CH#2:

DATE

Software development life cycle:

↳ SDLC is a systematic methodology/process for building the software product that ensures the quality and correctness of the build product.

↳ SDLC contains multiple process to produce high quality s/w that meets the customer expectations.

↳ Two stages of life cycle are:
- i) Engineering stage
- ii) Production stage

→ This is driven by less predictable smaller team doing design and synthesis activities.

→ This is driven by more predictable but larger team doing construction, test and deployment activities.

Life cycle Aspects	Engineering stage Emphasis	Production Stage Emphasis
> Risk reduction	Schedule, Technical feasibility	Cost
> Products	Architectural baseline	Product release baseline
> Activities	Analysis, planning, design	Implementation, testing
> Assesment	Demo, inspection and analysis	Testing
> Economics	Resolving diseconomy of scale	Exploiting diseconomy of scale
> Management	planning	Operations.

PAGE

DATE

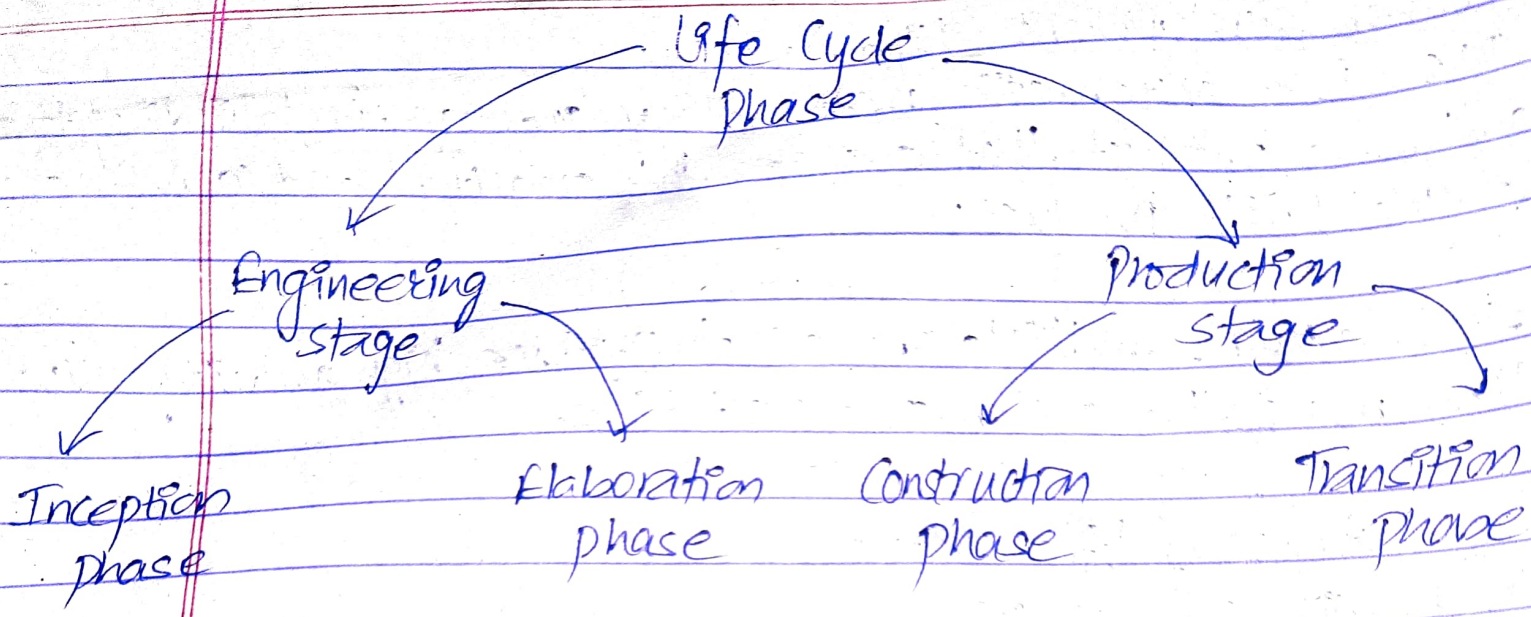
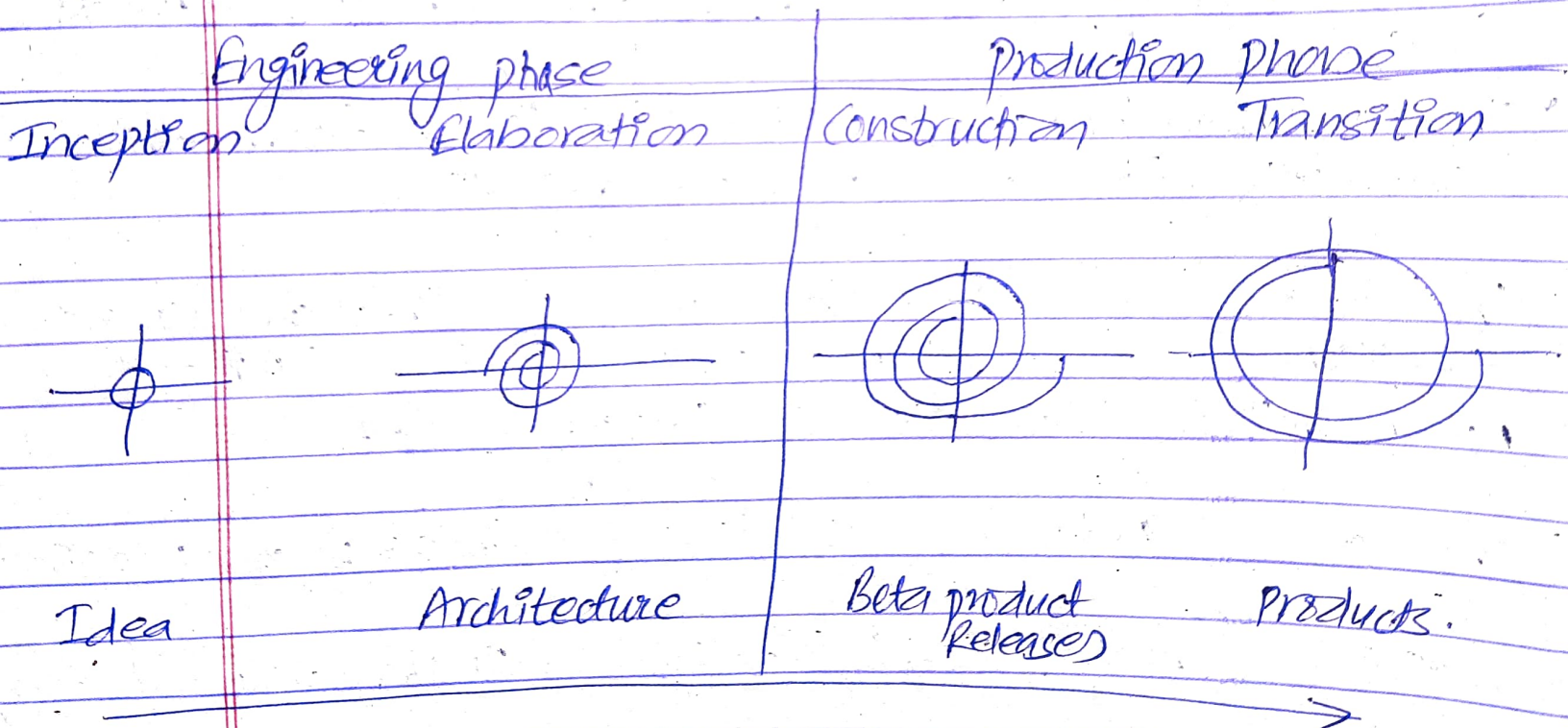


fig: Phases of life cycle ~~phases~~ process.



DATE

(1) Inception phase:

↳ This phase involves establishing goals and gathering the requirements need for the software development.

↳ Here, the overall objective of the life cycle of the project is determined.

↳ Project objective:
- Establish project scope and boundary condition,
- Establish the cost and schedule for the entire project.
- Establish potential risks,
- Establish the architectural design for the project.

↳ Essential Activities:

→ Formulating the scope of the project:
- Here, the user requirements and operational concept of the are captured and stored in the information repository.
- The information repository should be sufficient for the defining the project problem space and also determine the acceptance criteria.

→ Creating the architecture:
- Design trade off (losing one quality or aspect in return to other) problem space ambiguities and available tools, technology and components are analyzed.
- Depending on these information the decision for making/buying of component is made which helps in cost, schedule & effort estimation.

PAGE

DATE

→ Planning and preparing business case:
- Alternatives for risk management, staffing, cost, schedule & profitability are evaluated.

- The infrastruce sufficient for supporting the life cycle is determined.

↳ Evaluating Criteria:
→ Do all stakeholder agree on the scope defination and cost/schedule estimations?

→ Are requirements understood?
→ Are cost, schedule estimates, priorities, tacks and development process credible?
→ Do the depth and breadth of the architecture prototype do demonstrate the architecture baseline?

→ Are actual resource expenditures and planned expenditures ~~are~~ acceptable?

(II) Elaboration phase:
↳ This phase involves the indepth evaluation ~~of the~~ & study of the strong architecture & infrastructure.

↳ Also establish the strong architecture & infrastructure.

↳ Here, we analyze use case and other software diagrams and also work in improving the efficiency of the architecture.

PAGE

DATE

↳ Primary objective:

→ Baselining the architecture where a configurable managable snapshot of architecture is established where all changes are rationalized, tracked & maintained.

→ Baselining the vision.

→ Baselining the high fidelity plan for const^c phase,

↳ Essential Activities:

→ Elaborating the vision
> Establishing the high fidelity ~~vist~~ understanding of the critical use cases.

→ Elaborating the process and infrastructure:
> Establishing the automation support process, tools and construction process

→ Elaborating the architecture & selecting components
> lesson learnt from these activities may result in redesign of the architecture.

↳ Evaluating Criteria:

→ Is the vision stable,

→ Is the architecture stable,

→ Has the major risk elements addressed and credibly resolved?

→ Does all the stakeholder agree on the design and the visions?

→ Are actual response expenditure versus planned expenditures acceptable?

PAGE

DATE

[III] Construction Phase:

↳ In this phase, we perform the implementation of our software as per the planning and requirement specified.

↳ Here, we minimize the risk and eliminate it.

↳ In this phase, the management mindset undergoes a transition from the development of intellectual property during inception and elaboration to the development of deployable products.

↳ Primary Objective:

→ Minimizing the development cost by optimizing resources and avoiding unnecessary scrap & resource reworks.

→ Achieving adequate quality as rapidly as possible.

→ Achieving useful versions (alpha, beta and other test releases).

↳ Essential Activities:

→ Resource management and control,
→ Process optimization
→ Complete component development and testing against evaluation criteria.
→ Assessment of developed product against the acceptance criteria.

PAGE

DATE

↳ Evaluation Criteria:

→ Is this product baseline mature enough to be deployed in the user community?
→ Is this product baseline stable enought be deployed in the user community?
→ Are the stakeholder ready for transition to the user community?
→ Are actual resource expenditure verses planned expenditure resource?

[IV] Transition Phase:

↳ In this phase, we perform strict testing mainly termed as beta testing and deployment of the software to the user's community.

↳ Here, the developer performs their job from user point of view and improves the product as per their feedbacks

↳ Primary objectives:

→ Achieving user self-supportability,
→ Assures that the developed product is stable and performance is maintained.
→ Make sure that the user feedbacks are continuously recorded and changes are made to improve the end product.
→ Achieve the final product baseline as rapidly and cost efficiently as practical.

PAGE

DATE

↳ Essential Activities:

→ Synchronization and integration of concurrent construction increments into consistent deployed baseline.

→ Deployment specific engineering.

→ Assesment of the deployed product baseline against the complete vision and the acceptance criteria made.

↳ Evaluation Criteria:

→ Are the customers satified?

→ Are actual expenditure resources verses planned expenditure resources?

DATE

Various Elements of the software process

Artifacts of the process:

↳ To make the development process more managable and configurable different informations about the project is collected and organized into an element called Artifacts sets.

↳ Each set consists of related artifacts and represented on a uniform format (i.e. English text, c++, visual basic, java or UML model).

↳ The 3 main artifact sets are:
1) Management sets,
2) Engineering sets,
3) Pragmatic sets

1) Management sets:
→ The management set captures the artifacts associated with process planning and execution.

→ These artifacts ~~are~~ use ad hoc notations including text, graphics or any other representations to capture the contracts among various personnel involved in the project (like management, architects, developers, testers etc).

→ These artifacts are evaluated, assessed and measured through the following:
> Relevant stakeholder review,
> Analysis of change between current version of artifacts and previous versions.
> Major milestones demo of balance among all artifacts.

PAGE

DATE

> Evaluate the accuracy of business case and vision artifacts.

↳ The management sets artifacts includes:

(i) Work Breakdown structure:

→ This is an activity breakdown and financial tracking mechanism.

→ If the WBS is structured improperly, the project design might evolve into wrong direction.

(ii) Business Case:

→ This artifact provides all the information necessary to determine if the project is worth investing (or not).

→ It consists of all the financial informations, like expected revenue, expected cost, technical and management plans.

(iii) Release Specification:

→ This artifacts provides information about scope, plan and objective of our project.

→ It also consists the starting time of our project, duration for the completion of the project etc.

→ We make use of various sources like make/buy, analyses, architectural concern, budget, resources etc.

planning artifacts

PAGE

Planning artifacts.

DATE

(IV) Software development plan:

→ This elaborates the process framework into a detailed plan.

→ It must be according to the contract (if any) and also follow Organization standard.

(V) Release descriptions:

→ This document describes the result of each release, including performance against each of the evaluating criteria in the corresponding release specification.

(VI) Status Assesment:

→ This provides periodic snapshot of the project health and status including the risks assesments, quality indicator and management indicators.

→ Its main objective is to ensure that all expectations of all parties are synchronized and consistent.

(VII) Environment:

→ This docs should include requirements management, visual modelling, document automation, host and target programming tools, automated regression testing and continuous and integrated change management and feature & defect tracking.

(VIII) Deployment:

→ This documents can take many forms and depending on the project it could include several documents subsets for transition from production to operational stage.

Minimal artifacts.

PAGE

→ This artifacts includes, user manuals, software installation manuals, site survay, etc.

(IX) Software change order database:

→ In modern development process, we use iterative development process.
In this case, we can face multiple changes that are made to the software and to keep record of these changes, we make use of change order database.

→ It includes, the changes made, time when the change was made, what caused the change, etc.

Operational artifacts.

(2) The Engineering sets:

↳ This is an artifact sets containing implementation set, requirement set, Design set and deployment set.

↳ Here, the mechanism for evaluating the quality of the product / each artifact set is the transition from set to set.

↳ It includes:

① Requirement sets:

→ Here, structured text is used for the vision statement which documents the project scope that supports the agreement bet: the customer and the project team. (SRS).

DATE

→ UML notations are used for engineering representation of requirements models.

→ This set is the primary engineering context for evaluating the other 3 engineering sets (artifacts) and is the basis for test cases.

→ It includes : > Vision document,
> Requirements model

(ii) Design sets:
→ The UML notations are used to design the model for the solution and the design set contains varying levels of abstraction that represents the components of the solution space.

→ It includes : > Design model,
> Test model
> S/w architecture description.

(iii) Implementation sets:
→ This set includes the source code that represents the tangible implementation of components.

→ It also includes the necessary build version of the product for testing purpose.

→ It can be translated/compiled into deployment sets and includes : > Source code baseline,
> Associated compile time files,
> Components executables.

PAGE

DATE

(iv) Deployment sets:

→ This set of artifacts contains user deliverables and machine language notations, executable software and the build scripts, installation scripts, etc.

→ It includes: > Integrated product executable baselines
> Associated runtime files
> User manuals.

(3) Pragmatic Artifacts:

↳ It is the conventional document driven approach that generally wastes large amount of engineering time on developing, polishing, formatting, reviewing, updating, modifying and distributing documents.

↳ It is an approach the redirects this effort of documentation to simply improve rigor and easily understanding of source of data and information.

↳ This idea raises various cultural issues:

(i) People simply want to review data & informations but sometimes donot understand language of artifacts.

(ii) People who wants to review the data & information have no access to the tools.

(iii) Human readable engineering artifacts must take use of rigorous notations to fully that are complete & consistent.

PAGE

DATE

[1] Software Architecture perspective from Management perspective:

↳ Management team see/view a software architecture from a different viewport.

↳ They generally donot directly get involved in the production but makes sure that there is proper communication and intesaction maintained between the members.

↳ Also makes sure that the s/w architecture is precise and accurate.

↳ From this view, 3 different aspects of architecture are:

(i) An intangible (which cannot be touched) architecture of the system is designed.
- This includes all the necessary engineering to specify the complete bill of materials.
- Here, > make/buy decisions are resolved,
  > custom components are elaborated,
  > Individual component cost and assembly cost of the components are determined.

(ii) An architectural baseline is a slice of info across the engineering artifacts set that satisfies all the stakeholder by preparing a business case.

(iii) An architectural description (human readable representation) which ~~gives~~ includes the necessary information to get clear vision about the models.

PAGE

DATE

(II) Architecture from Technical prespective:

↳ Technical prespective generally explains the system's technical level of view.

↳ Here, the architecture framework is defined in terms of views that are abstracted abstraction of UML models in the design set.

↳ The design model includes the full depth and breadth of informations.

↳ This consists of 5 views:

(i) Design view:

→ It describes the architecturally significant elements of the design model.

→ The view is addresses the base structure & functionality of the solution.

→ It is modelled statically using class & object diagram.

→ It is modelled dynamically using behavioral diagram.

(ii) Use case view:

→ This describes how system are critically realized by elements of design model.

PAGE

→ It is modelled statically using Use case diagram and dynamically using behavioral diagram.

(iii) Process View:
→ It addresses the runtime collaboration issuess involved in executing the architecture on the distributed deployment model.

→ It is statically modelled using deployment diagram and dynamically modelled using behavioral diagram.

(iv) Component View:
→ Describes the architectural significant elements of the implement set.

→ It is modelled statically using component diagram and dynamically using behavioral diagram.

(v) Deployment View:
→ It addresses executable realization of the system.

→ It is statically modelled by deployment diagram & dynamically modelled by behavioral diagram.

DATE

Software process workflow:

↳ Workflow refers to the series of tasks that are performed to achieve the certain goal of a project.

↳ Every workflow contains, input, transformation & outputs.

↳ Software process is a set of related activities performed to achieve the quality s/w as an end product and the workflows that leads the s/w development in linear way by performing sequential tasks are called software process workflows.

↳ There are 7 software process workflow in SPM:

(i) Management Workflow
(ii) Environment Workflow
(iii) Requirement Workflow
(iv) Design Workflow
(v) Implementation Workflow
(vi) Assesment Workflow
(vii) Deployment Workflow

(i) Management Workflow:

→ In this workflow, some of the project controlling processes are carried out.

→ Here, artifacts like SRS, business case and vision docs are prepared and the stakeholders are ensured win win condition in terms of developing, executing and implementing s/w project.

PAGE

DATE

(ii) Environment Workflow:

→ Automating the process to coordinate and integrate tools and people with process through workflow.

→ It makes sure that the human errors are reduced and faster development with faster resource allocation is ensured.

→ Also makes sure that the issues are responded and solved.

(iii) Requirement Workflow:

→ Analyzing the problem domain for identifying and understanding the problems and finding solutions to the problem.

→ Evolving the requirement artifacts such as use case, requirements and design documentation/specification which helps in describing the function & design of the software.

(iv) Design Workflow:

→ This workflow includes the modelling of the software design such that it address the entire software design.

→ Evolving the architecture & design artifacts.

DATE

(V) Implementation Workflow:

→ Here, the actual design are programatically implemented as per the design documentation.

→ Also, the deployment documentation are evolved.

(VI) Assesment Workflow:

→ In this workflow, the build product are tested under multiple conditions to assess/ensure the quality of the product is maintained.

→ It also test whether the product is build as per the SRS docs (or not).

(VII) Deployment Workflow:

→ In this WF, the process of delivering the product to the end user is carried out.

→ The product is also deployed to various environments for acceptance.

DATE

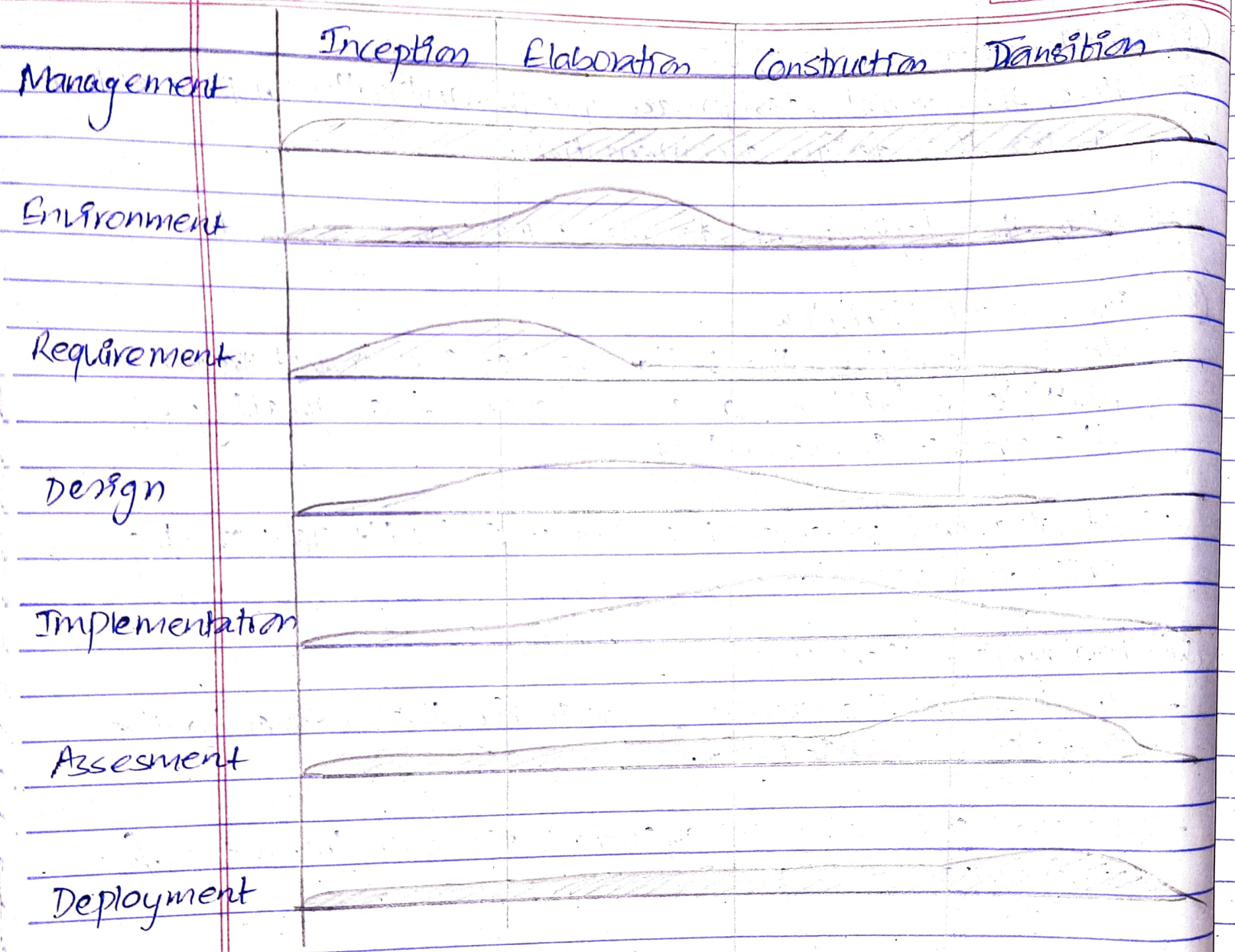


Fig: Activity level across the life cycle phase.

PAGE

DATE

Iterative Workflow:

↳ An Iterative Workflow consists of a set of activities in various proportions depending on where the iteration is located in the development cycle.

↳ The components that are needed in order to implement the scenerio are developed and integrated with the result of previous iteration.

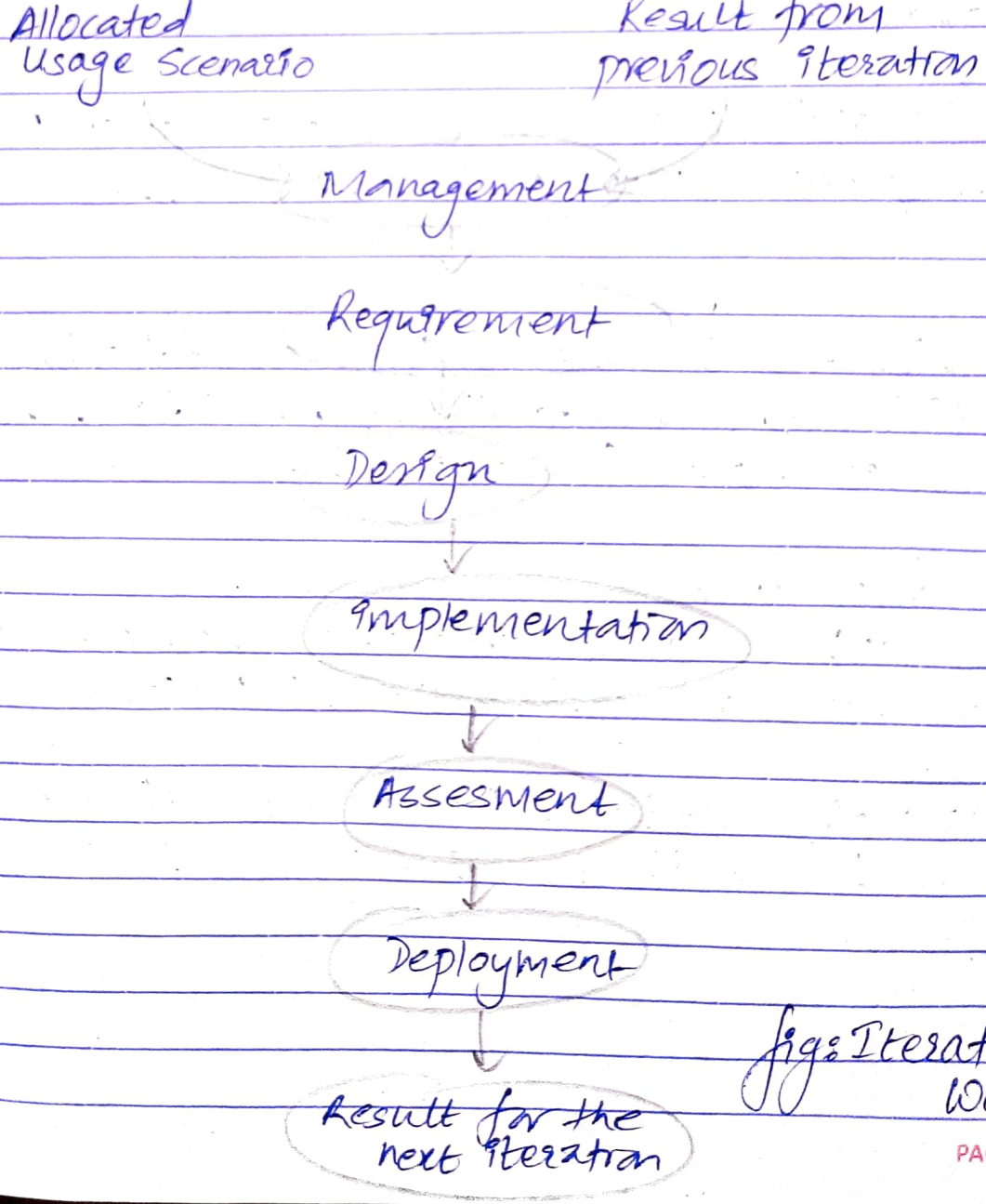


fig: Iterative Workflow

(i) Management:
→ Plan for the iteration to determine the content of the release and develop detail phan for the iteration.

→ Assign the task to the development teams.

(ii) Requirements:
→ Analyze the baseline architecture, baseline requirement and problem and determine the corresponding solutions.

→ Elaborate the use case to be demonstrated at the end of the iteration and evalution criteria.

(iii) Design:
→ Evolving the baseline architecture and the baseline design artifacts to elaborate the ~~fully~~ full design model and test model for the system.

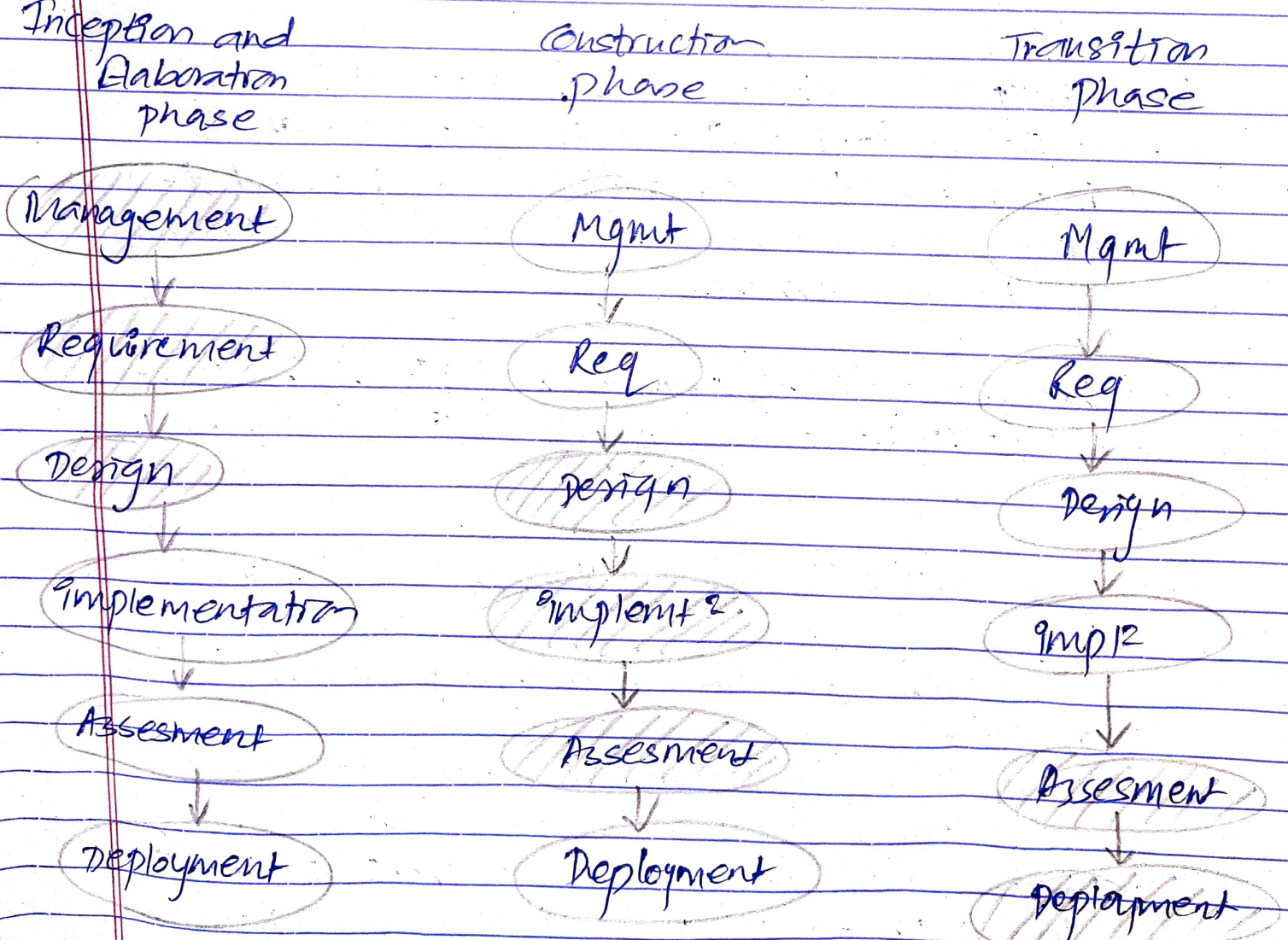
(iv) Implementation:
→ Developing or acquiring the new components and enhancing or modifying the existing components.

→ Integrating and testing the new or modified components with the existing baseline.

DATE

(v) Assesment:
→ Evaluating the result of the iteration under the various conditions.
→ Identifying if any rework and fixes need to be done before the release of the iteration build version.

(vi) Deployment:
→ Now, deploying (or releasing) the build version to the user domain or deploy it to the next iteration for further works.



PAGE

DATE

Status Monitoring:

↳ Status Monitoring is one of the most important task that is need to be performed while software development to make sure the progress is on right track.

↳ Continuous monitoring allows the project manager to identify the health of the project, identify necessary changes, avoid reworks and unnecessary activities etc.

↳ Checkpoints of the process & milestones:

→ While developing the software product it is important to held an event at the end of the every development phase to discuss the progress of the development.

→ These checkpoints provide visibility to milestones in life cycle and also system issues and problems.

→ These checkpoints generally provide following things:

ⓐ It synchronizes the management and engineering prespective.

ⓑ It verifies whether the goal and objective in every phase has been achieved (or not).

ⓒ It provides analysis and evaluation to determine if the project is proceeding as per the plan (or not)

PAGE

DATE

④ It also identifies risks, issues or problems that are essential and condition that are not tolerable.

⑤ Perform global assesment for whole life cycle, not just the current individual perspective or intermediate product.

↳ To synchronize the expectations of stakeholders through life cycle, the following three sequences are used:-

Inception	Elaboration		Construction			Transition
Itr° 1	Itr° 2	Itr° 3	Itr° 4	Itr° 5	Itr° 6	Itr° 7

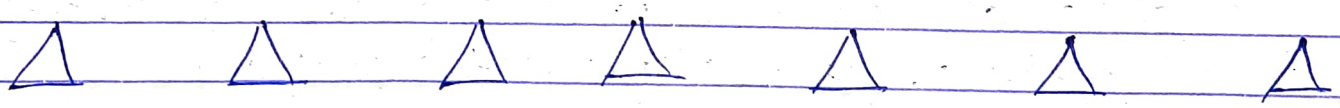
Lifecycle Objective Milestone

Lifecycle Architectural Milestone

~~Life~~ Initial Operational Capability milestone

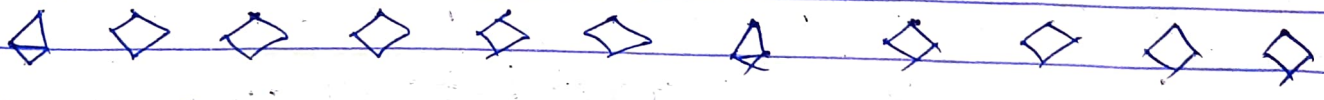
Prod Release Milestone

Major milestone [Strategic focus on global concern of the whole s/w prod].



Minor milestone [focus on the local concern of the current iteration]

Status assesment



[periodic synchronization of stakeholder expectations]

PAGE

DATE

[I] Major Milestones:

↳ This is the system-wide evaluation event which is performed at the end of each phase of the software development life cycle.

↳ These milestones can be used in various process model and even in the conventional waterfall model.

↳ It helps in providing visibility to the system-wide issues.

↳ Helps in synchronizing the management and engineering prespective

↳ Helps to verify if the targets in each phase has been successfully met (or not).

↳ They are used in convencing every stakeholder on the present status of the project.

↳ This milestone also ensures that the requirement understanding, Lifecycle planing, product's functionality and quality are balanced.

↳ Ensures the consistency between different artifacts.

[pg: 128]

PAGE

DATE

[II] Minor Milestone:

↳ These milestones generally focuses on the local concern of the product in each iteration.

↳ They are also called micro milestones.

↳ They are used by the project manager to maintain the control of activities of each day and keeps the development progress on track.

↳ They are iteration focused events that are conducted to review data or content of each iteration in detail.

↳ They divide the elapsed time of major milestones into short interval of time.

↳ ~~This~~ The minor milestone makes sure and gives us confident that we are on right track to achieve the ~~glob~~ major milestone.

↳ Earlier iteration simply focuses on analysis and design whereas the later iterations focuses on completness, consistency, usability and change management.

PAGE

DATE

[III] Status assessment:

↳ Status assessment are periodic events that provides management with frequent and regular insight into the progress being made in the software development.

↳ This event generally provides mechanism that is useful for addressing, communicating and resolving issues and problems regarding management, technical and project risks.

↳ They are done to address and have check on progress and quality indicator ensuring continuous monitoring to the dynamics of the project.

↳ It also provides communication between all the stakeholders.

PAGE

DATE

Deliverable	Milestone
1) They are tangible concrete product or service (i.e. piece of workable s/w).	1) Milestone can be tangible or intagible service that is not delivered to end user.
2) It signals the completion of the project.	2) It signals the checkpoint reached in the project development cycle.
3) It is concern of the client or user.	3) It is concern of the developer and management team.
4) It is showcased to the client and tested on the real environment for acceptance of the client.	4) It is used in status monitoring in order to synchronize the management and engineering prespective.
5) The feedback from the client are used in product improvement.	5) The feedbacks from the stakeholders are used in development process improvement.

PAGE