THADOMAL SHAHANI ENGINEERING COLLEGE

P.G. KHER MARG, T.P.S.-III, BANDRA (W), MUMBAI-50.
ON BEHALF OF
UNIVERSITY OF MUMBAI

CODE NO. _____

(To be filled in by the Candidate)
Candidate's Seat No. : (In figures) | | | | | |

(In words) _____

Examination : _____ Subject : _____

Part / Branch / Sem. Group : _____ Total No. of Supplements _____

Q. No.	1	2	3	4	5	6	7	8	9	10	11	12	TOTAL
Marks Obtained													

Junior Supervisor's Name	Junior Supervisor's Signature with date

BEGIN YOUR ANSWER FROM HERE

Signature of the Examiner

Module - 1. SE.

SE.

→ Software :- It is more than just a prog. code. we can say it is a collection of data or comp. instns which tells the comp. how to work. where as hardware is a physical term from which the sys. is built & it actually performs the work.

→ As I already told you that s/w is a collection of executable programming code, associated libraries & documentations.

→ software when made for a specific requirement is called as software product.

→ computer s/w is the product which is design & built built by the s/w engineers.

→ s/w consist of programs, documents, data (operating procedures).

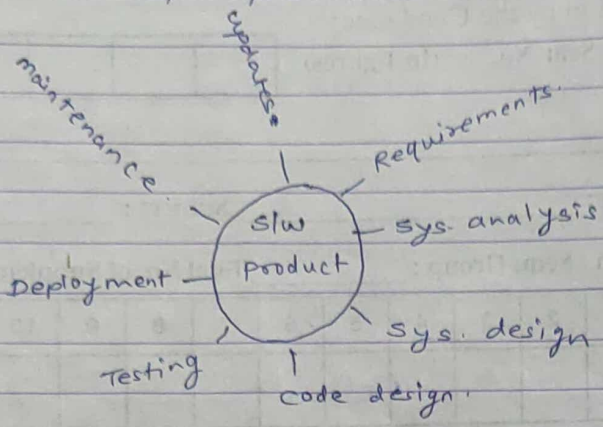
→ What is the work product :-

Software engineers points of view, the work product is set of programs, content (data) & other work products that are comp. s/w.

From the user's point of view, the work product is the resultant information that somehow makes the user's world better.

Def^n:- S/w is ① inst^{ns} when executed provide desired function, features.
② Data st^{re} structure that enable the programs to manipulate info^n.
③ descriptive info^n in both hard copy & virtual forms that describes the operation & use of the programs.

→ Engineering is all about developing products, using well defined, scientic principles & methods.



→ S/w engg. is an engg. branch associated with development of s/w product using well-defined scientific principles, methods & ~~prod~~ procedures. So the outcome of ~~the~~ s/w engg. is an efficient & reliable s/w product.

Role of S/W :-

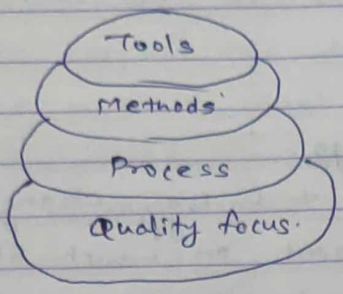
→ Here s/w takes on a dual role. Product & Vehicle/ platform.

Product :- delivers the computing potential embodies by comp. h/w or more broadly, by a n/w of computers that are accessible by local h/w.
s/w is an info^n. transformer - producing, managing, acquiring, modifying, displaying or transmitting info. that can be as simple as a single bit or as complex as a multimedia presentation derived from data acquired from dozens of independent sources.

Vehicle - is used to deliver the product, s/w acts as the basis for the control of the computer (os), the communication of info^n. (n/w) & the creation and control of other programs (s/w tools & environments).

Software characteristics / Nature of s/w.

Layered structure of s/w engg.:-



Here layered approach is divided into 4 layers.

(1) A quality focus :- An we already know that any engg. approach must be based on the quality. So most imp. aspect in soft. engg. is Quality focus.

(2) Process :- This layer is a foundation for SE. This layer holds all the technology layers together & enables the timely development of computer s/w. This layer also forms the base for management control of s/w project.

(3) Method :- This layer provide the "Technical questions" for building s/w. Here method contains array of tasks which include communication, requirement analysis, design modeling, program construction (implementation), testing & support.

(4) Tools :- Tools provide automated or semi-automated support for the "process" & "method". Here Now a days tools are integreated so that info' created by one tool can be used by another.

Activities of s/w engg. :- It includes many activities like.

→ Understanding the user requirements is the first most imp. activity in s/w engg.

→ According to the requirements, the process of designing the sys. & taking decisions is implemented, which leads to successful completion of development of product.

→ constructing s/w product based upon designs & decisions made earlier is the next activity.

→ Verify the performance & quality of s/w, testing the s/w

product is essential after it is constructed by an engineers

→ Last step is to provide maintenance for s/w product which is deployed to the customer.

Need of s/w engg.

- We know that as technology changes, the user requirements & environment on which s/w is working also changes. That's why every organization is based on the s/w engg. principles used by that organization.
- Implementing & managing large size of s/w, programmer requires a specific method to modularize the tasks so that size of s/w can't harm the s/w quality.

→ SE also provides methodology for implementing complex s/w systems with high quality.

→ With any standard method, it is difficult to find defects in the product & correct them as soon as possible. SE provides this functionality.

→ It also provides a way in which s/w sys. can be able to scale as needed in future.

characteristics of s/w :-

1) Software is developed or engineered, it is not manufactured.

→ Here s/w development & h/w manufacturing both activities requires the construction of a "product" but the approaches / methods are different.

→ In both the activities, good design is imp. for high quality. But in the process of manufacturing hardware there may be quality problem which may not present in case of s/w.

→ The main aim of both the activities is construction of a "product", but the perspectives are not same.

② Software doesn't "wear out"

- In case of H/w, failure rates are high as compared to s/w early in its life. Basically these failures are concerned with design or manufacturing defects.
- It is possible to correct the defects & drop the failure rate to a steady - state level for some time period.
- After some time period, there is again increase in the failure rate as there are environmental effects of dust, vibrations, mishandling, high or low temperature etc.
- In simple words, the h/w begins to wear out.

- In case of s/w,
  software does not have any risks of environmental effects.
- In early life of s/w, there may be high failure rates because of undiscovered defects. These However, these can be corrected.
- In it life change is an integral part of s/w. whenever any change is made, there is possibility of introduction of errors which increase the failure rate.
- Before s/w gets consistent state by corrections, any new change get introduced which again increases the failure rate.
- Gradually, the lowest failure rate level starts to increase which indicates that the s/w is deteriorating due to change.

③ Although the industry is moving toward component-based assembly, most s/w continues to be custom built.

- component reusability is an imp. aspect in s/w industry.
- It is responsibility of s/w engineer to design & implement a s/w component in such a way that it should be reused easily in many different programs.
- By using these reusable components s/w engineers can develop new appl.ns from existing components.

④ s/w is intangible.

⑤ Software is easy to reproduce.

⑥ The industry is labor-intensive.

Software applications :-

s/w become an integral part of the fields of human life. We name a field & find usage of s/w in that field. Here s/w appl^ns are grouped into follow areas.

1) sys. ~~stof~~ s/w :- Infrastructure s/w like compilers, editors, OS, drivers etc. comes under this category. Basically sys. s/w is a collection of programs to provi service to other programs.

2) Real time s/w :- These s/w are used to monitor, control & analyze real world events as they occur. e.g. weather forcasting s/w. Such s/w will gather & process the status of temperature, humidity & other environmental parameters to forecast the weather.

3) Embedded s/w :- This type of s/w is placed in ROM of the product & control the various functions of the product. Here the product could be an aircraft, automob security sys., ~~signally~~ signalling sys., control unit of power plants etc. The embedded s/w handles h/w components & is known as intelligent s/w.

4) Business s/w :- The s/w designed to process business appl^ns is called business s/w. This is the largest appl^n. area. Business s/w could be payroll, file monitoring sys., employee mgmt, account mgmt. Busines s/w helps us to take decisions based on available data. ERP, MIS & such other s/w are popular exam of business s/w.

5) Personal comp. slw :- These are used in personal computers like word processor, comp. graphics, multimedia & animating tools, database mgmt, comp. games etc.

6) AI slw :- It makes use of non-numerical algorithms to solve complex problems - eg. are expert systems, signal processing sys slw, artificial neural network etc.

7) web based slw :- The slw related to web applns come under this category e.g. HTML, Java, Perl, CGI etc.

8) Engg. & scientific slw :- Here scientific & engg. appln are grouped. Huge computing is required to process data. e.g. are CAD/CAM package, MATLAB, circuit analyzers etc.

Product & process :-

Product :- What is delivered to the customer is called a product. It include source code, & specification document, manuals, documentation etc.

Process :- is the way in which we produce slw. It is the collection of activities that leads to a product. An efficient process is required to produce good quality products.

Measures, metrics & measurement :-

The term measures, metrics & measurement are used interchangeably.

A measure provide a quantitative indication of the size, dimension, capacity, efficiency, productivity, extent, or reliability of some attributes of a product or process.

Measurement is the act of evaluating a measure.

A metric is a quantitative measure of the degree to which a sys., component or process possesses a given attribute.

19/7/22 - 35, 58 T1

we can define this - (eg. the no. of errors uncovered in the re of a single module)
"when a single data point has been collected, a measur has been established. Measurement occurs as the result of the collection of one or more data points (e.g. a no. of modwle reviews are investigated to collect measures of the no. of errors in each module). A s/w metric relates the individual measures in some way. (e.g. the avg. no. of errors found per review.").

Henie we collect measures & develop metrics to improove the s/w eg engg. practices.

s/w process & Product metrics.

S/w metrics are used to quantitatively characterise different aspects of s/w process or s/w products. Process metrics quantify the attributes of s/w development process & environment; whereas product metrics are measures for the s/w product. e.g. of product metrics includes productivity, quality, failure rate, efficiency etc. E.g. of product metrics are size, reliability, complexity, functionality etc.

S/w process :-

1) Requirement gathering & analysis :- In this phase stake holders communicate with customer & sys. users to gather the business requirements like who will use the sys? How user will interact with the sys? what should be the sys. i/p & o/p? etc. Depending on these requirements, SRS (s/w requirement specification) document is prepared.

2) Design :- This phase is an o/p of requirement gathering phase i.e. requirement specification document as an input. Based on requirement specification s/w design is prepared. O/p of this phase is design specification which specifies h/w & s/w requirements of sys. DFD, flowchart etc. are included by design specification document. Design specification acts as an i/p for implementation phase.

THADOMAL SHAHANI ENGINEERING COLLEGE

P.G. KHER MARG, T.P.S.-III, BANDRA (W), MUMBAI-50.
ON BEHALF OF
UNIVERSITY OF MUMBAI

CODE NO. _____

(To be filled in by the Candidate)
Candidate's Seat No. : (In figures) | | | | | |

(In words) _____

Examination : _____ Subject : _____

Part / Branch / Sem. Group : _____ Total No. of Supplements _____

Q. No.	1	2	3	4	5	6	7	8	9	10	11	12	TOTAL
Marks Obtained													

Junior Supervisor's Name	Junior Supervisor's Signature with date

BEGIN YOUR ANSWER FROM HERE

Signature of the Examiner

3) ~~Implemento~~ Implementation phase :-
This phase divides the sys. work into various smaller parts called modules. These modules are assigned to development team members & actual coding is stored. This is longest phase of sys. development. O/p of this phase is actual code using specific programming language.

4) Testing :- This phase involves testing of actual code of sys. against requirements of user in order to ensure that sys. will satisfy all needs of users. Various testing strategies used are unit testing, system testing, integration testing etc. O/p of testing phase is corrected & modified s/w code.

5) Deployment :- In this phase, the sys. is deployed at users site for their use. In this phase, customer uses the s/w & give feedback to development team for any changes or modifications in sys. if any.

6) maintenance :- Once the s/w is deployed actual problems from user's perspective will come up. It is responsibility of development team to solve user's problems time to time. This process is called maintenance of system. In this phase some additional functionality may need to add in the appl? as per user's requirement.

19/7/22 - 35, 58 T1

A Process Framework :-

A process framework establishes the foundation for a complete s/w process by identifying a small no. of framework activities that are applicable to all software projects, regardless of their size or complexity. The process framework encompasses a set of umbrella activities that are applicable across the entire s/w process.

Fig. shows, each framework activity is popwlated by a set of s/w engineering actions - a collection of related tasks that produces a major s/w engineering work product (e.g., design is a s/w engineering action). Each action is popwlated with individual work task that accomplish some part of the work implied by the action.

The following generic process framework (used as a basis for the description of process models) is applicable to the vast majority of s/w projects.

→ Communication :- This framework activity involves heavy communication & collaboration with the customer (& other stakeholders) & encompasses requirements gathering & other related activities.

→ Planning :- This activity establishes a plan for the s/w engg. work that follows. It describes the technical tasks to be conducted, the risks that are likely, the resources that will be required, the work products to be produced, & a work schedule.

→ Modeling :- This activity encompasses the creation of models that allow the developer & the customer to better understand s/w requirements & the design that will achieve those requirements.

→ Construction :- This activity combines code generation (either manual or automated) & the testing that is required to uncover errors in the code.

→ Deployment :- The s/w (as a complete entity or as a partially completed increment) is delivered to the customer who evaluates the delivered product & provides feedback based on the evaluation.

These five generic framework activities can be used during the development of small programs, the creation of large web applications, & for the engg. of large, complex computer based systems. The details of the s/w process will be quite different in each case, but the framework activities remain the same.

Using an e.g. derived from the generic process framework, the modeling activity is composed of 2 s/w eg engg. actions - analysis & design. Analysis encompasses a set of work tasks (e.g. requirement gathering, elaboration, negotiation, specification, & validation) that leads to the creation of the analysis model (and/or requirements specification). Design encompasses work tasks (data design, architectural design, interface design & component-level design) that create a design model (&/or a design specification).

Referring again to fig. (frame work activity) each s/w engg. action is represented by a no. of different task sets - each a collection of s/w engg. work tasks, related work products, quality assurance points, & project milestones. The task set that best accommodates the needs of the project & the characteristics of the team is chosen. This implies that a s/w engg. action (e.g. design) can be adopted to the specific needs of the s/w project & the characteristics of the project team.

The framework described in the generic view of s/w engg. is complemented by a no. of umbrella activities. Typical activities in this category include :-

- s/w project tracking & control :- Allows the s/w team to assess progress against the project plan & take necessary action to maintain schedule.
- Risk mgmt :- assesses risks that may effect the outcome of the project or the quality of the product.
- s/w quality assurance :- defines & conducts the activities required to ensure s/w quality.
- Formal technical reviews :- assesses s/w engg. work products in an effort to uncover & remove errors before they are propagated to the next action or activity.
- Measurement :- defines & collects process, project & product measures that assist the team in delivering s/w that meets the customer's needs, can be used in conjunction with all other frame-

work & umbrella activities.

- s/w configuration mgmt :- manages the effects of change throu out the s/w process.
- Reusability management :- defines criteria for work product reuse (including s/w components) & establishes mechanism to achieve reusable components.
- work product preparation & production :- encompasses the activities required to create work products such as models documents, logs, forms & lists.

Umbrella activities are applied throughout the s/w process.

All process models can be characterized with in the process framework. Intelligent appln of any s/w process model must recognize that adaptation (to the problem, project, team & organizational culture) is essential for success. But process models do differ fundamentally in :-

- The overall flow of activities & tasks & the interdependencies among activities & tasks.
- The degree to which work tasks are defined within each framework activity.
- The degree to which work products are identified & required.
- The manner which quality assurance activities are applied.
- The manner in which project tracking & control activities are applied.
- The overall degree of detail & rigor with which the process is described.
- The degree to which the customer & other stakeholders are involved with the project.
- The level of autonomy given to the s/w project team.
- The degree to which team organization & roles are prescribed.

Process models that stress detailed definition, identification & appln of process activities & tasks have been applied within the s/w engg. community for the past 30 years. When these prescriptive process models are applied, the intent is to improve sys. quality, to make projects more manageable, to make delivery dates & costs more predictable, & to guide

teams of s/w engineers as they perform the work required to build a sys.

Process models that emphasize project agility & follow a set of principles that lead to a more informal (but, proponents argue, no less effective) approach to s/w process have been proposed in recent years. These agile process models emphasize maneuverability & adaptability. They are appropriate for many types of projects & are particularly useful when web appl^ns are engineered.

→ Prescriptive process model :-

define a distinct set of activities, actions, tasks, milestones & work products that are required to engineer high-quality s/w. These process models are not perfect, but they do provide a useful roadmap for s/w engg. work.

→ who does it ? s/w engineers & their managers adapt a prescriptive process model to their needs & then follow it. In addition, the people who have requested the s/w have a role to play as the process model is followed.

→ what are the steps ?

The process guides a s/w team through a set of framework activities that are organized into a process flow that may be linear, incremental or evolutionary. The terminology & details of each process model differ, but the generic framework activities remain reasonably consistent.

Prescriptive models :-

Every s/w engineering organization should describe a unique set of framework activities for the s/w process it adopts. It should populate each framework activity with a set of s/w engineering actions, & define each action in terms of a task set that identifies the work (& work products) to be accomplished to meet the development goals. It should then adapt the resultant process model to accommodate the specific nature of each project, the people who will do the work, & the environment in which the work will be conducted. Regardless of the process model that is selected, s/w engineers have traditionally chosen a generic process framework that encompasses the following framework activities: communication, planning, modeling, construction & deployment.

19/2/00 - 25 SR T1

why we call prescriptive models because they prescribe set of process elements - framework activities, s/w engg actions, tasks, work products, quality assurance & change control mechanisms for each project. Each process model prescribes a workflow - that is, the manner in which the process elements are interrelated to one another.

All s/w process models can accommodate the generic framework activities, but each applies a different emphasis to these activities & defines a workflow that invokes each framework activity in a different manner.

1) Waterfall Model :-

There are times when the requirements of a problem are reasonably well understood - when work flows from communication through deployment in a reasonably linear fashion. This situation is sometimes encountered when well-defined adaptations or enhancements to an existing sys. must be made (e.g., an adaptation to accounting s/w that has been mandated because of changes to government regulations). It may also occur in a limited no. of new development efforts, but only when requirements are well-defined & reasonably stable.

The waterfall model is also called as classic life cycle, suggests a systematic, sequential approach to s/w development that begins with customer specification of requirements & progresses through planning, modeling, construction & deployment, culminating in on-going support of the completed s/w. The waterfall model is the oldest paradigm for s/w engg.

The problems that are sometimes encountered when the waterfall model is applied are :-

(1) Real projects rarely follow the sequential flow that the model proposes. ~~Althogh~~ Although the linear model can accommodate iteration, it does so indirectly. As a result, changes can cause confusion as the project team proceeds.

(2) It is difficult for the customer to state all requirements explicitly. The waterfall model requires this & has difficulty ~~acc~~ accommodating the natural uncertainty that exists at the beginning of many projects.

③ The customer must have patience. A working version of the program will not be available until late in the project time-span. A major ~~bu~~ blunder, if undetected until the working program is reviewed, can be disastrous.

In an interesting analysis of actual projects, that the linear nature of the waterfall model leads to "blocking states" in which some project team members must wait for other members of the team to complete dependent tasks. In fact, the time spent waiting can exceed the time spent on productive work! The blocking state tends to be more prevalent at the beginning & end of a linear sequential process.

Today, software work is fast paced & subject to a never-ending stream of changes. The waterfall model is inappropriate for such work. However, it can serve as a useful process model in situations where requirements are fixed & work is to proceed to completion in a linear manner.

THADOMAL SHAHANI ENGINEERING COLLEGE

P.G. KHER MARG, T.P.S.-III, BANDRA (W), MUMBAI-50.
ON BEHALF OF
UNIVERSITY OF MUMBAI

CODE NO. _____

(To be filled in by the Candidate)
Candidate's Seat No. : (In figures) | | | | | |

(In words) _____

Examination : _____ Subject : _____

Part / Branch / Sem. Group : _____ Total No. of Supplements _____

Q. No.	1	2	3	4	5	6	7	8	9	10	11	12	TOTAL
Marks Obtained													

Junior Supervisor's Name	Junior Supervisor's Signature with date

BEGIN YOUR ANSWER FROM HERE

Signature of the Examiner

(2) RAD Model (Rapid Appln. Development) :-

→ It is an incremental s/w process model that emphasizes a short development cycle. It is a "high speed" adaptation of the waterfall model, in which rapid development is achieved by using a component based construction approach.

→ If requirements are well understood & project scope is constrained, the RAD process enables a development team to create a "fully functional system" within a very short time period (e.g., 60 to 90 days).

→ Like other process models, the RAD approach maps into the generic framework activities.

→ Communication works to understand the business problem & the info. characteristics that the s/w must accommodate.

→ Planning is essential here because multiple software teams work in parallel on different sys. functions.

→ Modeling has 3 major phases - business modeling, data modeling & process modeling - & establishes design representations that serve as the basis for RAD's construction activity.

→ Construction explains the use of pre-existing s/w components & the appl? of automatic code generation.

→ Finally, deployment establishes a basis for subsequent iterations, if required.
→ Fig. shows, time constraints imposed on a RAD project demand "scalable scope". If a business appl? can be modularized in a way that enables each major function to be completed in less than 3 months, it is a candidate for RAD.
→ Each major f.n can be addressed by a separate RAD team & then integrated to form a whole.
→ Like all process models, the RAD approach has drawbacks –
→ ① For large, but scalable projects, RAD requires sufficient human resources to create the right no. of RAD teams,
→ ② RAD requires developers & customers who are not committed to the rapid-fire activities necessary to complete the sys. in a much abbreviated time frame, RAD projects will fail.
→ ③ If a sys. cannot be properly modularized, building the components necessary for RAD will be problematic.
→ ④ If high performance is an issue, & performance is to be achieved through tuning the interfaces to sys. components, the RAD approach may not work.
→ ⑤ RAD may not be appropriate when technical risks are high (e.g. when a new appl.n makes heavy use of new technology.)

③ Prototyping Model :–
→ It refers to developing s/w appl.n prototypes (early approximation / version) which displays the behaviour of product under development but may not contain the exact logic of the original appl.n
→ Prototyping allows user to evaluate developers proposal & try it before actual implementation.
→ Prototyping model is widely used popular s/w development model as it helps to understand user requirements in early stage of development process.
→ Phases of prototyping models are as follows :–
1) Requirement gathering & analysis.

2) Quick design. 3) Build prototype.
4) User evaluation 5) Refining prototype.
6) Engineer a product.

→ 1) Requirement gathering & analysis :-

Prototype begins with requirement gathering & analysis. In this phase, various users of sys. are interviewed in order to know their sys. requirements or expectations from sys. Requirement specification report is generated as an o/p of this phase.

→ 2) Quick design :-

After gathering & analysis of user requirements quick design (Prototype design) of sys. is developed. Quick design is not detailed design of sys., it contains only necessary characteristics of the sys. which gives basis idea of sys. to the end users.

→ 3) Build prototype :-

Based on feedback received from user about quick design of sys., the 1st prototype of sys. is created. Prototype is working model of sys. under development.

→ 4) User Evaluation :-

Once prototype is built, proposed sys. is presented to end user of sys. for its evaluation. User determines strengths & weaknesses of prototype i.e. what needs to be added or what is to be removed. All the comments & suggestions given by user in feedback are passed to developers.

→ 5) Refining prototype :-

Depending on comments & suggestions came from user, developers refine previous prototype to form new prototype of sys. New prototype is again evaluated. The process of evaluation & refining prototype continues until all user requirements are met by prototype.

→ 6) Engineer a product :-

Once evaluation & refining of prototype completes i.e. when user accepts final prototype of sys., the final sys. is evaluated thoroughly & deployed at user's site. Deployment of engineered product is followed by regular maintenance of sys.

When to use Prototype model :-

It should be used when the desired sys. needs to have a lot of interaction with the end users.

Typically online sys. in which web interfaces have a very high amount of interaction with end users, are best suited for prototype model. It might take a while for a sys to be built that allows ease of use & needs minimal training for the end user.

Prototyping ensures that the end users constantly work with the sys & provide a feedback which is incorporated in the prototype to result in a useable sys. They are excellent for designing good human computer interface systems.

Advantages :-

1) It enables early evaluation of sys. by providiving working model to end users at early stage of development.
2) Refining of prototype results in better implementation of sys. requirements.
3) Communication betⁿ developer & user reduces ambigwity.
4) More involvement of user in development process results in meeting user's requirement at greater extent.

Disadvantages :-

1) Refining of prototype continues until user is completely satisfied, thus it is time consuming & expensive process.
2) More involvement of user in development process is not accepted by developer always.
3) Refining of prototype again & again may disturb the working of development team.
4) Practically prototyping model results in increasing the complexity of sys. as scope of sys. extends beyond original plan.

④ Evolutionary Process model :-

→ It is based on the principle that " stages consist of growing increments of an operational s/w product, with the way of evolution being discovered by operational experience".

→ According to the business need & the changing nature of the market there are lot of improvements required in the s/w product over a time.

→ Due to this lot of improvement is required in the product hence evolutionary development model is iterative in nature.

⑤ Incremental Process Model :-

→ Incremental model applies the waterfall model incrementally. The series of releases is referred to as " increments" with each increment providing more functionality to the customers.

→ After 1st increment, a core product is delivered, which can be used by the customer.

→ Based on customer feedback, a plan is developed for the next increments & modifications are made accordingly.

→ This process continues with increments being delivered until the complete product is delivered. The incremental philosophy is also used in the agile process model.

→ Phases of incremental model :-

Communication :- Helps to understand the objective.

Planning :- Required as many people (s/w teams) work on the same project but different functions at same time.

Modelling :- Involves business modelling, data modelling & process modelling.

Construction :- This involves the reuse of s/w components & automatic code.

Deployment :- Integration of all the increments.

Characteristics of Incremental Model :-

→ sys. is broken down into many mini development projects.

→ Partial systems are built to produce the final system.
→ First tackled highest priority requirements.
→ The requirement of a portion is frozen once the incremented portion is developed.

Advantages of Incremental model :-

→ After each iteration, regression testing should be conducted. During this testing, faulty elements of the s/w can be quickly identified because few changes are made within any single iteration.
→ It is easier to test & debug than other methods of s/w development because smaller changes are made during each iteration. This allows for more targeted & rigorous testing of each element within the overall product.
→ Customer can respond to features & review the product for any needed or useful changes.
→ Initial product delivery is faster & costs less.

Disadvantages of Incremental Model :-

→ Resulting cost may exceed the cost of the organization.
→ As additional functionality is added to the product, problems may arise related to system architecture which were not evident in earlier prototypes.

* Difference betⁿ waterfall Model & Incremental Model.

	Parameter	Waterfall model	Incremental model.
→	Simplicity	Simple	Intermediate
→	Risk Involvement	High	Easily manageable.
→	Flexibility to change	Difficult	Easy.
→	User involvement	Only at beginning	Intermediate
→	Flexibility	Rigid	Less flexible.
→	Maintenance	Least	Promotes maintainability
→	Duration	Long	Very long.

⑥ Spiral Model :-

→ It is a combination of iterative & waterfall model.

→ Spiral model has 4 phases of development each of these phases is called as spiral.

① Identification :-

This phase identifies all business requirements of system at the beginning. It involves clear understanding of requirement by communication betⁿ stakeholders & customer.

In the subsequent iterations all subsystem requirements & unit requirements are identified.

② Design :-

In first iteration, design phase develops conceptual design of system based on initially gathered requirements.

In further spirals or iterations, it develops logical design, physical design, architectural design & final design of sys.

③ Construct :-

Initially construct phase develops a code for conceptual design to get user feedback.

In next subsequent spirals, detailed working model of software is constructed with increment number & are delivered to customer for feedback.

④ Evaluation & risk analysis :-

In this phase mgmt risks like cost overrun are identified & monitored, technical feasibility of sys. is also done. At end of each spiral customer evaluates slw for potential risks in system & provides feedback.

Spiral model can be used for projects where budget & risk evaluation are important factors.

If customers are not sure about their requirements then spiral model is useful than waterfall model.

Advantages :-

→ It is more flexible to changing requirements.

→ Requirements are achieved more accurately.

→ User can see the sys. from 1st iteration to end of development.

→ Risk mgmt is easier.

Friday

123 - Evolutionary.
129 - Incremental.
77 - Spiral
83 - Icarbon model.
126 - V model
106 - Concurrent model.

Disadvantages :-
→ It is difficult to mange development process.
→ Not useful for small projects developme
→ spri spiral can run indefinitely.
→ It requires excessive documentation work as documentation is prepared for each iteration.

(7) Concurrent Models :-

→ The concurrent development model is also known as concurrent engg.
→ This model helps s/w team in the representation of iterative as well as concurrent elements of various proces models.
→ e.g., consider the modeling activity which has been defined for the spiral model is carried out by calling one or more s/w engg. actions like prototyping, analysis & design.
→ From fig. we can observe schematic representation regarding a s/w engg. activity in the scope of the modeling activity with the help of concurrent modeling approach.
→ The state of activity modeling may be any out of the states which are noted at any particular time.
→ In the same way, it is possible to represent remaining activities, actions or tasks such as communication, construction etc. in an analogous manner.
→ All the activities regarding s/w engg. are executed simultaneously but exist in different states.
→ e.g., consider in the early phase of a project, first iteration of the communication activity (not displayed in ~~da~~ diagram) has been completed & exists in the awaiting changes state.
→ The modeling activity whose state is inactive at the time of completion of initial communication, now changes its states to under development.
→ If customer enforces that changes in requirements

THADOMAL SHAHANI ENGINEERING COLLEGE

P.G. KHER MARG, T.P.S.-III, BANDRA (W), MUMBAI-50.

ON BEHALF OF

UNIVERSITY OF MUMBAI

CODE NO. _____

(To be filled in by the Candidate)

Candidate's Seat No. : (In figures) | | | | | |

(In words) _____

Examination : _____ Subject : _____

Part / Branch / Sem. Group : _____ Total No. of Supplements _____

Q. No.	1	2	3	4	5	6	7	8	9	10	11	12	TOTAL
Marks Obtained													

Junior Supervisor's Name	Junior Supervisor's Signature with date

BEGIN YOUR ANSWER FROM HERE

Signature of the Examiner

should be necessarily done, the modeling activity shifts from the under development state into the awaiting changes state.

→ Concurrent modeling describes a sequence of events which will activate transitions in betⁿ states for all the s/w engg. activities, actions or tasks.

→ e.g., in previous phase of design, an inconsistency in the requirements model is uncovered.

→ Because of it the event analysis model correction, is generated that will activate the requirements analysis action from the done state into the awaiting changes state.

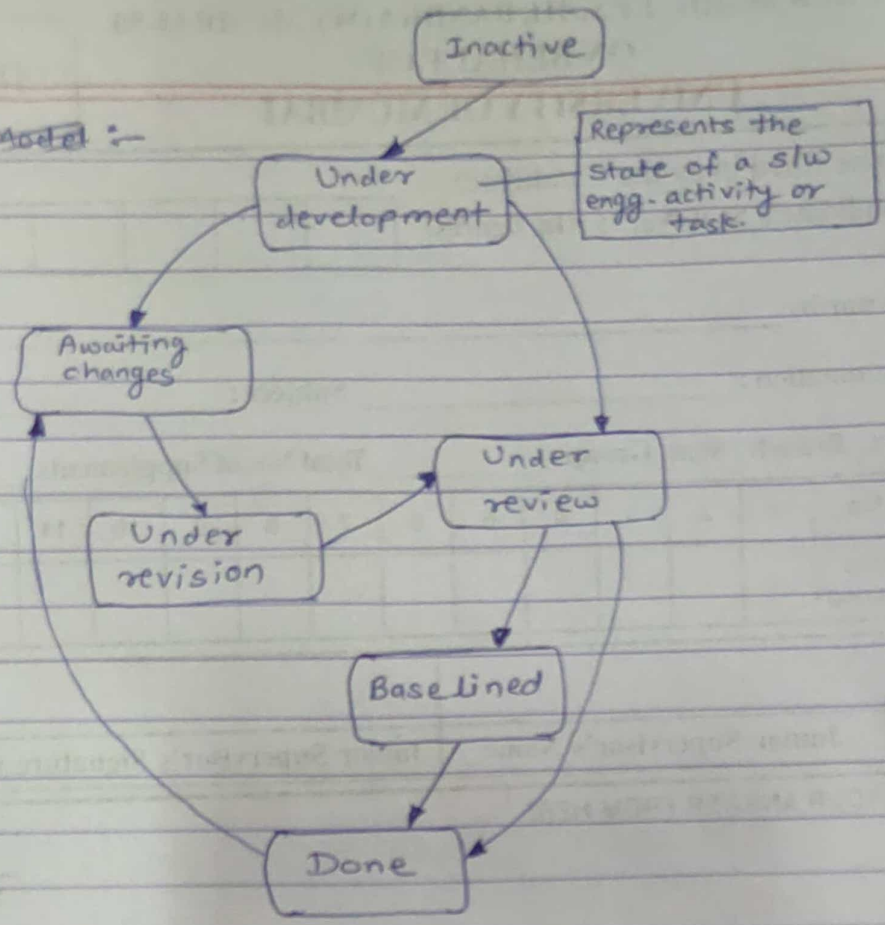
→ Concurrent modeling is ~~prov~~ proved to be compatible for all the types of s/w development & offers an exact picture of the ongoing state of a project.

→ Instead of encircling all the s/w engg. actions, & tasks to a series of events, a process n/w is defined by the concurrent modeling.

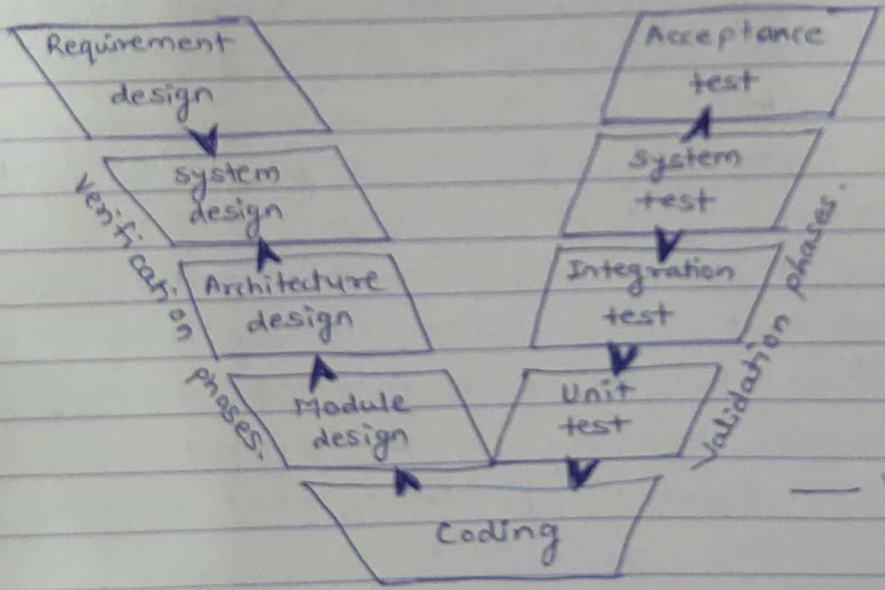
→ All the activities, actions or tasks on the n/w execute concurrently with other activities, actions or tasks.

→ In the process n/w, events which are generated at one point activate transitions betⁿ the states.

(7) V Model :-



(8) V - Model :-



→ In s/w development, V model represents a development process that may be considered an extension of the waterfall model.

→ Here, instead of moving down in a linear way, the process steps are bent upwards after the coding phase to form the typical V shape.

→ The V-Model explains the relationship bet^n each phase of the development life cycle & its associated phase of testing.

→ The horizontal & vertical axes represent time or project completeness (left-to-right) & level of abstraction respectively. This model is divided into two phases: verification & validation.

→ Verification phases :-

1) Requirement analysis :- The 1st step in the verification process is requirement analysis. The requirements of the sys. are collected by analyzing the needs of the user(s).

→ This phase is concerned with establishing what the ideal sys. has to perform. It doesnot determine how the s/w will be designed or built.

→ Usually, the users are interviewed & a document called the user requirements document is generated.

→ The user requirements document will describe the system's functional, interface, performance, data, etc. requirements as expected by the user.

→ It is used by business analysts to communicate their understanding of the sys. to the users.

→ The users carefully review this document as this document would serve as the guideline for the system designers in the system design phase. The user acceptance tests are designed in this phase.

→ There are different methods for gathering requirements of both soft & hard methodologies including; interviews, questionnairs, document analysis, observation, throw-away prototypes, use case & static & dynamic views with users.

2) System design :- In this phase, sys. engineers analyze & understand the business of the proposed sys. by studying the user requirements document.

→ They figure out posibilities & techniques by which the user requirements can be implemented.

→ If any of the requirements are not feasible, the user is informed regarding the issue. A resolution is found & the user requirement document is edited accordingly.

→ The s/w specification document which serves as a blueprint for the development phase is generated.

→ This document contains the general sys. organization, menu structures, data structures etc. It may hold business scenarios, sample windows, reports for the better understanding.

→ Other technical documentation like entity diagrams, data dictionary will be produced in this phase. The documents for sys. testing are prepared here.

3) Architecture Design :- The design of comp. architecture & s/w architecture can be referred to as high-level design.

→ The baseline in selecting the architecture is that it should ~~not~~ realize all which typically consists of the list of modules, brief functionality of each module, their interface relationships, dependencies, database tables, architecture diagrams, technology details etc.

→ The integration testing design is carried out in the particular phase.

4) Module design :- This phase can be referred to as low-level design. The designed sys. is divided into smaller units or modules & each of them is explained so that the programmer should be able to start coding directly.

→ The low level design document or program specifications will contain a detailed functional logic of the module, in pseudo-code :-

* Database tables, with all elements, including their types & sizes.
* All interface details with complete API (Appl.ⁿ Programming Interface) references.
* All dependency issues.
* Error message listings.
* complete input & outputs for a module.

→ The unit test design is developed in this stage.

→ Validation Phases :- Following are the phases of validation in V-Model.

1) Unit Testing :- In V-Model, Unit Test Plans (UTPs) are developed during module design phase.
→ These UTPs are executed to eliminate bugs at code level or unit level.
→ A unit is the smallest entity which can independently exist, e.g. a program module.
→ Unit testing verifies that the smallest entity can function correctly when isolated from the rest of the codes/units.

2) Integration Testing :- Integration Test Plans are developed during the Architectural Design Phase.
→ These tests verify that units created & tested independently can coexist & communicate among themselves. Test results are shared with customer's team.

3) System Testing :- These Tests Plans are developed during System Design Phase.
→ System Test Plans are composed by client's business team.
→ System Test ensures that expectations from application developed are met. The whole appl? is tested for its functionality, inter-dependency & communication.
→ system testing verifies that functional & non-functional requirements have been met.
→ Load & performance testing, stress testing, regression testing, etc. are subsets of system testing.

4) User acceptance Testing :- User Acceptance Test (UAT) Plans are developed during the requirement analysis phase.
→ Test Plans are ~~comb~~ composed by business users. UAT is performed in a user environment that resembles the production environment, using realistic data.
→ UAT verifies that delivered system meets user's requirement & system is ready for use in real time.

Advantages :-

→ This model is simple & easy to use.
→ Testing activities like planning, test designing are occured prior to coding. This avoids wastage of time.
→ Proactive defect tracking - means diagnosis of defects is done at early stage.
→ Avoids the downward flow of the defects.
→ Good for small projects in which requirements are easily understood.

Disadvantages :-

→ This model is very rigid & least flexible.
→ The development of s/w is done in the implementation phase hence no early prototypes regarding the s/w are produced.
→ If there are changes in midway, then there is need to update the test documents along with requirement documents.

Capability Maturity Model (CMM).

→ The CMM invented by Software Engineering Institute (SEI) states a growing sequence of levels of a s/w development organization.
→ CMM is used to create & refine s/w creation procedure of organization.
→ The higher the level, the better the s/w development procedure so reaching every level is a cost consuming & time consuming procedure.
→ CMM is same as ISO 9001, one of the ISO 9000 sequences of standards determined by the International organization for standardization (ISO).
→ ISO 9000 standard state an efficient quality sys. for manufacturing & service industries; ISO 9001 applied particularly to s/w development & maintenance.

Following are levels of CMM :-

1. Level 1 : Initial.
   → This s/w procedure is stated as inconsistent & occasionally chaotic.
   → Stated procedures & standard ~~prac~~ practices that present are discarded at the time of crisis.
   → Success of the association depends on efforts taken by each person, capacity & heroics.
   → The heroes move on to another association taking their knowledge or lessons ~~lear~~ learnt with them.

2. Level 2 : Repeatable.
   → This level has a fundamental & consistent project management processes to keep the track of cost, schedule & functionality.
   → The procedure is in region to repeat the latest successes on projects with similar appl.ns
   → Prog. mgmt is a most imp. feature of a level two association.

3. Level 3 : Defined.
   → S/w procedure for mgmt & engg. actions are documented, standardized & incorporated into a standard s/w procedure for the overall association & each & every project developed by association use an approved, tailored version of the standard s/w procedure of association for creating, testing & maintaining the s/w product.

4. Level 4 : Managed.
   → Management can successfully control the efforts required for s/w development through precise measurements.
   → At this level, association set a quantitative quality aim for both s/w procedure & s/w maintenance.
   → At this level, the performance of procedures is controlled through statistical & other quantitative mechanisms, & is quantitatively predictable.

5. Level 5: Optimizing.

→ Characteristic of this level is concentrating on repeated improving procedure performance with the help of both increasing & innovative technological corrections.

→ At this level, modifications to the procedures are done to increase the procedure performance & at the same time maintaining ~~stas~~ statistical probability to get the recognized quantitative procedure-improvement aims.

CMMI (Capability Maturity Model Integration).

→ The CMM model's appl^n in s/w development has been sometimes problematic. Applying multiple models that are not integrated within & across an organization could be costly in training, appraisals & improvement activities.

→ The CMMI project was formed to sort out the problem of using multiple models for s/w development processes, thus the CMMI model has superseded the CMM model, though the CMM model continues to be a general theoretical process capability model used in the public domain.

→ CMMI framework consists of a collection of computer programs based on knowledge, sys. engg., s/w engg., integrated product & process development & provider sourcing.

→ CMMI framework has 3 groups as:-

1. CMMI for Development (CMMI-DEV)
2. CMMI for service (CMMI-SVC)
3. CMMI for Acquisition (CMMI-ACQ)

→ These 3 groups forms model components which are uniquely designed for particular business process & they may contain some core processes as a part of them which will be same among these groups.

→ It is also possible to extend CMMI framework by making addition of model component to it.

THADOMAL SHAHANI ENGINEERING COLLEGE

P.G. KHER MARG, T.P.S.-III, BANDRA (W), MUMBAI-50.
ON BEHALF OF
UNIVERSITY OF MUMBAI

CODE NO. _____

(To be filled in by the Candidate)

Candidate's Seat No. : (In figures) | | | | | |

(In words) _____

Examination : _____ Subject : _____

Part / Branch / Sem. Group : _____ Total No. of Supplements _____

Q. No.	1	2	3	4	5	6	7	8	9	10	11	12	TOTAL
Marks Obtained													

Junior Supervisor's Name	Junior Supervisor's Signature with date

BEGIN YOUR ANSWER FROM HERE

Signature of the Examiner

Agile Process :-

→ Agile s/w development describes an approach to s/w development under which requirements & solutions evolve through the collaborative effort of self-organizing & cross-functional teams & their customers (end users).

→ It advocates adaptive planning, evolutionary development, early delivery, & continual improvement, & it encourages rapid & flexible response to change.

→ The term Agile was popularized, in this context, by the Manifesto for Agile s/w development.

→ The values & principles adopted in this manifesto were derived from & underpin a broad range of s/w development frameworks, including Scrum & Kanban.

→ There is significant subjective evidence that adopting agile practices & values improves the agility of s/w professionals, teams & organizations.

Agile s/w development values :-

→ Based on their combined experience of developing s/w & helping others do that, the seventeen signatories to the manifesto proclaimed that they value:

* Individuals & Interactions over processes & tools.
* Working s/w over comprehensive documentation.

★ Customer collaboration over contract negotiation
★ Responding to change over following a plan.

→ As per the view of Scott Ambler (canadian s/w engg engineer, consultant & author):
→ Tools & processes are important, but it is more imp to have competent people working together effectively.
→ Good documentation is useful in helping people to understand how the s/w is built & how to use it, but the main point of development is to create s/w, not documentation.
→ A contract is imp. but is no substitute for working closely with customers to discover what they need.
→ A project plan is imp., but it must not be too rigid to accommodate changes in technology or the environment, stakeholder's priorities, & people's understanding of the problem & its solution.

Agility Principles:-

→ Manifesto for Agile s/w Development is based on 12 principles.

1) Customer satisfaction by early & continuous delivery of valuable s/w.
2) Welcome changing requirements, even in late development
3) Working s/w is delivered frequently (weeks rather than months).
4) Close, daily cooperation bet^n business people & develop
5) Projects are built around motivated individuals, who should be trusted.
6) Face to face conversation is the best form of communication (co-location).
7) Working s/w is the primary measure of progress.
8) Sustainable development, able to maintain a constant pace.
9) Continuous attention to technical excellence & good desig
10) Simplicity is essential.
11) Best architectures, requirements & designs emerge from self-organizing teams.

12) Regularly, the team reflects on how to become more effective, & adjusts accordingly.

Advantages of Agile process :—
→ Stakeholder engagement.
→ Early & predictable delivery.
→ Allows for change.
→ Focuses on users.
→ Transparency.
→ Predictable costs & schedule.
→ Focuses on Business value.
→ Improves Quality.

Differece bet^{n} Prescriptive Process Model & Agile Process Model.

Parameter	Prescriptive	Agile.
Basic Aim	Developed to bring order to structure to the s/w development process.	These models satisfy customer through early & continuous delivery.
Functionality.	It can accommodate changing requirements.	Defines a distinct set of activities, actions, tasks, milestones & work products that are required to engineer high-quality s/w.
Popularity	More Popular.	Less popular.
e.g.	water fall model, Incremental model.	scrum, extreme programming (XP), Dynamic Sys. development method (DSDM) Adaptive s/w development (ASD)

Extreme Programming (XP) :-

→ XP is a s/w development methodology which is intended to improve s/w quality & responsiveness to changing customer requirements.

→ As a type of agile s/w development, it advocates frequent "releases" in short development cycles, which is intended to improve productivity & introduce checkpoints at which new customer requirements can be adopted.

→ other elements of extreme programming include: Programming in pairs or doing extensive code review, unit testing of all code, avoiding programming of features until they are actually needed, a flat mgmt structure, code simplicity & clarity, expecting changes in the customer's requirements as time passes & the problem is better understood, & frequent communication with the customer & programmers.

→ The methodology has the idea that the beneficial elements of traditional s/w engg. practices are taken to "extreme" levels.

→ e.g., code reviews are considered a beneficial practice, taken to the extreme, code can be reviewed continuously, i.e. the practice of pair programming.

XP Values :-

→ Extreme programming initially has 4 values in 1999:- "Communication, simplicity, feedback & courage."

→ A new value, "respect" was added in second edition of extreme programming.

→ Five values are described as:-

1. Communication:-

→ Building s/w systems requires communicating system requirements to the developers of the system. In formal s/w development methodologies, this task is accomplished through documentation.

→ Extreme programming techniques can be viewed as methods for rapidly building & dise disseminating institutional

knowledge among members of a development team.

→ The goal is to give all developers a shared view of the sys. which matches the view held by the users of the system.

→ To this end, extreme programming favors simple designs, common metaphors, collaboration of users & programmers, frequent verbal communication & feedback.

2. Simplicity :-

→ Extreme programming encourages starting with the simplest solution. Extra functionality can be added later.

→ Difference betⁿ this approach & more conventional sys. development methods is the focus on designing & coding for the needs of today instead of those of tomorrow, next week or next month.

→ This is summed up as "You aren't gonna need it" [YAGNI] approach.

→ Coding & designing for uncertain future requirements implies the risk of spending resources on something that might not be needed, while perhaps delaying crucial features.

→ Related to the "communication" value, simplicity in design & coding should improve the quality of communication. A simple design with very simple code could be easily understood by most programmers in the team.

3) Feedback :-