SOFTWARE PROJECT MANAGEMENT

* Project Management and CMM (Capability Maturity Model: Guidelines for Improving the Software Process)

- to consistently improve process performance (the actual result achieved in a project executed using the process) on projects, you must enhance the process capability (the range of results that can be expected in a project when it is executed using a process); the process itself must become more mature.

* Maturity levels in the CMM

- each maturity level (except level 1) is characterized by KPAs (Key Process Areas).

Level 5 - Optimizing
- Process Change Management
- Technical " "
- Defect prevention

Level 4 - Manged
- S/W Quality Management
- Quantitative Process "

Level 3 - Defined
- Organization Process Focus
- " " Definition
- Training program
- integrated S/W Management
- S/W Product Engineering
- Intergroup Coordination
- Peer Reviews

Level 2 - Repeatable
- Requirements Managements
- S/W project planning
- S/W " tracking & oversight
- S/W Subcontract management
- SQA (S/W Quality Assurance)
- S/W Configuration management

Level 1 - Initial

* Project Management Process:
1. project planning stage / phase
2. " execution " "
3. " closure " "

1. project planning (the major activities of the project manager, in this phase, are):
   = perform startup and administrative tasks
   = Create a project plan and schedule -
   - define the project objectives
   - identify a suitable standard process for project execution
   - tailor the standard process to meet project requirements
   - define a process for managing changes in requirements
   - estimate the effort

- plan for human resources and team organization
- define project milestones and create a schedule
- define the quality objectives and a quality plan to achieve them
- make a defect prevention plan
- identify risks and make plans to mitigate them
- define a measurement plan for the project
- define a training plan for the project
- define project-tracking procedures

= perform a review of the project plan and schedule
= obtain authorization from senior management
= define and review the configuration management plan
= orient the project team to the project management plan

---

2. Project execution:

= execute the project as per the project plan
= track the project status (and status report)
= review the project status with senior management (report)
= monitor compliance with defined project process
= analyze defects and perform defect prevention activities
= monitor performance at the program level
= conduct milestone reviews and replan if necessary

---

3. project closure (postmortem)

= a systematic wind-up of the project after customer acceptance
= to learn from the experience so that the process can be improved
= post-project data analysis
= metrics analysis
= process assets (materials, such as templates and guidelines, used to aid in managing the process itself) are collected for future use
= lessons are recorded
= create project closure report

* The project planning infrastructure =

key elements:

① Process Database (PDB) contains the performance data of completed projects. It contains data on risks, effort & effort distribution, defect & defect distribution, size, and other project characteristics.

To use the information in PDB during planning, project managers often find information about similar projects particularly useful. Similarity check can be done, based on, languages used, platforms, database used, tools used, size and effort in the project.

The data captured in the PDB can be ~~used~~ classified as follows:

- project characteristics {name, project managers names, module leaders, business units, process deployed, application domain, hardware platform, language used, DBMS used, brief statement of goals, risks information, duration, team size}
- project schedule {expected and actual start and end dates}
- project effort {data on initial estimated effort and the total actual effort, distribution of actual effort among various stages, such as project initiation, requirements management, design, build, unit testing, and other phases}
- size (LOC, line of code)
- defects (number of defects found in various defect detection activities and number of defects injected in different stages)

② Process Capability Baseline (PCB) summarizes the process performance across projects, thereby specifying the range of results to be expected if the processes are followed. It contains measures such as

- delivered quality
- productivity
- schedule
- effort distribution
- defect injection rate
- in-process defect removal efficiency
- cost of quality
- defect distribution

③ Process Assets and The Body Of Knowledge system:

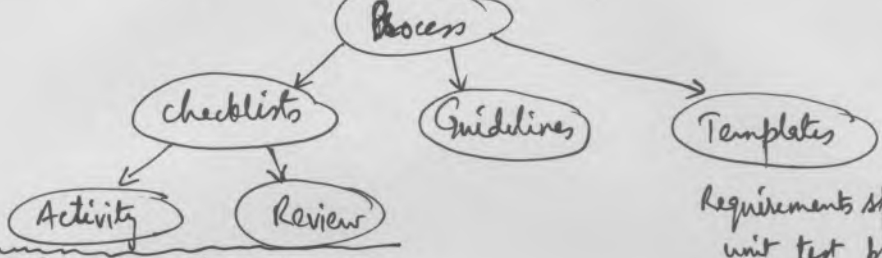
process assets are:

— checklists (activity checklist = a list of activities that constitute a process step
{ review checklist = for reviewers to draw attention to defects that are likely to be found in an output.)

— Guidelines for executing a step

— templates/forms (provide the structure of documents in which the output of a process or step can be captured)

They improve productivity by reducing effort required to do some tasks, and improve quality by reducing defects or catching them early.



Requirements analysis checklists
unit test and system test plan "
configuration management "
status report "
Requirement review "
Functional design review "
project plan " "
Code " "

Requirements specification document
unit test plan "
Acceptance test " "
project management plan
configuration " "
Metrics analysis report
Milestones status "
defect prevention analysis report

Many organizations have developed systems to effectively leverage the collective experience and knowledge of their employees. A system called Body of Knowledge (BOK) is used to encapsulate experience.

The web-based BOK system has its own keyword or author-based search facility. The knowledge in BOK, which is primarily in the form of articles, is organized by topics, and are related to lessons learned and best practices.

* Process Planning =

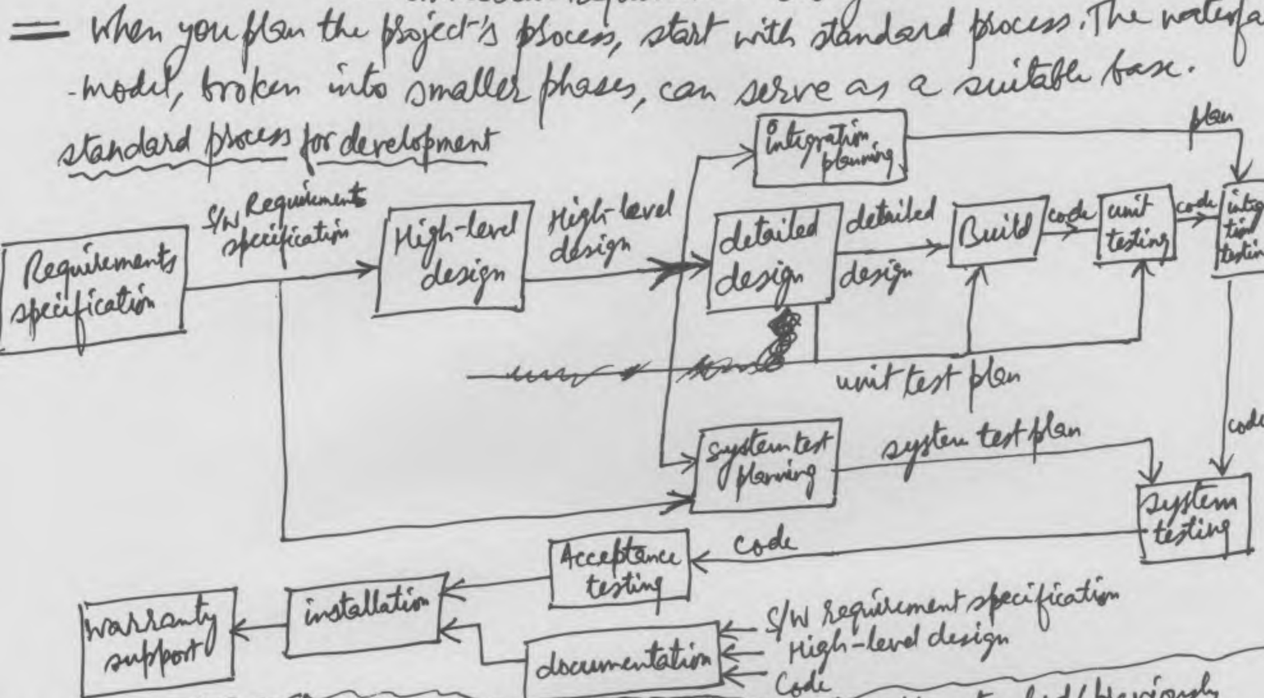
* The Development Process:

The main process planning issue in a project is designing the development process to be used for building the software to satisfy the customer.

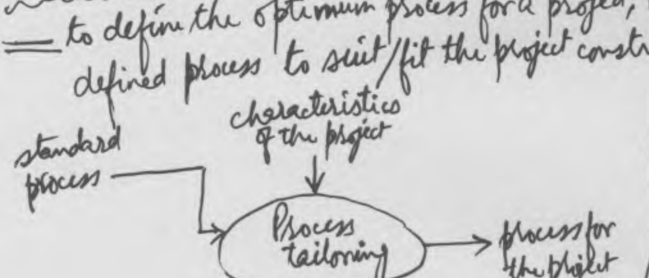
This process is supported with a change management process to accommodate requirement changes.

= When you plan the project's process, start with standard process. The waterfall model, broken into smaller phases, can serve as a suitable base.

standard process for development



= to define the optimum process for a project, tailor (adjust) the standard/previously defined process to suit/fit the project constraints (particular business or technical needs).



* perform tailoring at two levels: summary and detailed

①. **summary level**: apply overall guidelines for tailoring the standard process (depending on project characteristics) i.e. provide some general rules regarding certain types of detailed activities. The characteristics are,

* experience and skill level of the team & project manager
* peak team size
* clarity of the requirements
* project duration
* application ~~criticality~~ criticality

②. detailed tailoring of activities. Tailoring guidelines can help.

* Requirement change management process:

assesses the impact of each change request and also keeps track of the cumulative impact.

the steps are:

1. Log the changes. (change request form & log)
2. perform an impact analysis on the work products.
3. estimate the effort needed for the change requests.
4. Reestimate the delivery schedule
5. perform a cumulative cost impact analysis.
6. review the impact with senior management if thresholds are exceeded
7. obtain customer sign-off
8. rework work products

* Effort estimation and scheduling:

Goal: to generate reasonable estimates that will work most of the time.

- Use past data to estimate. Prefer data from similar projects to general process capability data. Use a model to estimate, but allow flexibility for adjusting estimates to accommodate project-specific factors.
- employ different models in different situations. Bottom-up estimation is effective when project details are known. Use top-down approach if you can estimate the size and productivity, and the use-case approach when using a use-case-based development approach.
- For the overall schedule and the high-level milestones, use the existing flexibility to meet proposed dates. Once the overall schedule and milestones are fixed, determine the resource requirement for each phase from the phase-wise effort estimate.
- Detailed scheduling is a dynamic task; take into account people issues while assigning tasks. It is not necessary to completely refine the schedule at the start. You can develop details for the tasks in the overall schedule as the need arises.
- The detailed schedule forms the planned activity list for the project. Capture all activities planned in the project in this document and use it later to track activities.

*= Effort Estimation and scheduling =

effort estimation models : defines the project characteristics whose values (or their estimates) it needs and the ways these values are used to compute the effort.

- A common approach is to use a simple equation to obtain an estimate of the overall effort from the size ~~of~~ estimate. This equation can be determined through regression analysis of past data on effort and size. Then, once the overall effort for the project is known, the effort for various phases or activities can be determined as a percentage of total effort.

Many models have been proposed that use this top-down approach to estimation, with the COCOMO model being the most famous. Models using function points (instead of LOC) as size units have also been built.

In the bottom-up approach, you obtain the estimates first for parts of projects and then for the overall estimate. That is, the overall estimate of the project is derived from the estimates of its parts. One bottom-up method calls for using some type of activity-based estimation.

Estimating schedule : once the effort is known or fixed, various schedules (or project duration) are possible, depending on the number of ~~resources(people~~ resources (people) put on the project. For example, for a project whose effort estimate is 56 person-months, a total schedule of 8 months is possible with 7 people. A schedule of 7 months with 8 people is also possible, as is a schedule of approximately 9 months with 6 people.

* Procedure for estimation (bottom-up estimation) :

1. identify programs in the system and classify them as simple, medium, or complex (S/M/C). As much as possible, use either the provided standard definitions or definitions from past projects.
2. if a project-specific baseline exists, get the average build effort for S/M/C programs from the baseline.
3. if a project-specific baseline does not exist, use project type, technology, language, and other attributes to look for similar projects in the process database. Use data from these projects to define the build effort of S/M/C programs.

4. If no similar project exists in the ~~process~~ process database and no project-specific baseline exists, use the average build effort for S/M/C programs from the general process capability baseline.

5. Use project-specific factors to refine the build effort for S/M/C programs.

6. Get the total build effort using the build effort of S/M/C programs and the counts for them.

7. Using the effort distribution given in the capability baseline or for similar projects given in the process database, estimate the effort for other tasks and the total effort.

8. Refine the estimates based on project-specific factors.

Steps in top-down estimation approach:

1. Get the estimate of the total size of the software in function points.
2. Using the ~~productiv~~ productivity data from the project-specific capability baseline, from the general process capability baseline, or from similar projects, fix the productivity level for the project.
3. Obtain the overall effort estimate from the productivity and size estimates.
4. Use effort distribution data from the process capability baselines or similar projects to estimate the effort for the various phases.
5. Refine the estimates, taking project-specific factors into consideration.

Use-case points estimation approach:

The use-case points approach ~~employed~~ is based on the approach from Rational and is similar to the function points methods. This approach can be applied if use-cases are used for requirement specification.

1. Classify each use-case as simple, medium, or complex. The basis of this classification is the number of transactions in a use-case, including secondary scenarios. A transaction is defined to be an atomic set of activities that is either performed entirely or not at all. A simple use-case has three or fewer transactions, an average use-case has four to seven transactions, and a complex use-case has more than seven transactions.

a simple use-case is assigned a factor of 5
medium " " " " 10
Complex " " " " 15

2. Obtain the total unadjusted use-case points (UUCPs) as a weighted sum of factors for the use cases in the application. That is, for each of ~~these~~ the three ~~cas~~ complexity classes, first ~~the obtain~~ obtain the product of the number of use cases of a particular complexity and the factor for that complexity. The sum of ~~the~~ the three products is the number of UUCPs for the application.

3. Adjust the raw UUCPs to reflect the project's complexity and the experience of the people on the project. To do this, first compute the technical complexity factor (TCF) by reviewing the factors given as:

Factor	weight
distributed system	2
Response or throughput performance objectives	1
end-user efficiency (online)	1
complex internal processing	1
Code must be reusable	1
Easy to install	0.5
Easy to use	0.5
Portable	2
easy to change	1
concurrent	1
includes special security features	1
Provides direct access for third parties	1
Special user training facilities required	1

and rating each factor from 0 to 5. A rating on 0 means that the factor is irrelevant for this project; 5 means it is essential. For each factor, multiply its rating by its weight from the above table and add these numbers to get the TFactor. Then,

$$TCF = 0.6 + (0.01 * TFactor)$$

Factor	weight
Familiar with internet process	1.5
Application experience	0.5
Object-oriented experience	1
lead analyst capability	0.5
motivation	1
stable requirements	2
Part-time workers	−1
difficult programming language	−1

~~and add these numbers to get the TFactor. Then,~~

~~$TCF = 0.6 + (0.01 * TFactor)$~~

4. similarly, compute the environment factor (EF) by going through above table and rating each factor from 0 to 5. For experience-related factors, 0 means no experience in the subject, 5 means expert, and 3 means average. For motivation, 0 means no motivation on the project, 5 means high motivation, and 3 means average. For the stability of requirements, 0 means extremely unstable requirements, 5 means unchanging requirements, and 3 means average. For part-time workers, 0 means no part-time technical staff, 5 means all part-time staff, and 3 means average. For programming language difficulty, 0 means easy-to-use programming language, 5 means very difficult programming language, and 3 means average. The weighted sum gives the EFactor, from which the EF is obtained:

$$EF = 1.4 + (-0.03 * EFactor)$$

5. compute the final use-case points (UCP):

$$UCP = UUCP * TCF * EF$$

For effort estimation, assign on an average, 20 person-hours per UCP for the entire lifecycle. This will give a rough estimate. Refine this further as follows.

Count how many factors are less than 3 and how many factors are greater than 3. If the total number of factors that have a value less than 3 are few, 20 person-hours per UCP is suitable. If there are many, use 28 person-hours per UCP. In other words, the range is 20 to 28 person-hours per UCP, and the project manager can decide which value to use depending on the various factors.

for example: if UCP = 218, then,

$218 * 20 = 4360$ person-hours = 499 person-days (at 8.75 hr/day)

or, 513 person-days (at 8.5 hrs/day)

SCHEDULING activity can be ~~broke~~ broken into two subactivities:

1. determining the overall schedule (the project duration) with major milestones

often, a rule of thumb, called the square root check, to check the schedule of medium-sized projects,

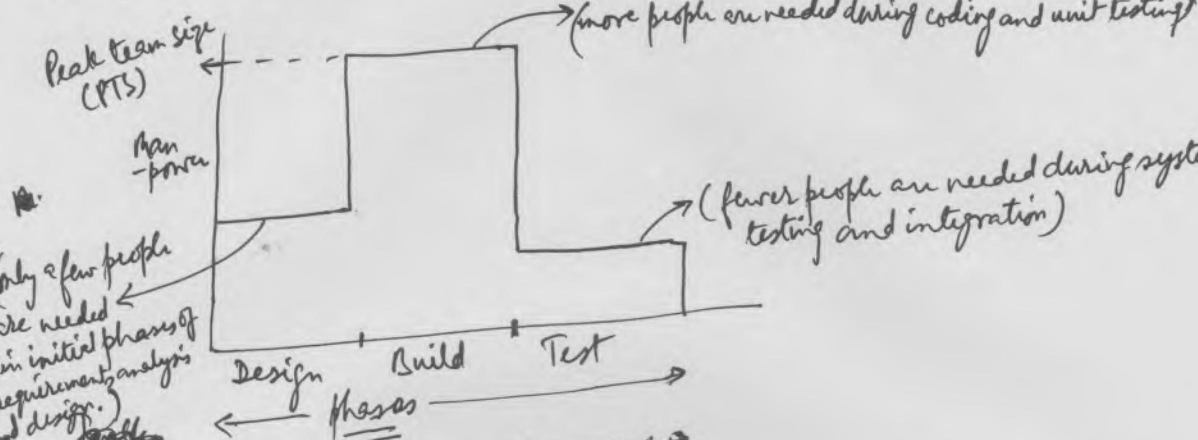
$$(\text{proposed})\ \text{schedule} \approx \sqrt{\text{effort (in person-months)}}$$

the schedule can be met if ($\sqrt{\text{effort}}$) resources are assigned to project.

For example: if effort estimate = 50 person-months, then a schedule of about 7 to 8 months will be suitable with about 7 to 8 full-time resources.

effort = schedule X resource

Once the overall duration (schedule) of the project is fixed, the schedule for the major milestones must be determined. To determine the milestones, you must first understand the manpower ramp-up that usually takes place in a project. The number of people in software project tends to follow the Rayleigh curve. In the beginning and the end, few people work on the project; the peak team size (PTS) is reached somewhere near the middle of the project.



This approach can lead to some people being unoccupied at the start and towards the end. This slack time is often used for training. Project-level training is generally needed in the technology being used and the business domain of the project, and this training consumes a fair amount of effort. Similarly, the slack time available in the end can be utilized for documentation and other closure tasks.

generally, 40% of schedule → design (20% for high-level, 20% for detailed)
40% of schedule → build
20% of schedule → integration & system testing

and manpower (1:2:1) for design, build, integration & testing
effort distribution (1:4:1)

2. detailed scheduling :- break the tasks into small schedulable activities in a hierarchical manner.

* Quality Planning =

- ensure that final software has few defects
- quality control procedures are planned and executed
- a quantitative quality goal is set for the project, and execution of the process is monitored quantitatively.

✓ = manage quality by using the number of <u>defects</u> as the metric of quality

✓ = set quality goal for a project in terms of the number of defects during acceptance testing. Use past data on process capability to set this goal.

✓ = using past data, estimate the defect levels for the various defect detection stages in the process. Compare these estimates to the actual number of defects found during project execution to see whether the project is progressing satisfactorily toward achieving the goal or whether some correction is needed.

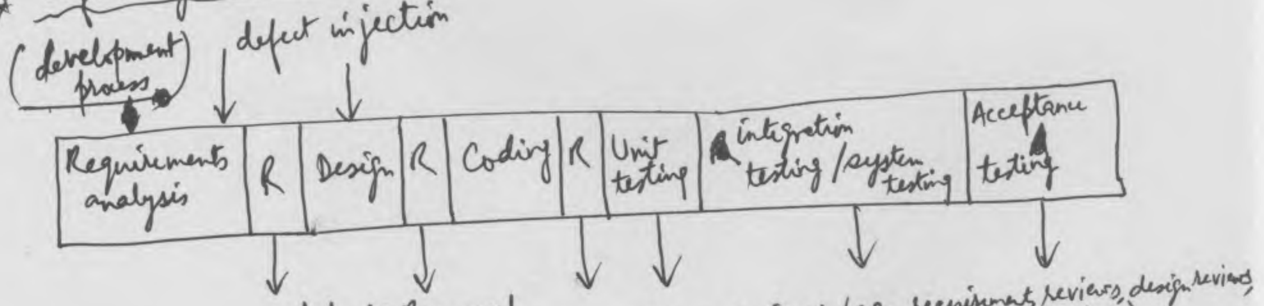
✓ = in addition to testing, plan for <u>reviews</u>, clearly specifying the review points, review items, and review types.

✓ = if the quality goal of project is higher than past performance, it can not be achieved using the same process as earlier projects. To achieve the higher goals, you must enhance the process.

✓ = use defect prevention as a strategy to achieve higher quality and productivity goals in a project. For defect prevention, identify the defect prevention team, the points at which defect analysis will be done, and the expected benefits.

---

* <u>defect injection and removal</u>:



defect Removal
by quality control activities of <u>reviews</u> (e.g., requirements reviews, design reviews, coding reviews)
and testing

* cost of defect removal increases as the latency of defects increases.

Quantitative approaches to quality managements —

~~defect remov~~

~~- defect removal efficiency (DRE) metric~~

- defect prediction (set quality goal in terms of delivered defect density.) (set intermediate goals by estimating the number of defects that may be identified by various defect detection activities; then compare actual number of defects to the estimated defect levels)
- statistical process control (SPC) ~~for~~ (set performance expectations of the various QC processes, such as testing and reviews, in terms of control limits. If the actual performance of the QC task is not within the limits, you analyse the situation and take suitable action. The control limits resemble prediction of defect levels based on past performance but can also be used for monitoring quality activities at a finer level, such as review or unit testing of module.)

Quantitative quality management planning:

- setting the quality goal: use data from similar projects in the past data (PCB) ~~or~~

or, follow the steps: (quality goal in terms of defects per function point):

1. set the quality goal in terms of defects per FP (function point)
2. estimate the expected productivity level for the project
3. estimate the size in FP as (expected productivity * estimated effort)
4. estimate the number of AT (acceptance testing) defects as (quality goal * estimated size)

---

quality goal in terms of the process's defect removal efficiency:

quality goal in terms of defect removal efficiency.

1. set the quality goal in terms of defect removal efficiency.
2. estimate the total number of defects from the defect injection rate and the estimated size, or by the effort-based defect injection rate and the effort estimate.
3. estimate the number of AT (acceptance testing) defects from the total number of defects and the quality goal.

Defect prevention planning :- activities steps are :

= identify a defect prevention team within the project
= have a kick-off meeting and identify existing solutions
= plan for defect prevention
- set defect prevention goals for the project
- see that DP (defect prevention) team is trained on DP and causal analysis, if needed
- define the frequency at which defect prevention activities will be carried out
= do defect prevention
- at defined points, collate (~~collect~~ collect and combine) defects data
- identify the most common types of defects by doing Pareto analysis
- perform causal analysis and prioritize the root causes
- identify and develop solutions for the root causes
- implement the solutions
- Review the ~~status~~ status and benefits of DP at project milestones
= Capture learning
- in the metrics analysis report and BOK (Body of knowledge), capture the learning and benefits you have obtained.
- submit all outputs of DP as a part of the process assets

* Risk management:

- identify risks by a list of commonly occurring risks. In addition, look ahead and try to visualize everything that can go wrong in the project.
- For risk prioritization, a simple and effective mechanism is to classify the probabilities of risks and their impacts into categories such as low, medium, and high. and then manage the risks that have high probabilities and impact.
- For the top few risks, plan the risk mitigation steps, and ensure that they are properly executed during the project.
- monitor and reevaluate the risks periodically, perhaps at milestones, to see whether the risk mitigation steps are having an effect and to revisit risk perception.

Concepts:

* risk management entails additional cost.
* it is not easy to measure the value of risk management.

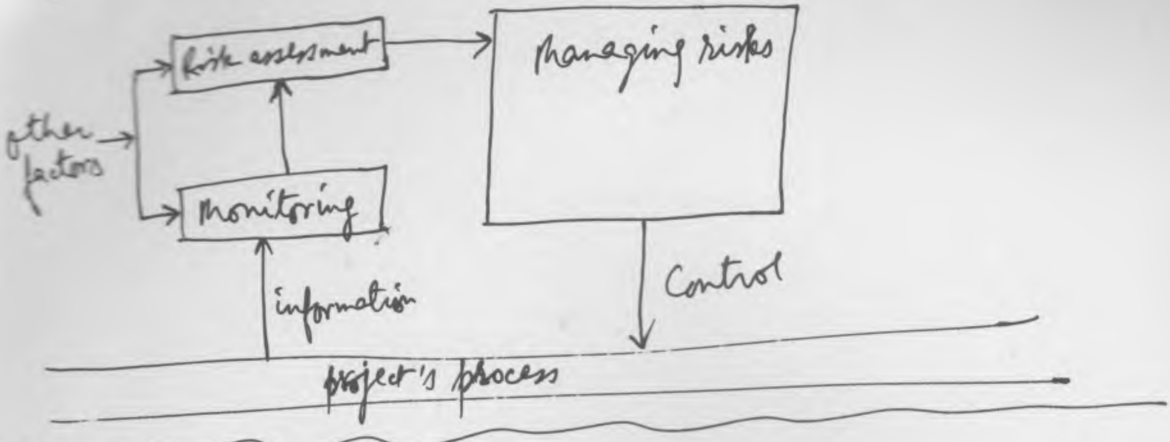
(for prioritization) the expected value of the loss for the risk is called

$$\text{risk exposure} = RE(R) = \underbrace{Prob(R)}_{\text{probability of risk, R occurring}} \times \underbrace{Loss(R)}_{\text{total loss incurred}}$$

: Risk management Activities:

- Risk management
  - Risk assessment
    - Risk identification
    - Risk analysis
    - Risk prioritization
  - Risk control
    - Risk management planning
    - Risk resolution
    - Risk monitoring

Risk management and project execution :—



risk prioritization =

1. for each risk, rate the probability of its happening as low, medium, high. If necessary, assign probability values in the ranges given for each rating.
2. For each risk, assess its impact on the project as low, medium, high or very high. If necessary, assign a weight on ~~the~~ a scale of 1 to 10.
3. rank the risks based on the probability and effects on the project; ~~for example~~
4. select the top few risk items for mitigation and tracking

Risk categories

Probability	Range
Low	0.0–0.3
medium	0.3–0.7
High	0.7–1.0

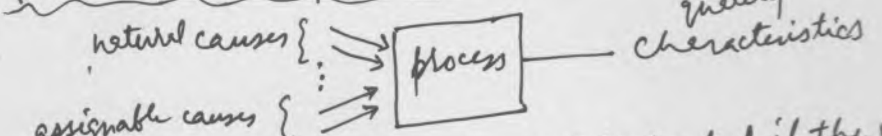
Impact categories

Level of consequences	Range
Low	0.0–3.0
Medium	3.0–7.0
High	7.0–9.0
very High	9.0–10.0

= Measurement and Tracking planning =

- during project planning, you must decide how you plan to monitor the progress of the project.
- plan to measure size, schedule, effort, and defects. These suffices for most software projects.
- classify effort in a few categories, and collect effort data using an automated system with activity codes for each category. To avoid inaccuracies due to poor memory recall, log effort data frequently.
- log defects and track them to closure. For a defect, also record its type, detection stage, injection stage, and severity to support analysis such as defect removal efficiency, delivered quality, and defect injection rate.
- For performance analysis at milestones, establish acceptable limits for performance variation from planned for effort, schedule, and defects. During project execution, if the performance goes beyond these limits, management intervention may be warranted.

Process monitoring through statistical process control (SPC):



assignable causes occur once in a while, have large influence over variability in process performance. These must be eliminated. ✓

A process is said to be under statistical control if the variability in the quality characteristics is due to natural causes only. The goal of ~~statistics~~ SPC is to keep the production process in statistical control.

Control charts are favorite tool for applying SPC. To build a control charts, the output of a process is considered to be a stream of numeric values representing the values of characteristic of interest. Subgroups of data are taken from this stream, and the mean values for the subgroups are plotted, giving an x-bar chart. A low control limit (LCL) and an upper control limit (UCL) are established. If a point falls outside the control limits, the large variability is considered to be due to assignable causes. Another chart, called an R-chart, plots the range (the difference between the minimum and maximum values) of the chosen subgroups. ....

By convention, LCL and UCL are frequently set at 3-sigma around mean, where sigma is standard deviation - - -

When a point representing an output falls outside the control limit. Generally, two types of actions are performed:

① rework the output so that it has acceptable characteristics—that is, take corrective action.

② conduct further analysis to identify the assignable causes and eliminate them from the process—that is, take preventive actions.

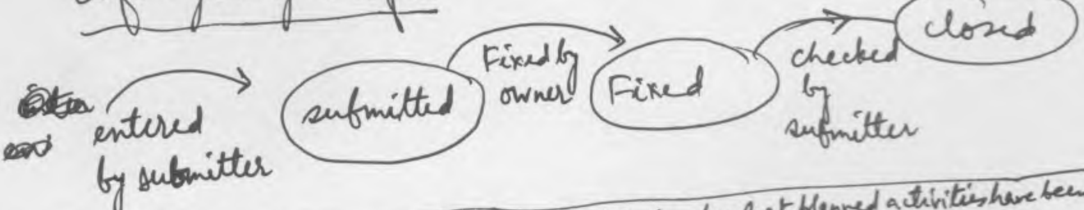
Measurements:

Collecting effort data:

to help project manager monitor the effort, each employee records in a weekly activity report (WAR) system and effort spent on various tasks. This online WAR system entry consists of a sequence of records, one for each week. Each record is a list of items, with each item containing the following fields:

- Program code
- module code
- activity code (type of activity)
- activity description
- Hours for Monday through Sunday

Life cycle of a defect



Project Tracking:

- activity tracking (looks at what planned activities have been completed)
- defect "
- issues "

to keep track of project's status along the effort, schedule, and quality dimensions, project manager also plan for:

- activity-level monitoring
- status reports
- milestone reports

The structure of the project management plan =

the project management plan template has four major sections:

- project summary (a high-level ~~section~~ overview of project.)
    - start and end dates
    - project leader
    - contacts at customer end
    - project objectives
    - major commitments made to customer on milestones and deliverables
    - ~~assumptions~~
    - assumptions made
    - details of billing
- project planning (lists the outputs of executing the various project-planning procedures)
    - development process being used
    - tailoring notes
    - requirement change management process
    - " traceability plans
    - effort and schedule estimates along with their basis
    - people requirement by skill, role, monthwise, or a combination of these.
    - development environment needed
    - tools employed
    - training plans
    - quality plans
    - risk management plan
- project tracking (measurements to be taken and the systems to be used for recording data, various project tracking activities to be undertaken, the frequency and nature of progress reporting, escalation procedures.)
- project team (project team and its structure, roles and responsibilities of various people)

Configuration Management =

A software product typically consists of many programs and documents, and these items change and evolve before they are ready for the final system. For this reason, S/W configuration management is an important issue.

- define CM process so that it lets projects handle concurrent updates, undo a change, obtain the latest version of a program, determine the status of a program, and prevent unauthorized changes. Use version control, change request tracking, and library management mechanisms to support these capabilities.
- develop a CM plan seperately from the project management plan. The CM plan must specify the environment, configuration items and their naming conventions, storage areas for the items in different states, and the method of managing changes to the items, including version numbering and reconciliation, access control, and release and backup policies.
- perform CM audits and status checking to ensure that CM plan is being followed.

goal of CM is to manage the evolving configuration of the S/W system. ie, ~~system~~ systematically controlling the changes ~~so that~~ – those due to evolution of ~~wst~~ work products and those due to requirement changes

Planning and setting up configuration management =

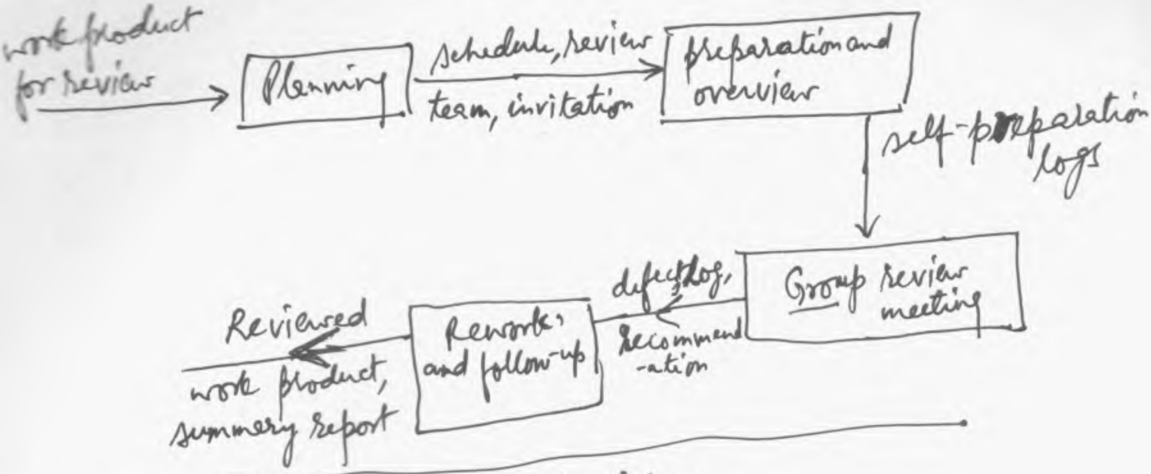
- identify configuration items, including customer-supplied and purchased items.
- define a naming and numbering scheme for the configuration items.
- define the directory structure needed for CM
- define access restrictions
- define change control ~~plan~~ procedures.
- identify and define the responsibility and authority of the CC or Configuration Control Board (CCB)
- define a method for tracking the status of configuration items
- define a backup procedure
- define a reconciliation procedures, if needed
- define a release procedure
- define an archival procedure
- identify points at which the configuration items will be moved to the baseline

* Reviews =

The purpose of a review is to identify defects and issues a ~~●~~ work product through a process of ~~[illegible]~~ formal and structured review by a group of peers. Review are cost effective and can be applied even to work products that can not be executed. Reviews are an important ~~[illegible]~~ technique for improving both quality and productivity as well as for providing visibility into the state of the project.

- include external experts in review team to argument the talent of project team.
- use well-defined and structured review process with a clear ~~guidance~~ guidelines and formal data collection. The process should include planning, self-review, and a group ~~of~~ meeting.
- during the review, focus exclusively on finding defects and issues. Defects and issues are resolved later.
- when it is more practical, use a one-person review for work products. For the one-person review, follow the same process and data collection guidelines as for group reviews.
- monitor each review for effectiveness. Create performance expectations from past data, and use them to evaluate a ~~review~~ review's effectiveness.
- if a review's performance is not as expected, analyze ~~and causes the~~ the causes and take corrective and preventive actions.
- ~~to~~ understand the impact of reviews, conduct simple experiments within the project. Data from within their own organization convinces people in ways that no amount of outside data can.

The Review process (inspection)



Guidelines for reviews in projects =

- the project management plan
- the requirement specification
- the system test plan
- the high-level design
- the integration test plan

self-preparation log

project code:
work product ID:
Reviewer name:
Effort spent for preparation (hours):
Issue list:

SL#	Location	Description	Criticality	Seriousness
1.				

Group review meeting log

project code: meeting type:
moderator: Scribe:
Author: Reviewer(s):
SEPG member: Observer(s):
Date: work product ID:
Effort spent on review meeting (person-hours):
Defects to be closed by (date):
Defect List:

SL#	defect location	type	Severity	Stage injected	Descrip-tion

open Issues log:

SL#	Issue Description	Assigned to	Targeted date	closed date

Monitoring and control:

- use statistical process control (SPC) concepts implemented through control charts.

① The Review Capability Baseline:

- set the control limits for the various parameters and then use that range to determine the effectiveness.

Control limits have been determined for the following performance parameters: the coverage rate during preparation, the coverage rate during the group review meeting, the defect density for minor or cosmetic defects, and the defect density for serious or major defects (the overall defect density is simply the sum of the two preceding defect densities). These limits are determined from past data and from the review capability baseline.

② Analysis and control Guidelines:

Summary report of a review

Project:

work product type:	project plan, v.1.0
size of product:	14 pages
moderator:	meera
Reviewer(s):	Biju, meera
Author:	JC

Effort (person-hours)

a. Overview meeting	0
b. Preparation	10 person-hours
c. Group review meeting	10 person-hours
Total effort	20 person-hours

Defects

number of critical defects	0
" " major "	3
" " minor "	:
" " cosmetic "	:
" " defects detected during preparation	
" " " " group review meeting	
" " open issues raised	
Total number of defects	

Result moderator reexamination

Recommendations for Next phase

unit to undergo group review

" " one-person "

Comments (moderator)

Prepared by: ; Date: _ _

Project monitoring and control:

When a plan is executed, regardless of how carefully the planning was done, things frequently do not work out as planned. With proper monitoring, a project manager can check whether or not the project is progressing as planned. If it is not progressing along the desired path, control must be applied to ensure that the project still meets its objectives.

- track the completion of scheduled activities, the defects found, and the issues that come up. Use a weekly status report for regular tracking and reporting.
- at project milestones, compare the actual values for schedule, effort, and defects with the estimated values. If the deviation exceeds the predetermined threshold, take corrective and preventive actions, if the situation warrants. Also, revisit the risks and situations that affect risks.
- evaluate some tasks immediately after they have been executed and take corrective actions if the performance is not within the expected range, as determined from past data. Reviews and unit testing are best suited for this level of tracking
- Analyze the defect data from the first few modules in the project to understand the root causes of the defects. Then take actions to eliminate the root causes. Later, repeat this analysis to understand the impact of defect prevention.
- Audit the project formally for compliance with the defined processes. Based on the noncompliance reports, take corrective and preventive actions.

Apply management control

execute project

gather measurements

Analyze and predict

[project monitoring and control cycle]

Project tracking :

- Activities tracking
- defect "
- issues "
- status report =
  - customer complaints
  - milestones achieved this week
  - milestones missed " " and the reasons for them
  - " planned for the next week
  - issues requiring clarification or attention
  - escalation, if any
  - estimated work **versus** available time by milestone

Status summary

Project	life-cycle stage	Next Milestone Date	% complete	Number of resources	Remarks
..	..	..	..	..	..

tasks completed and missed

task completed	task missed

task planned for the week 4 July, 2000 To 9 July, 2000

— — —

— — —

~~issues / milestones~~ ~~mist~~

issues / miscellaneous items

open issues : - -

misc. items : - .

milestone analysis:

- actual versus estimated analysis of effort and schedule
- monitoring quality
- risk-related monitoring

~~milestone analysis~~

milestone analysis report

milestone name: date:

schedule (for the milestone)

planned date	Actual date	slippage
--	--	--

Reasons for deviation:
Actions taken to bring schedule back in control:
overall impact on project:

Effort (person-hours)

estimated effort before this milestone	Actual effort before this milestone	deviation	Estimated effort from Milestone to project end	Effort available until project end

Reasons for deviations:
Actions taken:
overall impact on project:

defects:

no.	QC Activity	size of work product	estimated effort from MSP (person-hours)	Actual effort (person-hours)	estimated defects	Actual defects	defect deviation
-	-	-	-	-	-	-	-

Reasons for deviation:
Action taken:
Impact on quality Goals:
Defect prevention activities:

Requirements change Tracking

total number of major requirement changes to date
" " " minor " " " "

Risks

Previous state						Current status		
serial no.	Probability	impact	exposure	Risk item	status of mitigation Action	Probability	impact	Exposure
1.	-	-	-	-	-	-	-	-

customer complaints

number of customer complaints raised
" " " " closed
" " " " Open

Training

Planned :
Actual :

Group Reviews:

Planned :
Actual held:

issues :

---

* Defect Analysis and prevention :-

= performing Pareto Analysis:

1. list all defects identified so far
2. calculate the total number of defect by type
3. sort defects by type in descending order of number of defect
4. calculate the percentage of each defect type with respect to the total number of defects detected
5. identify the defect type that is the cause for about 80% of the total defects

= performing causal Analysis:

cause-effect diagram (fishbone diagram)

1- clearly define the problem (the effect) to be studied. For defect prevention, it typically is "too many defect of type X"

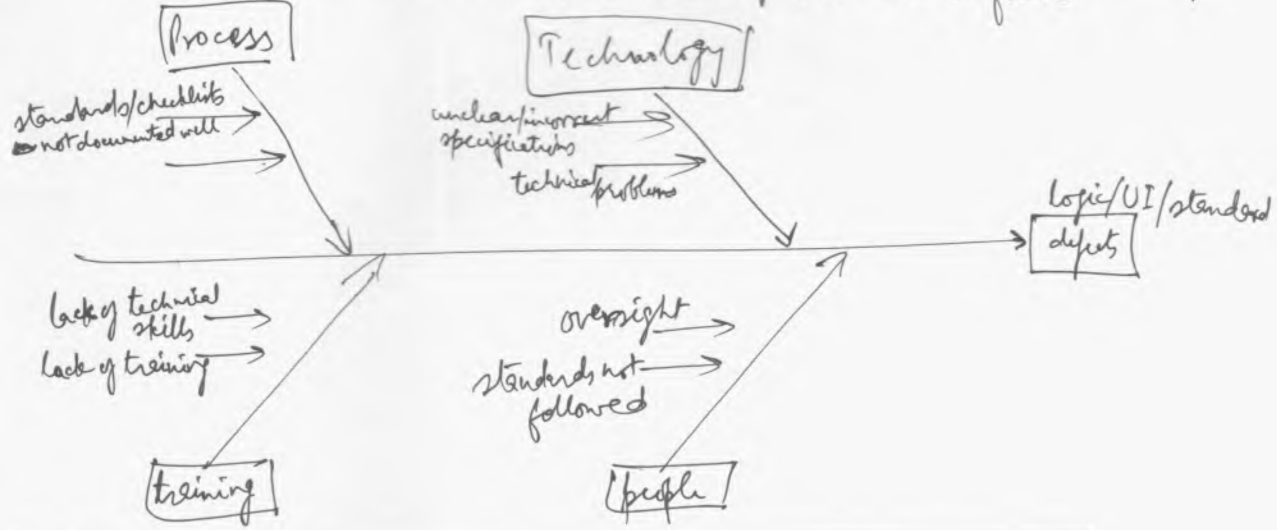
2. draw an arrow from left to right with a box containing the effect drawn at the head. This is the backbone of the diagram.

3. determine the major categories of causes. These could be the standard categories or some variation to suit the problem.

4. write these major categories in boxes and connect them with diagonal arrows to the backbone. These form the major bones of the diagram.

5. Brainstorm for the subcauses of the major causes by asking repeatedly, for each major cause "why does this major cause produce the effect?"

6. add the subcauses to the diagram clustered around the bone of the major cause. If necessary, further subdivide these causes. Stop when no worthwhile answer to the question can be found.



---

* Follow-up Actions:

- noncompliance report with corrective action:

Non-conformity report

Project/Dept: . . .

date: . . .

QSD Ref: . . .

Severity: serious/minor

Non-conformity:

Corrective action:

Action by:

Action date:

Preventive action:

Action by:

Action date:

Auditor:

Auditee:

(signature)

(signature)

Followup action:

Closed by: (signature)

Recommendation:

- noncompliance Report with preventive action:

Non-Conformity Report.

project/Dept.:
ISO 9001 clause:
QSD Ref.:

date:
severity: serious/minor

Non-Conformity

Corrective action:

Action by:
Action date:

Preventive action:

Action by:
Action date:

Auditor:
Auditee:
(Signature)
(Signature)

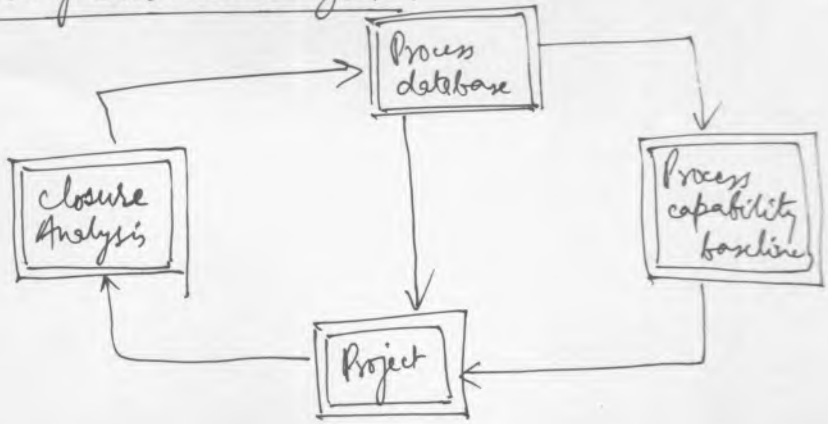
Closed by:
(signature)

Recommendation:

Project Closure Analysis

- keep the project closure analysis metric-based. Analyze the data to understand the performance of the project and the causes for any major deviations. These causes can serve as a source of improvement initiatives.
- metrics analysis should report the final quality delivered, the productivity achieved, the distribution of effort, the distribution of defects, the defect removal efficiency of various quality activities, and the cost of quality.
- collect reusable process assets such as plans, checklists, standards, and guidelines, and make them available for others.

The role of closure analysis:



Closure Analysis Report:

- General and process-related info.
- risk management
- size
- effort
- defects
- causal analysis
- process assets

⇒

1. General info
2. performance summary
3. process details
4. tools used
5. risk management
6. size
7. schedule
8. effort
9. defects
10. causal analysis and lessons learned
11. process assets submitted: