GANPAT UNIVERSITY

U. V. PATEL COLLEGE OF ENGINEERING

2CEIT502

SOFTWARE ENGINEERING

UNIT 2

PROCESS MODELS

(SOFTWARE DEVELOPMENT LIFE CYCLE)

Prepared by: Prof. Ravi Raval (Asst. Prof in C.E Dept. , UVPCE)

Unit 2: Process Models

Contents:

- What is software process?
- What is SDLC?
- Prescriptive Models
- The Waterfall model
- Incremental process models
- Evolutionary Process Models
- Comparisons of Models
- Selection of model for development
- The Unified process

Unit 2: Process Models

What is software Process?

A process is a collection of activities, actions and tasks that are performed when some work product is to be created.

Activity: An activity strives to achieve a broad objective (e.g communication with stake holders) and is applied regardless of the application domain, size of the project, complexity of the effort, or degree of rigor with which software engineering is to be applied.

Action: An action (e.g. an architecture design model) encompasses a set of tasks that produce a major product.

Task: A task focuses on small, but well-defined objective (e.g. conducting a unit test) that produces a tangible outcome.

Unit 2: Process Models

Five activities under process framework:

(1) Communication

(2) Planning

(3) Modeling

(4) Construction

(5) Deployment

Software engineering process framework activities are complemented by a number of umbrella activities.

(1) Software project tracking and control.

(2) Risk Management

(3) Software quality Assurance

(4) Technical Reviews

(5) Measurement

(6) Software configuration management

(7) Reusability management

(8) Work product preparation and production

Unit 2: Process Models

What is software development life cycle (SDLC)?

SDLC used to facilitate the development of a large software product in a systematic, well-defined and cost-effective way.

Why to follow SDLC?

- Helps to understand the entire process.
- Enhances a structured approach to development.
- Enables planning of resources in advance.
- Enables subsequent controls in advance.
- Aids management to track progress of the system.

Apart from that

Entry criteria and exit criteria has to be specified.

Any model must have atleast 5 phases and atmost 9 phases.

Unit 2: Process Models

So, on an average, any SDLC model can have 7 to 8 phases. These are:

(1) Project initiation and planning OR Recognition of need OR Preliminary Investigation.

(2) Project identification and Selection OR Feasibility Study

(3) Project analysis

(4) System Design

(5) Coding

(6) Testing

(7) Implementation

(8) Maintenance.

Unit 2: Process Models

Now let's explore each phase one by one.

(1) Identify, understand and describe problem fully.

(2) Feasibility study types:

A. Organizational feasibility: How well the proposed system supports the strategic objectives of the organization.

B. Technical Feasibility: Hardware, software and network capability, reliability and availability

C. Economic Feasibility: Cost saving, Increased revenue, decreased investment, increased profits

D. Operational Feasibility: End user acceptance, management support, customer-supplier and government requirements.

(3) Project analysis

(4) System Design

(5) Coding

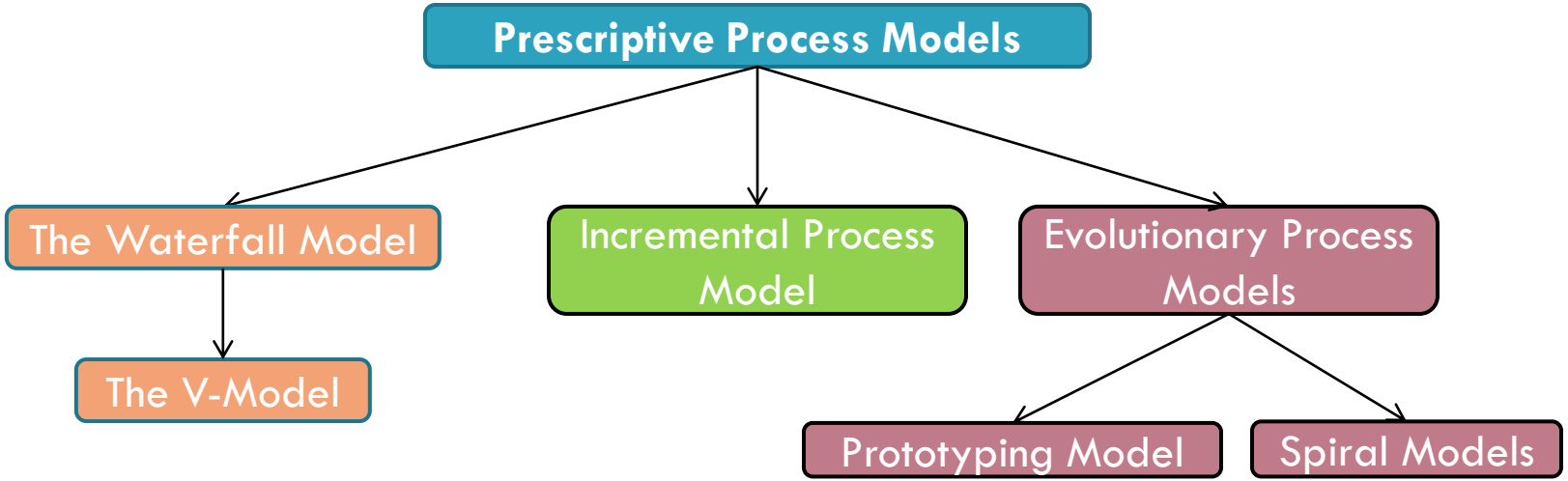
(6) Testing

(7) Implementation

(8) Maintenance: Corrective maintenance, Adaptive maintenance, Perfective Maintenance, Preventive Maintenance

Unit 2: Process Models

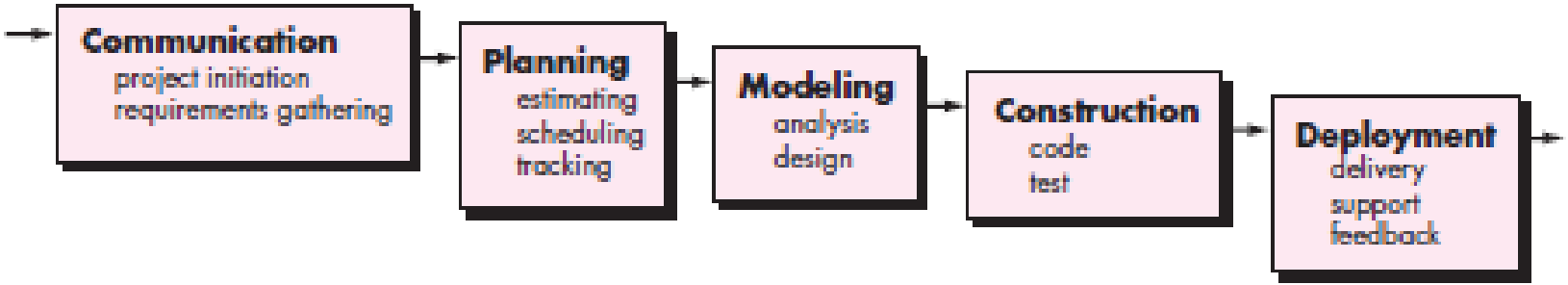
Prescriptive Process Models



Unit 2: Process Models

Prescriptive Process Models:

(1) Waterfall Model

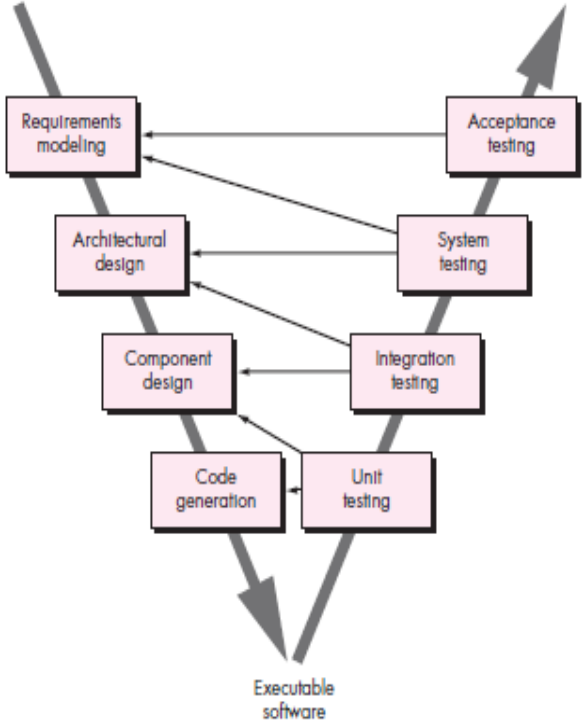


Unit 2: Process Models

Prescriptive Process Models:

(1) The V Model (variation of waterfall)

Advantages:	This model is not suitable for accommodating any change.
It is a linear Model	A working version of the system of is not seen until late in the project's life
It is a segmental model	It does not scale up well to large project
It is systematic and sequential	It involves heavy documentation. Documentation requires at end of each phase.
It is simple one to understand and follow	We cannot go in the backward direction while SDLC performs
It has proper documentations.	There is no sample model for clearly in realization the customers needs.
Disadvantages:	There is no risk analysis
It's difficult to define all requirements at the beginning of a project.	If there is any mistake or error in any phase then it will propagate to the last



Unit 2: Process Models

Prescriptive Process Models:

(2) Incremental Process Model

Each linear sequence produces

Deliverable "increment" of SW

- For ex. Word Processing SW

1st Increment: Basic file

Management

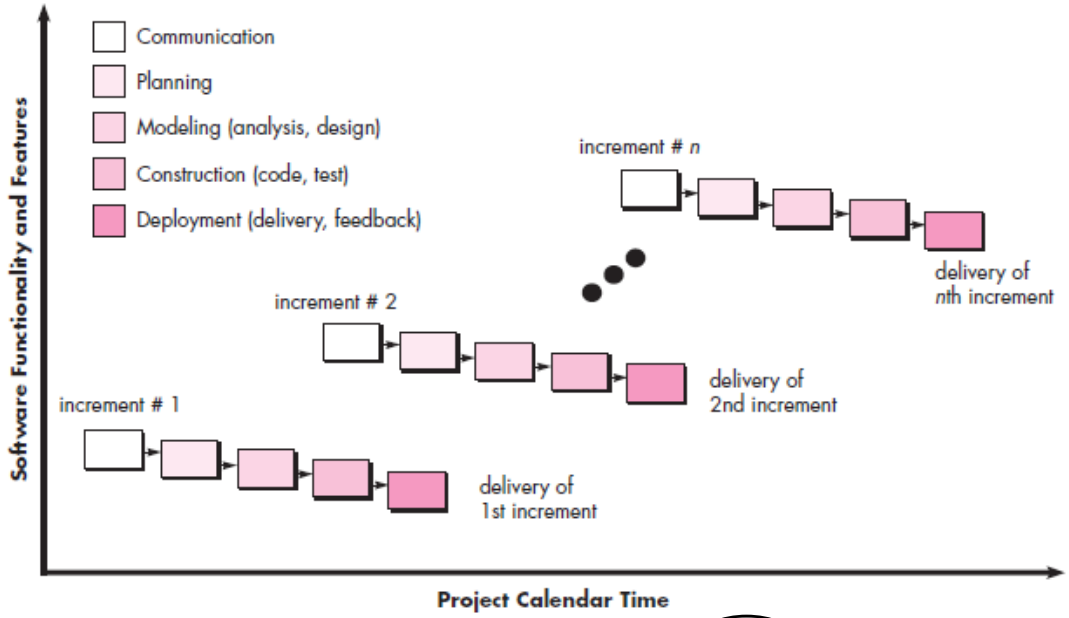
2nd Increment: More sophisticated

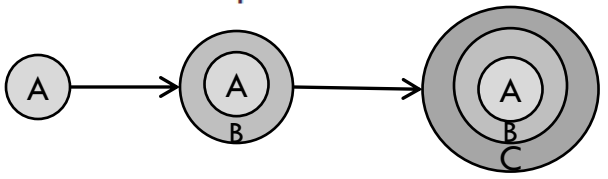
Editing & document production capability

3rd Increment: Spelling and grammar

checking

4th Increment: Advanced page layout capability





Unit 2: Process Models

Prescriptive Process Models:

(2) Incremental Process Model

Advantages:

1. The feedbacks from early increments improve the later stages.
2. The possibility of changes in requirement is reduced because of the shorter time span between the design of a component and its delivery.
3. Users get benefits earlier than with a conventional approach. Early delivery of some useful components improves cash flow, because you get some return on investment early on.
4. Smaller sub-projects are easier to control and manage.
5. Gold plating, that is the requesting of features that are unnecessary and not in fact used is less as uses will know that they get more than one bite of cherry if a feature is not in the current increment then it can be included in the next.
6. The project can be temporarily abandoned if more urgent work crops up.
7. Job satisfaction is increased for developers who see their labours bearing fruit at regular, short intervals

Unit 2: Process Models

Prescriptive Process Models:

(2) Incremental Process Model

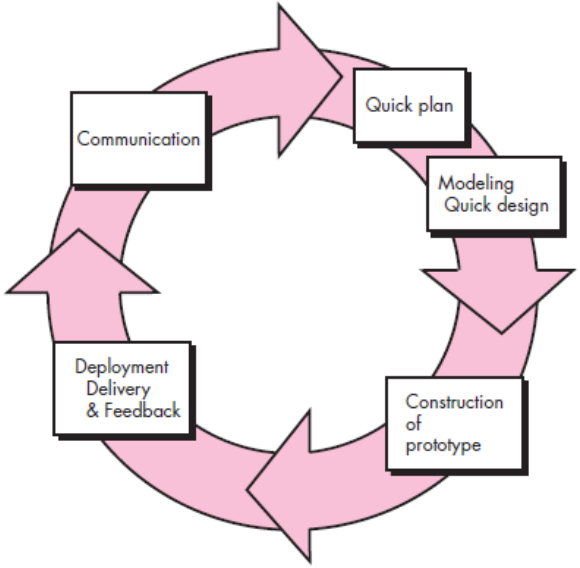
Disadvantages:

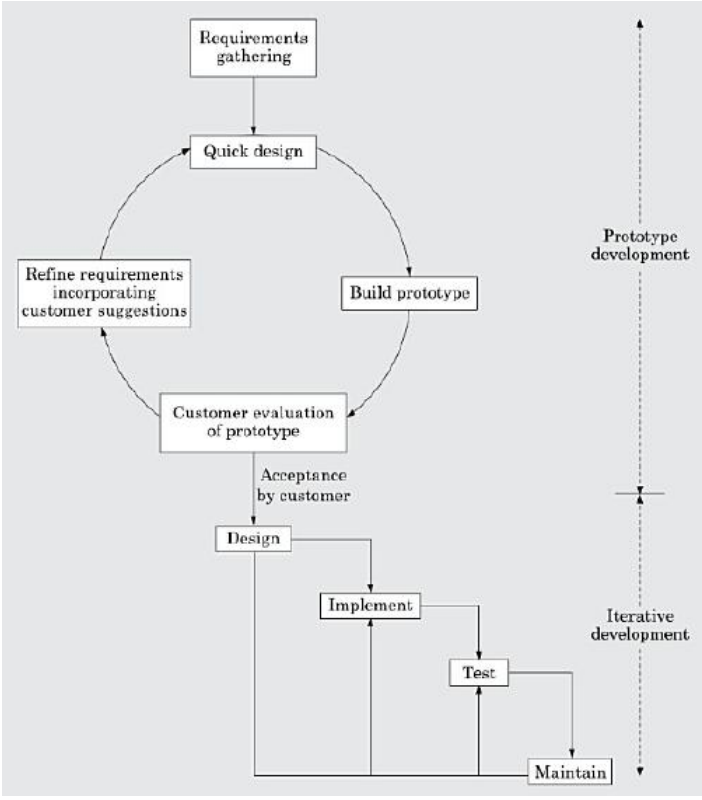
1. Software breakage, that is, later increment may requirement modification to earlier increments.
2. Programmer may be more productive working on one large system than on a series of smaller ones.
3. Some problems are difficult to divide into functional units (modules), which can be incrementally develop and delivered.
4. Testing of modules result into overhead and increased cost.

Unit 2: Process Models

Evolutionary Process Models:

(3) Prototype Model:





Unit 2: Process Models

Evolutionary Process Models:

(3) Prototype Model:

Advantages:

1) Suitable for large system for which there is no manual process to define the requirements
2) Prototyping make requirements more clear and system more transparent.
3) Flexibility in design and development is also supported by the model.
4) System training
5) Quality of software is good
6) User training to use the system.
7) User services determination

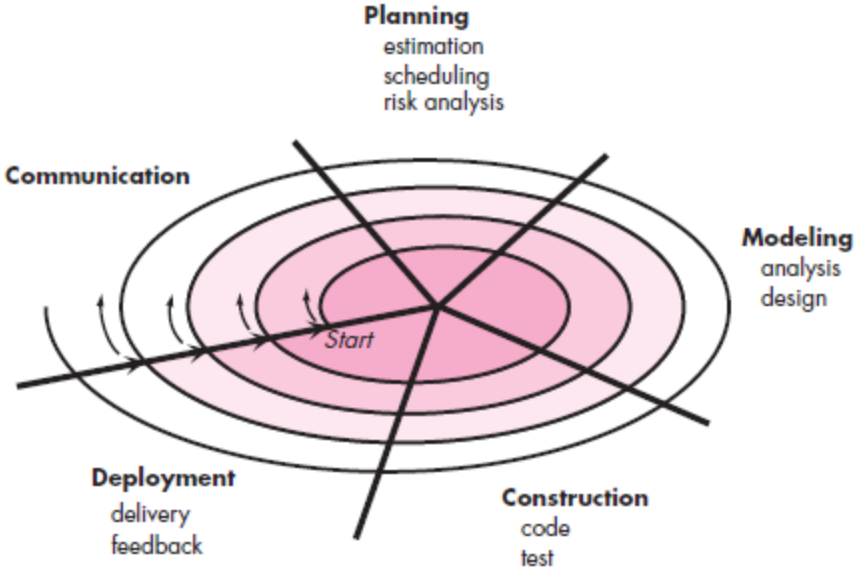
Disadvantages:

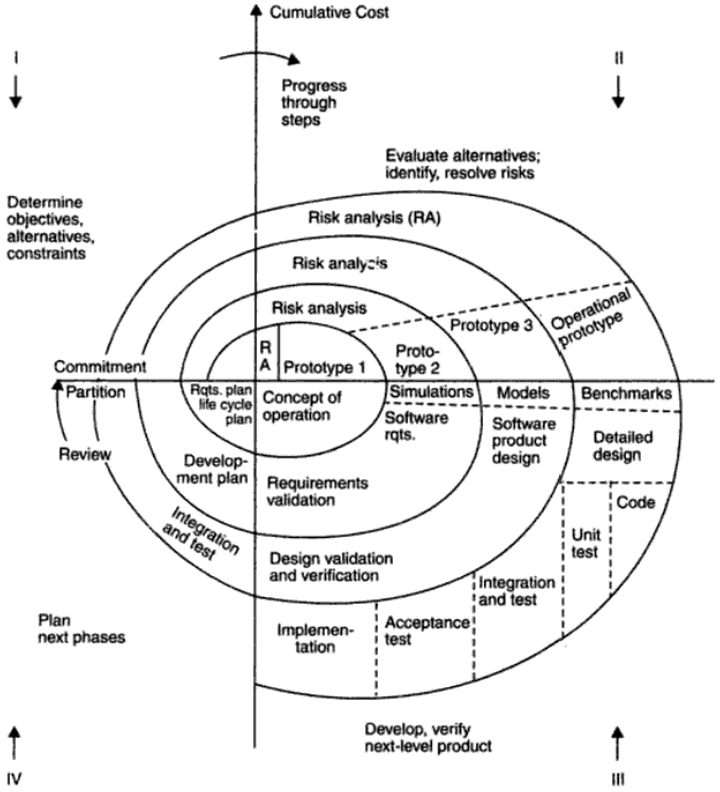
1) The developer often compromises in order to get working prototype quickly. (use of inappropriate OS, algorithms AND/OR programming technology)
2) End users may not like to know the difference between a prototype and a well-engineered fully developed product.
3) If not managed properly, the interactive process of prototype demonstration and refinement can continue for long duration.
4) If end user is not satisfied with initial prototype, he may loose interest in the project.

Unit 2: Process Models

Evolutionary Process Models:

(4) Spiral Model:





Unit 2: Process Models

Evolutionary Process Models:

(4) Spiral Model:

Advantages	Disadvantages
It is a risk driven model	No Strict standards for software development
It's very flexible	No particular beginning or end of particular phase
Less documentation is needed	Different people may find it complex to use
It makes a use of prototyping	Requires expertise risk management & excellent management skills
It is more realistic model for software development.	Not suitable for small project. Sometimes the cost of risk analysis may exceed the actual cost of the project

Unit 2: Process Models

Comparisons of all models:

	Strength	Weakness	Types of Project
Waterfall Model	• Simple • Easy to execute • Intuitive and logical	• All or nothing approach • Requirements frozen early • Disallows changes • Cycle time too long • May choose outdated hardware technology • User feedback not allowed • Encourages requirement bloating	• For well understood problems • Short duration project • Automation of existing manual systems.
Prototyping Model	• Helps in requirement elicitation • Reduces risk • Leads to a better system	• Front heavy process • Possibly higher cost • Disallows later changes	• System with novice users • Best for uncertain requirements

Unit 2: Process Models

Comparisons of all models:

	Strength	Weakness	Types of Project
Iterative / Incremental Model	• Regular/Quick deliveries • Reduces risk • Accommodates changes • Allows user feedback • Allows reasonable exit points • Avoids req. bloating • Prioritizes requirements	• Each iteration can have planning overhead • Cost may increase as work done in one iteration may have to be undone later. • System architecture and structure may suffer as frequent changes are made	• For businesses where time is of essence • Where risk of long project cannot be taken • Where requirements are not known
Spiral Model	• Controls project risks • Very flexible • Less documentation needed	• No strict standards for software development • No particular beginning or end of particular phase	• Project built on untested assumptions

Unit 2: Process Models

Selection of a life cycle model:

How to chose a model for a given software to be developed?

Selection of model based on characteristics of requirement				
Requirements	**Waterfall**	**Prototype**	**Iterative**	**Spiral**
Are requirements easily understandable and defined?	Yes	No	No	No
Do we change requirements quite often?	No	Yes	No	Yes
Can we define requirements early in the cycle?	Yes	No	Yes	No
Requirements are indicating a complex system to be built	No	Yes	Yes	Yes
Selection based on Status of development team				
Less Experience on similar projects	No	Yes	No	Yes
Less domain knowledge (new to technology)	Yes	No	Yes	Yes
Less experience on tools to be used	Yes	No	No	Yes
Availability of training	No	No	Yes	No

Unit 2: Process Models

Selection of a life cycle model:

Selection based on Users participation				
Requirements	**Waterfall**	**Prototype**	**Iterative**	**Spiral**
Users involvement in all phases	No	Yes	No	No
Limited users participation	Yes	No	Yes	Yes
Users have no previous experience of participation in similar projects	No	Yes	Yes	Yes
Users are experts of problem domain	No	Yes	Yes	No
Selection based on type of project with associated risk				
Project is the enhancement of existing system	No	No	Yes	Yes
Funding is stable for the project	Yes	Yes	No	Yes
High reliability requirements	No	No	Yes	No
Tight project schedule	No	Yes	Yes	Yes
Use of reusable components	No	Yes	No	Yes
Are resources (time/money/people etc.) scarce?	No	Yes	No	No

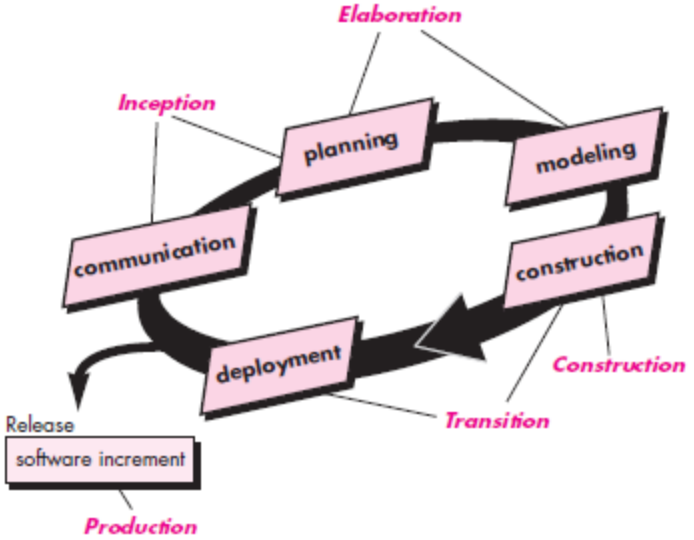
Unit 2: Process Models

The Unified Process:

The **Unified Software Development Process** or **Unified Process** is an iterative and incremental software development process framework.

Five Phases of Unified process:

1) **Inception**
2) **Elaboration**
3) **Construction**
4) **Transition**
5) **Production**




THANK YOU!
ANY QUESTIONS?
memegenerator.net