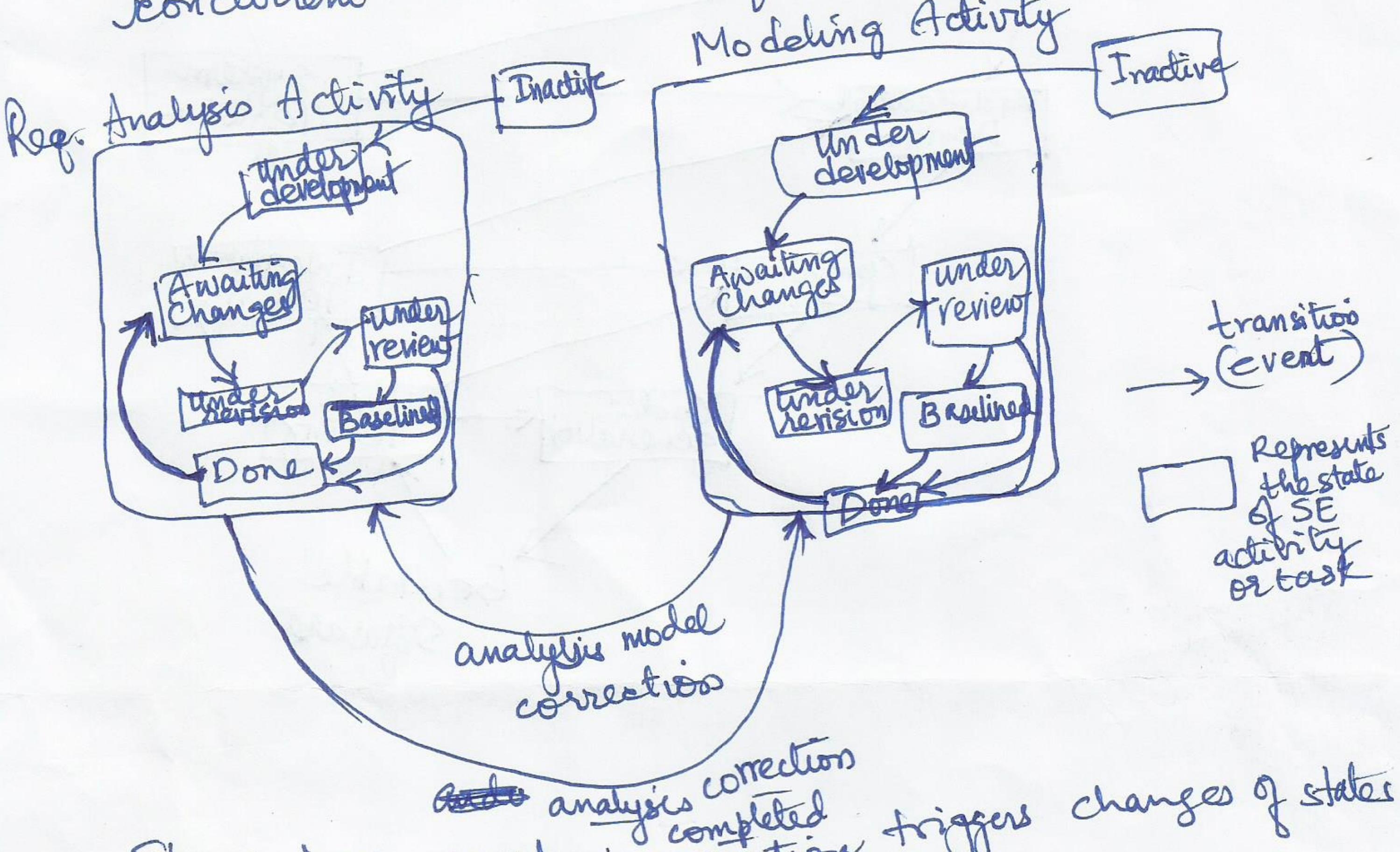


It is a systematic, step-wise approach like water fall but incorporates iterative framework

- Hard to estimate project cost & time
- Hard to control (speed of cycles ~~not~~ is not prescribed)
- Require expert risk assessments otherwise it fails.

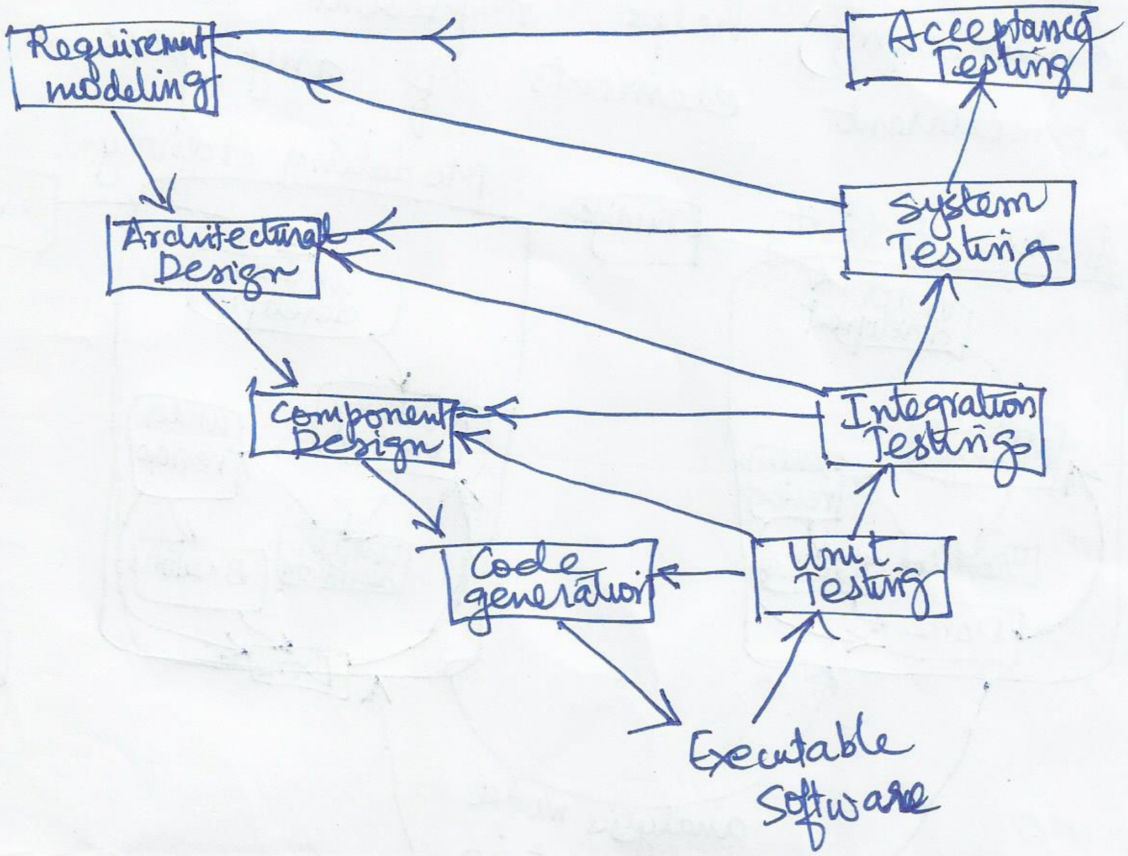
Concurrent models.

The concurrent development model (concurrent engineering) helps represent iterative and concurrent elements of any process model



Shows how analysis correction triggers changes of states

V-model : this is a variation of waterfall model. Associates QA activities with framework activities.



* When is waterfall model applicable?

- Requirements of problem are reasonably well understood
- Work flow from communication through deployment is reasonably linear manner
- Existing system is already present and needed to be expanded.
- New products may be developed only if requirements are well known and stable.

* When is incremental process model (linear sequences are applied in staggered manner) applicable?

- Requirements are well defined but process flow is not linear in manner
- Compelling need to provide limited functional software quickly and then refine & expand it in later releases.
- Staffing is not available to deliver complete product

* What is the difference between prototyping and incremental/evolutionary models?

- Prototyping delivers low quality, barely operational model but incremental model delivers a functional but a stripped down version ~~[illegible]~~ of product.
- Prototyping ~~[illegible]~~ delivers only a ~~evaluation~~ product that can be evaluated but not actually used (operational). Incremental model delivers both an operational product as well as an evaluation product to customers.

* when is a RAD (Rapid Application Development) model applicable?

- Requirements are well known and limited (constrained)
- Components to construct the product are available
- Rapidly develop a fully functional system in short duration (eg: 60-90 days)
- Requires ~~to~~ hire more people (parallel teams)
- Strong committment from customers and developers
- System is modular in nature
- technical risks are not very high

When are evolutionary process models applicable?

- tight market deadlines require a core product (whose requirements are well understood) be developed
- It is not clear how product will evolve (extensible) in the future

When to use prototyping?

Prototyping paradigm:

Requirements are fuzzy

- detailed input, output & processing is not defined
- developer is unsure of efficiency of algorithm or adaptability of OS
- developer is not sure about human-machine (HCI) interface

When is spiral model applicable?

The spiral model

- large-scale systems are required
- Requirements are not well understood
- Risks are high - requires expert risk analysis
- Should not be fixed budget

Process should focus on flexibility, extensibility and speed of development.

Customer satisfication is ultimate test for product quality.

Modern process model should address the modern realities of software development

- continuous change
- evolving systems
- tight time lines