Benefits
- Improve productivity
- Reduce busy work
- Avoid recursive work
- Support Integrity → Maintains quality
- ~~Effective Communication~~

DATE

Process Automation:

↳ Process automation refers to the use of machine or softwares for performing some repetitive and recurring tasks during software development & testing.

↳ These s/w / machines can perform the task on their own without direct assistance of the people which allows them to focus on other essential tasks.

↳ To an iterative process, process automation and change management is a very critical task where if the change is expensive, the development team resist and won't allow the change.

↳ There are various tools for automation.

WF	Env tools & Process Automation
Management	Workflow automation, Metrics Automation
Environment	change management, document automation
Requirement	Requirement Management
Design	Visual modelling
Implementation	Editor, compiler, debugger
Assessment	Test automation, defect tracking
Deployment	Defect tracking
Process	Organization Policy
life cycle	Inception Elaboration Const^{n} Transition

PAGE

DATE

↳ These tools are essential across overall software process and corelates very well to the process frameworks.

↳ Each tools is mapped to the process workflow and each workflow has a distinct role for automation support.

1) Management:
→ Tools used are Software cost estimation Tools, ~~WBS~~ ~~WBS~~ WBS tools.
→ The Workflow mgmt sys provides the useful & flexible tools to improve the way you work in an efficient manner.

2) Environment:
→ Automating development process and also developing an infrastructure for supporting different project IOF are essential in engineering activities.

3) Requirement:
→ If the process wants strong traceability among requirements and designs then architecture is optimizes the requirement traceability.
To make the process highly-effective, the process automation is used.

4) Design:
→ Visual modeling is used in design workflow where it generally captures the design models,

PAGE

DATE

represent them in human-readable format and also translates into the source code.

5) Implementation:

→ The major focus in this workflow is to write and test the software as per the SRS docs.

→ The relies on tools like editors, debuggers and compilers.

6) Assessment & Deployment:

→ This includes test automation & test management tools.

→ Also Defect tracking tools.

Tools:

Project Evaluation & Review Technique

Tool	Example:
1) planning tool	— PERT tool, estimation tool spreadsheet
2) Editing tool	— Text editors, diagram editors, word editors.
3) Change mgmt tools	— Requirement traceability tools; change control systems.
4) Configuration mgmt tools	— Version mgmt systems, system building tools.
5) Language processing Tools	— Compiler, interpreter
6) Program analysis tools	— static analysis, dynamic analysis
7) Testing tools	— Test data generators, Testing automation
8)	

DATE

~~Configura~~ Change management:

↳ It is a systematic approach to deal with the changes (or modification) of an organizational goal, plan, process and technologies.

↳ The major goal of change management is to strategically implement the changes made to the development process without affecting the development cost, schedule and other part of the process.

↳ Change management must take into consideration:
- > How an adjustment or replacement will affect (or impact) the process and system,
- > Will it impact the employee within the organ².
- > Will it affect the cost & schedule of our project.

↳ It includes:

(1) Software Change Orders:

→ The atomic unit of software work that is authorized to create, modify, or obsolence components within a configuration baseline is called (SCO).

→ This is a key mechanism for partitioning, allocating & scheduling the software work against an established software baseline.

→ It is also used in accessing the progress and quality.

→ The basic structure/fields of SCO are title, descriptions, metrics, resolution, assessment & disposition

PAGE

DATE

11) Configuration Baselines:

→ A Configuration Baseline is an agreed description and review of attributes of product, that afterward serve as basis for further development and defining the change and these changes can only be done by conducting a formal review and procedures.

→ i.e. After a milestone is achieved, it is formally reviewed and once it is accepted, it becomes the baseline for further development.

→ Thus, A baseline is a milestone and reference point in s/w development that is marked by completion or delivery of one or more software configuration items. ~~and formal aproval~~

→ It is task of SCM (S/W Configuration Management) to maintain integrity of set of products.

→ Aims of baseline:

1) It can reduce and control vulnerability.
i.e. Weakness of projects that can easily lead the changes to be uncontrolled.

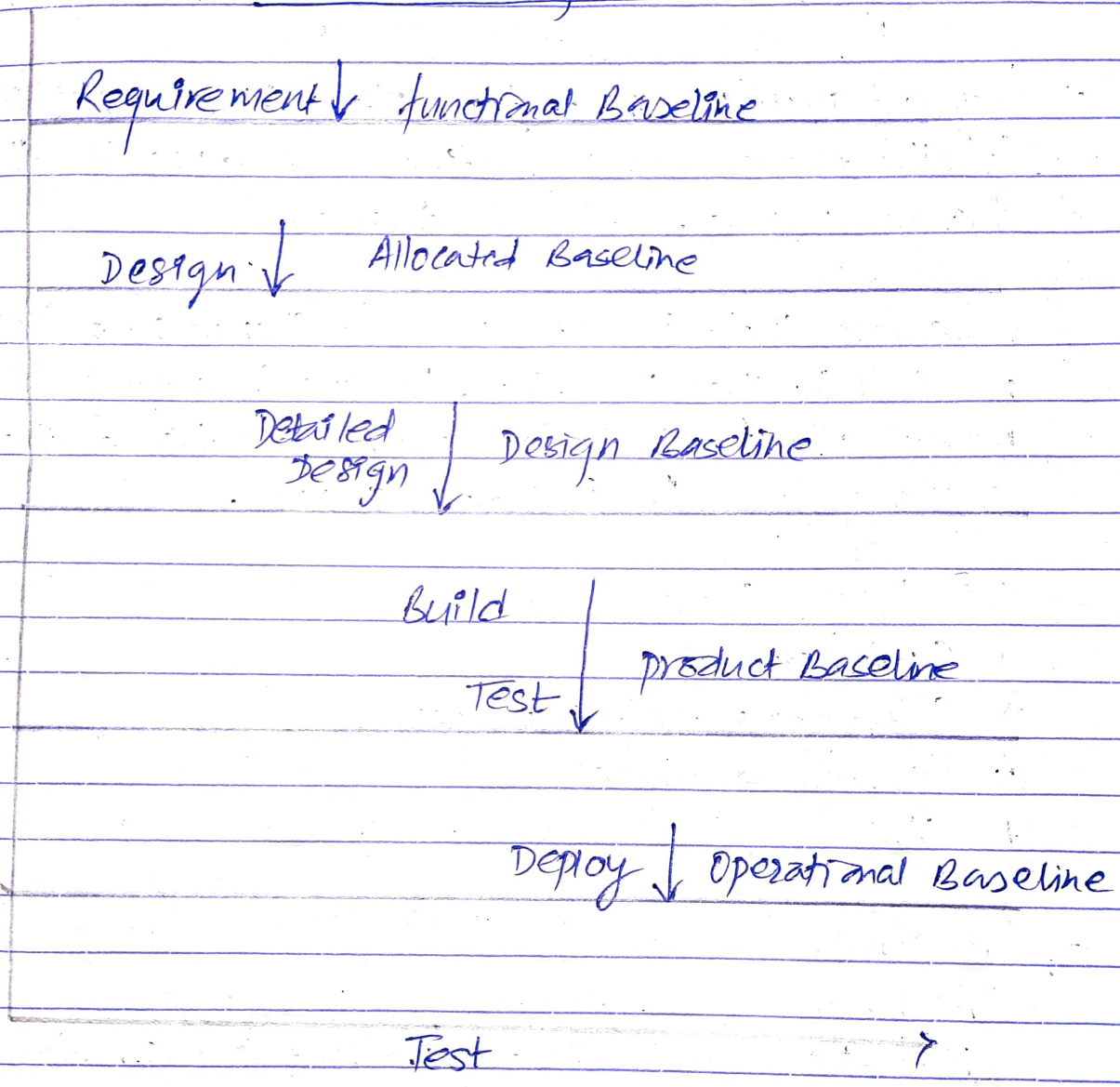
2) - This can be achieved by fixing & changing configuration items in the development life cycle of product.

PAGE

DATE

2) Each elemement that is associated with the baseline must be kept under formal change control.

Baseline Progression



↳ 3 levels of baselines releases:

ⓐ Major release : It represents the major change in the baseline and a new generation of product or project.

ⓑ Minor release : It represents same basic products but with enhanced features, performance & quality.

ⓒ An interim : A release corresponding to developmental configuration that is intended to be transient

DATE

iii) Configuration Control Board:

→ The configuration control board (CCB) is a team of people that functions as the decision making authority on the content of configuration baselines.

→ They are qualified group of people with responsibility for the process of regulating and approving the change made to the baselines.

→ Whenever there is any software change order (SCO) received by the CCB, then they perform various review and analysis to determine whether the changes must be included (or not).

→ This board includes:
a) Customer,
b) Software project manager,
c) system engineer
d) User

→ Procedure:
i) Proposed:
- First the SCO is created/prepared and submitted to the CCB.
- The proposed docs must include the technical description of the problem and an estimate of resolution effort.

PAGE

DATE

2) Accept, Reject and or Archieve:

→ The CCB assigns a unique identity and accepts, rejects or archieve the proposed change.

→ If accepted, the changes are included in the next release.

→ If rejected, the changes cannot be included to the release

→ If archieved, the changes are noted but will be included in the future release.

→ Now, after complete verification of the SCO fields, the CCB assigns the SCO to a responsible person for resolution.

3) Progress:

→ The responsible person analyzes, implements and tests the solution as per the SCO.

→ This includes, updating docs, release notes and also prepare more SCO is necessary.

→ Now, submits the solution to assessment team for verification.

4) Assessment:

→ The independent testing teams make sure that the SCO is completely resolved and if so, the SCO is submitted to CCB for final disposition & closure.

5) Closed:

→ When the development organization, independent test organization and CCB agrees the SCO is resolved, it is transitioned to a closed status.

PAGE

DATE

Process Customization:

↳ Process Customization is the process of tailoring (or customizing) the overall process of software development to meet the specific goal & objective as per the requirement defined.

↳ The tailoring of the project activities can be done on the basis of their scale.

Rank	1	2	3	4	5	6	7
Small scale	Design	Implementation	Deploy	Rqwirc	Assess	Mgmt	Env
Large scale	Managmt	Design	Req	Assess	Env	Impl	Deploy

High Technical Complexity
> More domain experience
> longer inception & Elaboration
> More iteration for risk mgmt
> less predictable cost & schedule

Low management complexity
> less emphasis on risk mgmt
> less process formality
> More emphasis on individual skills
> longer production & transition phase

High mgmt complexity
> Emphasis on risk mgmt
> More process formality
> More emphasis on team work
> longer inception & elaboration

> More emphasis on existing SOP
> Shorter inception & elaboration
> Predictable cost & schedule
> Fewer iteration.

Low Technical Complexity

PAGE

DATE

Core Metrics:

↳ Metrices are the indicators (or statistics) that we can use in our project that enables us to improve our understanding by simply removing the unnecessary decision and make well informed decisions.

↳ These metrices are the key to success which helps in improving how projects are managed and delivered to the end user.

↳ This can also help in decision making during cost & schedule estimation and increases the accuracy over time.

↳ Advantages:

(i) Helps in measuring or calculating and understanding the maturity of organization.

(ii) They help in managing project and resource in more managed way.

(iii) Provides estimates for cost, budget and schedule of the project.

(iv) Can be used in determining the project risks.

↳ Metrices for modern Process:

(1) Management indicator
- Work & progress,
- Budgeted cost & expenditures,
- Staffing and team dynamics

PAGE

DATE

(f) Quality indicter:
- Change Traffic & Stability
- Breakage and modularity
- Rework and adaptability
- MTBF and maturity.

(g) Management indicater:

→ These indicater are generally used for those activities or process which are managed and using these metrices the management task becomes smoother.

→ (a) Work and progress:
→ This indicates the work that is preformed over given period of time.
→ It is used in planning the next cycle, next phases or next iteration.
→ Generally uses, SLOC, functional point Analysis, object points, scenario, test cases, SCOS.

(b) Budgeted cost and expenditures:

→ It includes the cost that is incurring over the time for the development of the product.

→ This gives insight to the financial state of the project and make decision accordingly.

→ Generally uses: Cost per month,
full time staff per month,
% of budget expenditure

PAGE

DATE

© Staffing and team dynamics:

→ This deals with the personnel involved in the dev process of the S/w.

→ It analyzes the resource plan vs the actual hiring rate and can be used to make optimum use of resources.

→ Generally uses: People per month added,
People per month leaving.

(9) Quality indicators:

→ They are the key performance indicators (KPI) which are essential for realization of a project.

→ These indicators are needed to carefully monitor that the team are working on proper task and the developed product has maintained quality.

(a) Change traffic & stability:

→ They indicates convergance schedule that means to indicate convergance points in schedule where two or more activities come together and explain dependencies of their successor activity.

→ Generally uses: SCO opened vs SCO closed,
by type (1/2/3/4/5)

PAGE

DATE

(b) Breakage and modularity:

→ This indicates the average breakage and modularity per change over the period of time.

→ They indicate convergence, s/w scrap and the quality indicators.

→ Generally uses: Reworked SLOC per change

(c) Rework & adaptability:

→ Average rework done per change over given period of time.

→ Generally uses: Average hours per change

(d) MTBF & maturity:

Mean time between failures (MTBF)

→ Rate of defects over the given period of time.

→ It indicates the test coverage, robustness for us.

→ Generally uses: failure counts, ✓
test hours until failed. ✓

PAGE

DATE

CASE technology

↳ It stands for "Computer Aided Software Engineering".

↳ It is the implementation of computer facilated tools that supports the software development and evolutionary process.

↳ CASE ensure a high-quality and defect-free softwares.

↳ It also provides checkpointing and disciplined approach and helps designers, developers, testers, managers and other involved personnel to see the project milestone during development.

↳ Advantages:

(i) Allows all the personnel to view the project milestone easily.

(ii) The overall project quality of product is improved since an organized approach is used during the process development.

(iii) Chances of meeting real world requirement are high.

(iv) Provides organizations with competitive advantages.

↳ Disadvantages:

(i) High cost ~~and to~~

(ii) learning curve

(iii) Tool mix → Appropriate selection of tool is required.

PAGE

CASE Tools:

1) Diagramatic Tools
2) Computer Display report generators
3) Analysis tools
4) Central Repository
5) Documentation Generator
6) Code Generator

DATE

↳ CASE classification:

→ This allows us to understand different types of CASE tools and their contribution in our project activities.

→ 1) Functional ~~classification~~ Prespective:

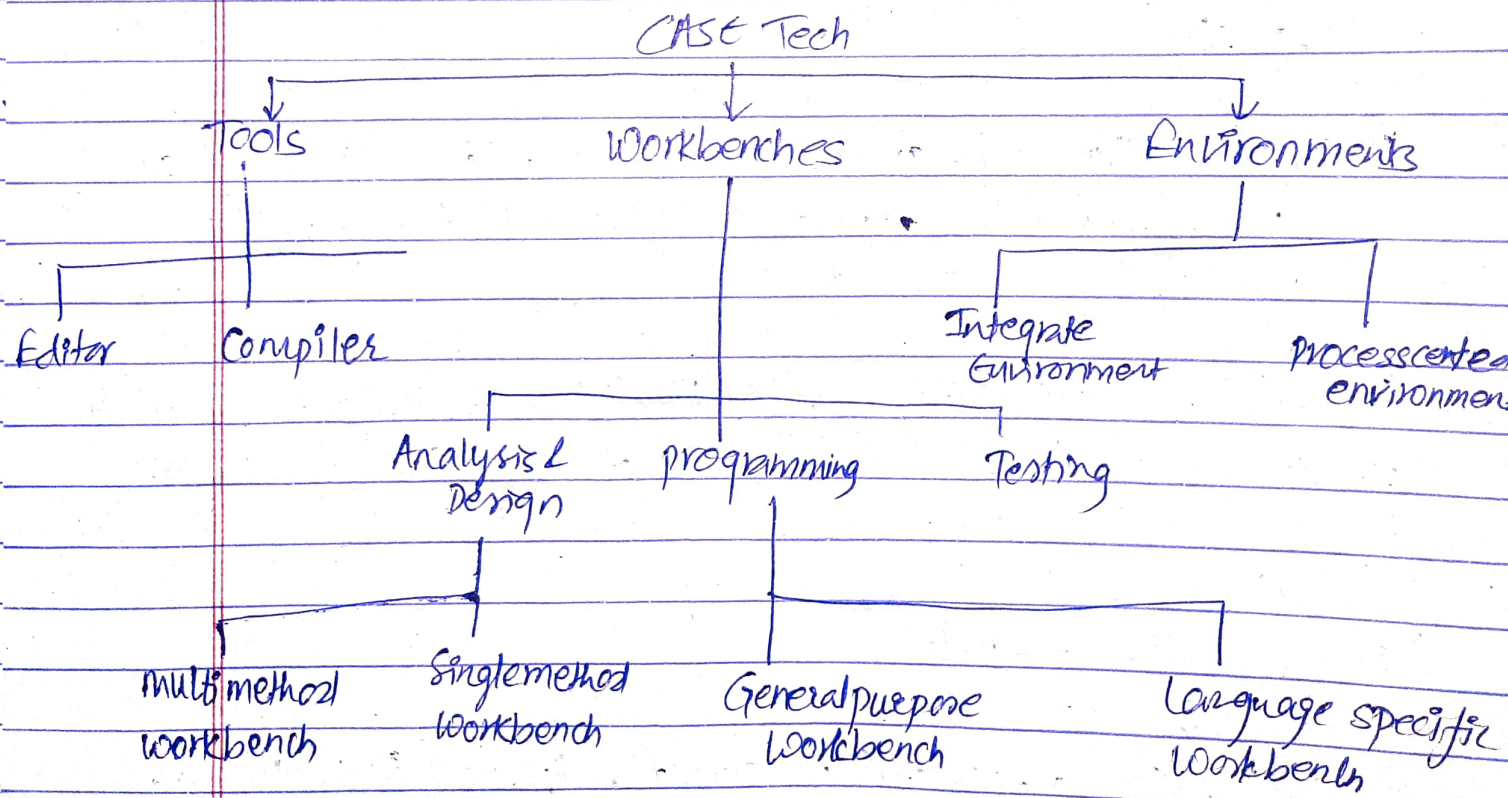
— Tools are classified according to their specific functionalities.

2) Integration perspective:

— Tools are classified according to their organization into integrated unit.

3) Process perspective:

— Tools are classified according to their process activities that are supported.



PAGE

DATE

(8) How would you tackle issues related to mgmt of complex software?

↳ The project that are related to building new technologies and requires higher budget, time and resources can be termed as a complex software.

↳ The issues in complex S/W can be:
(a) lack of resources,
(b) improper communication,
(c) scope creep
(d) late risk resolution,
(e) Budgeting issue,
(f) unrealistic deadline
(g) ~~Inad~~ Risk

↳ Solutions:
(a) Getting the design write by focusing on the architecture first.
- The expert architect must be hired who is responsible for creating a efficient architecture for the project.
- ~~If the~~ With a good architecture team, an average developer team can succeed.
But, with a poor architecture team, team of experts also cannot succeed.

(b) Managing risk through iterative process.
→ The risks must be frequently monitored, tracked & control.
→ Also, implement necessary risk mitigation process.

PAGE

DATE

(c) Reducing the complexity of process by using Work Breakdown structure:

→ The complex problem must be divided/decomposed into smaller units and individually implemented.

(d) Proper Change management:

→ The formal change management process must be applied.

→ Software change order (SCO) must be prepared before changing is done which is submitted to CCB who are responsible to analyze and approve the change proposed.

(e) Avoiding the overhead of book keeping by using Round Trip engineering:

→ RTE is used to maintain consistency and integrity between multiple artifacts representing similar information in different ways.

PAGE

Ch#4

DATE

Cultural Shifts: / Modern Process Transition

↳ Several cultural shifts must be overcome for sucessfully transition of conventional software management to the modern software management.

↳ It Involves:

① "lower level and Mid-level managers are performers"

→ There should be no puremanager in an org^n with less than or equal to 25 personnel.

→ A indicator that the project is in ~~trouble~~ trouble is noticed when the manager donot take part in planning & ownership of the project and they are affected by the transition.

② Requirement and design are fluid and tangible:

→ The ~~transition~~ conventional development focused to much on producing documents to describe the project progress rather than the engineering aspects.

→ The transition to the modern approach with less documents driven environment will be embraced by the engineers & but opposed by the traditional contract manager.

③ Ambitious demonstrations are encouraged:

→ The purpose of the early testing / life cycle demonstration is to expose flaws in our initial estimation but not to put the project to ~~dead~~ facade.

PAGE

→ Stakeholder might overreact due to the earlier mistakes, digressions or immature design. And, the mgmt team might ~~release~~ likely resist the transition.

→ But, the user, developer and engineering team will embrace the transition.

(iv) Good and Bad project performance is much more obvious earlier in the life cycle.

→ In an iterative development, success breeds success and early failure are extremely risky to turn around.

→ Failure on the early stage can create serious problem in the development of the process.
So, a best team of software architecture must be hired inorder to do planning and architecture design which makes it easier and effective for the implementation by the prod² team.

→ But, most organization has scarce resource for early estimation.

(v) Early increment will be imature:
→ Initial increments of the product is always not perfect and work as per the expectation.

→ The user & customer might not get impressed by the increment but the stakeholder accepts the imaturity as a natural process of development.

DATE

(VI) Artifacts are less important earlier and more important later:

→ It is a waste of time to worry about the details of the artifacts until a baseline is achieved.

→ Since, the project might evolve during the life cycle which causes the initial artifacts to be obsolated.

→ The development team will embrace the transition whereas the traditional contractor monitor might resist the change.

(VII) Real issues are surfaced & resolved systematically:

→ On a healthy project making progress, it should be easy to differentiate bet= real and apparent issues.

→ Depending on the situation, a cultural shift could affect almost all the team members.

(VIII) Quality assurance is everyone job and not a seperate discipline.

(IX) Performance issue arises early in the life cycle.

(X) Investment in automation are necessary.

(XI) Good s/w organization would be more profitable.

(XII)