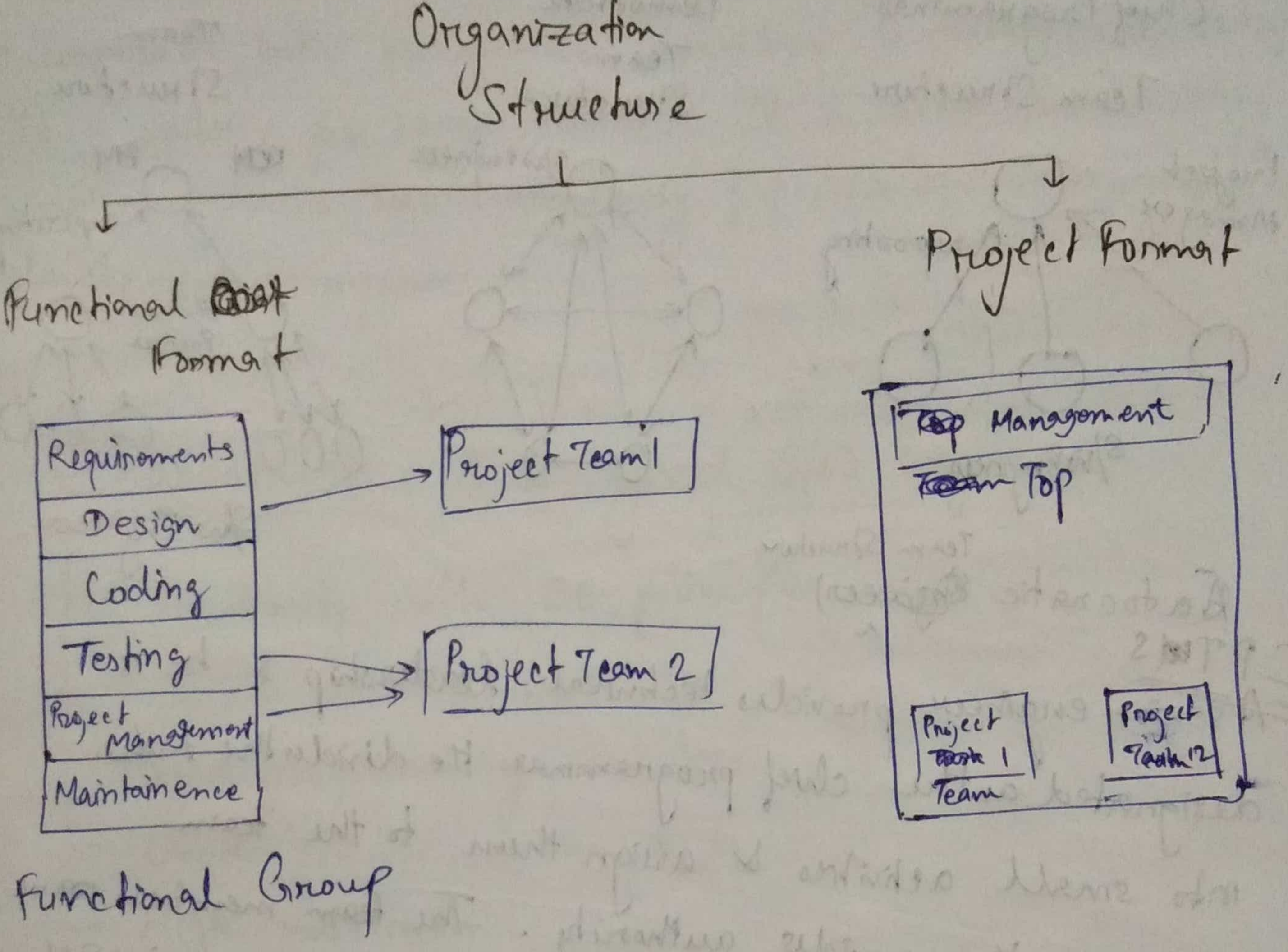
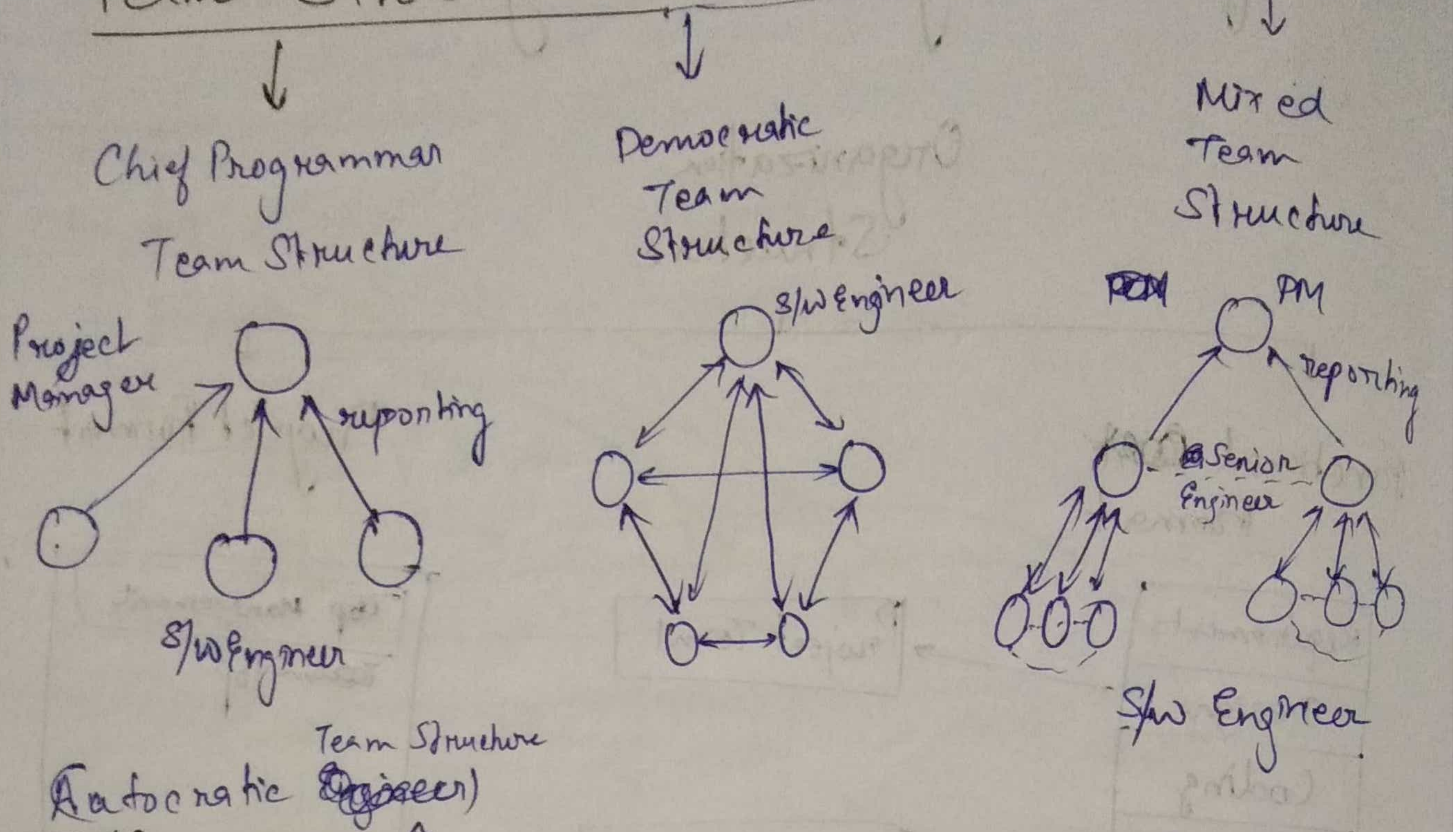
Software Project Monitoring & Control



Advantage of functional format over project — format

- Ease of Staffing
- Production of good quality documents
- Job Specialization
- Efficient handling of the ~~projects control or~~ problems associated with manpower turnover

Team Structure



CPTMS

~~A~~ senior engineer provides technical leadership & he is designated as the chief programmer. He divides the tasks into small activities & assign them to the team members. He provides authority. The team members work under constant supervision of the chief programmer. It inhibits their original thinking, this is subject ~~of~~ to a single point failure since too much responsibilities are shouldered by the chief programmer.

DTS

It doesnot enforce any formal team hierarchy. A manager will provide the administrative leadership & at different times different members will provide technical leadership. It leads to high morale & job satisfaction. It is suitable for projects requiring less than 5 or 6 engineers & categorical R&D projects. For large projects this structure becomes chaotic. It encourages egoless programming.

MTS

It draws upon the ideas of both the chief programmar team structure, and the democratic team structure. The team organization encorporates both hierarchial reporting & democratic setup. It is suitable for large teams. It is suited for handling complex programs. This is extremely popular in many software development companies.

Characteristics of a good S/w Engineer

- Familiarity with SE principles.
- Good domain knowledge
- Good programming activities.
- Good communication skills
- High motivation
- Sound knowledge of fundaments of CS
- Intelligence
- Ability to work in a team.
- Disciplne.

Software Risk Management

- ↳ Categories of Risk
- ↳ How to access a risk
- ↳ strategies to contain a risk
- ↳ Risk leverage
- ↳ Risk Handling in case of schedule slippage

Categories of Risk

i) Technical Risk
ii) Project Risk
iii) Business Risk

Project Risk

- ↳ Budget
- ↳ Schedule of the project
- ↳ Resources - Hardware, Software, Manpower
- ↳ Customer satisfaction

Technical Risk

↳ Qualities are not present which are required by the project

Business Risk

↳ Target customer should not be affected. The item should have no extra excavation.

There should be same extra or unique features in that particular product.

Risk Assessment

$p = r * s$

p = priority with which the risk is to be handled.

r = probability of the risk becoming true

s = severity of damage caused due to the risk becoming true

Risk Containment Strategies

i) Avoid the Risk
ii) Transfer the Risk
iii) Risk Reduction

Avoid the Risk → Sit & Discuss to modify the scope of the project.

* If extra work is needed, the incentive should be given to motivate the employees to work for extra man hours.

Transfer the Risk → LIC policy, life insurance the third party involvement to transfer the risk. Not take ownership but transfer the risk.

Risk Reduction Shortage of manpower in the midway, so there should be a 3rd party who can be trained & fill the shortage of the resigned people.

Risk Leverage

$$RL = \frac{(\text{Risk exposure before reduction} - \text{Risk exposure after reduction}}{\text{Cost of Reduction.}}$$

Risk related to schedules slippage

↳ intangible nature of project (not known, unclear, non transparent)

Software Configuration Management

Release vs Version vs Revision

Release ↓ A new release is created if there is only a bug fix, minor enhancements to the functionality, usability, etc.

Version ↓ A new version of a s/w is created when the is a significant change in functionality technology or the hardware it runs on

Revision → A new revision refers to a minor bug fix in the s/w

Necessity for S/W Configuration Management

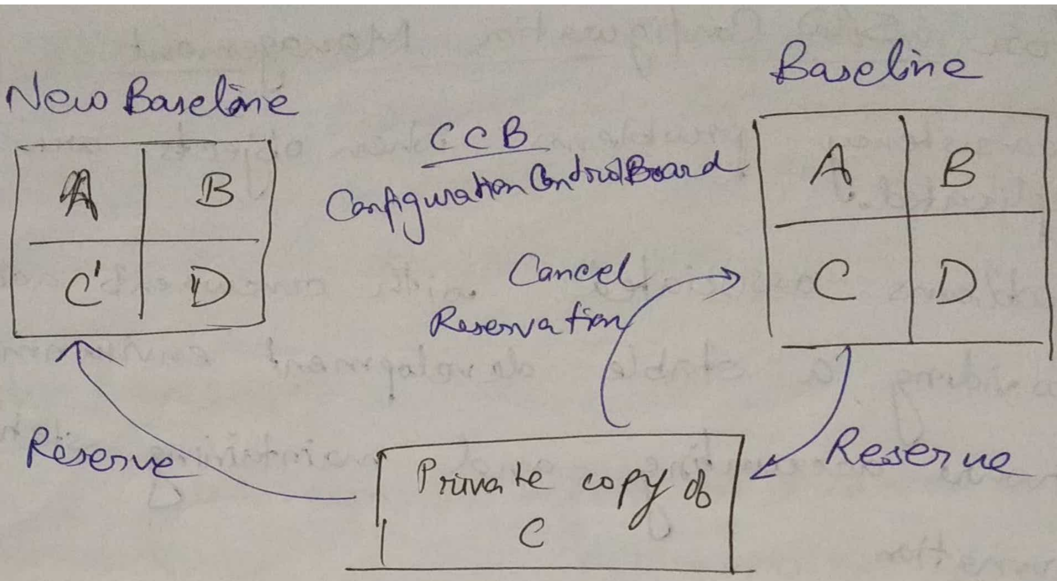
- Inconsistency problems when objects are replicated.
- Problems associated with concurrent access
- Providing a stable development environment
- Software accounting and maintaining status information.
- Handling varients.

S/W Configuration Management Activities

- Configuration identification involves deciding which parts of the system should be kept track of
- Configuration control ensures that changes to a system happen smoothly

The controllable objects include

1) Requirements Specification Document
2) Design documents
3) Tools needed to build the system. that include, compilers, linkers, loaders, parsers, libraries, lexical analyzers.
4) Source code for each module,
5) Test Cases & Test plans
6) DPR (Detailed Project Report)



Reserve & Restore creation in Configuration.

CCB → It is constituted from the development team
(Configuration Control Board) members. for every change that needs to be carried out the CCB reviews the changes & certifies the following things. about the change.

a) Change is well motivated

b) The developers has considered & documented the effects of the change.

c) The changes should interact well with those done by other developers.

d) appropriate people from CCB have validated the change

Software Reliability

Repeatable vs Non-Repeatable Software Development Organization

↓ person-independent

↓ depends on a team of members

↳ trust worthines & dependibility of a software

Reasons for Software Reliability being difficult to measure

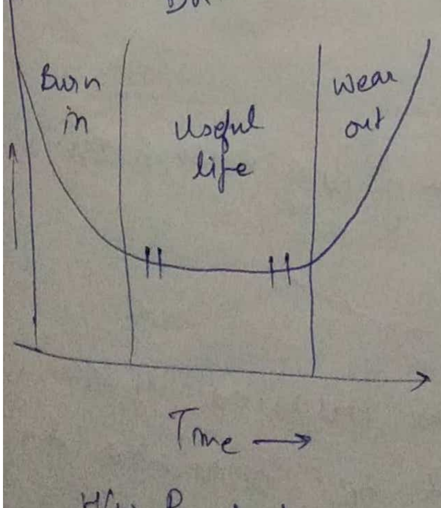
↳ reliability improvement due to fixing a ~~measure~~ single bug depends on where the bug is located in the code.

↳ perceved reliability of a s/w product is highly observer dependent

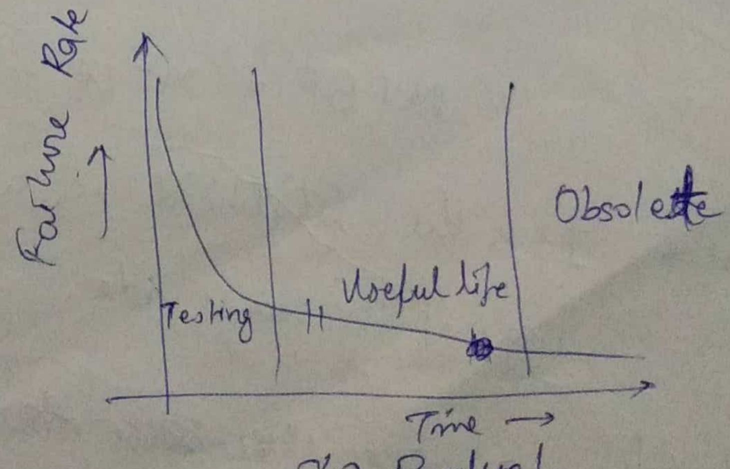
↳ reliability of a product keeps changing as error are detected and fixed.

HW vs SW Reliability

Bathtub Curve



H/w Product



S/w Product

Reliability Metrics

i) ROCOF (Rate of occurence of failure)
ii) MTTF (Mean time to Failure)
iii) MTTR (Mean time to Repair)
iv) MTBF (Mean Time Between Failure)
v) POFOD (Probability of Failure on Demand)
vi) Availability

ROCOF → It measures the frequency of occurence of failure (Rate of failure)

MTTF → $\sum_{i=1}^{n} \frac{t_{i+1} - t_i}{n-1}$ avg time b/w 2 consecutive failures.

MTTR → It the average ~~[illegible]~~ time taken to track the errors that cause the ~~[illegible]~~ failure & fix them

MTBF → It is the sum of MTTF & MTTR

$$MTBF = MTTF + MTTR$$

POFOD → It is the likelihood of a system failing when a service request is ~~[illegible]~~ made.

Availability → ~~when the [illegible] [illegible]~~ It is the measure of how likely the system shall be available for use ~~[illegible]~~ over a given period of time.

Classification of S/w Failures

1) Transient

2) Permanent

3) Recoverable

4) ~~Tolerable~~ Unrecoverable

5) Cosmetic

Transient → failures that occur for certain input values while invoking a function.

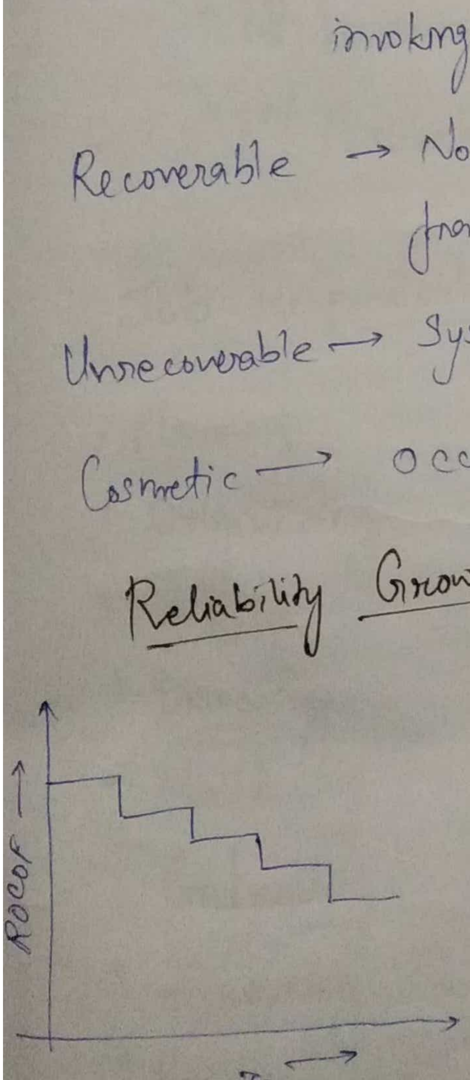
Permanent → failures that occur for all sets of input values for invoking a function.

Recoverable → No amount of manual help needed to recover from the failure

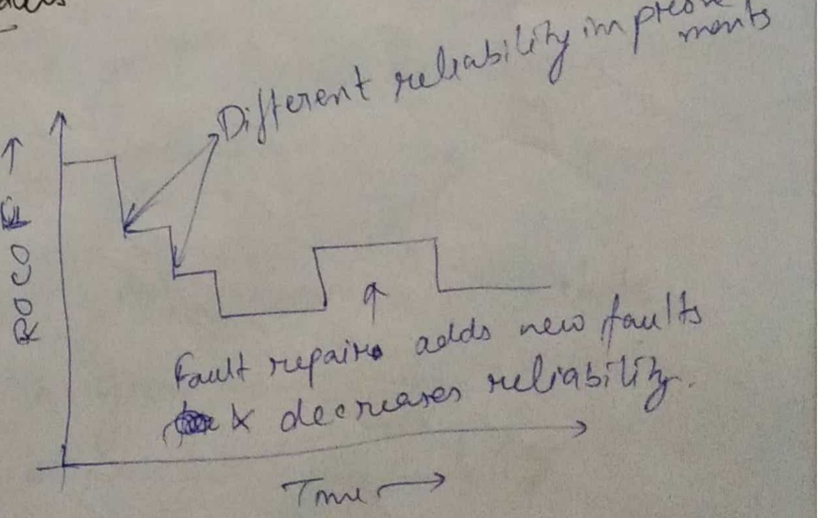
Unrecoverable → System needs to be restarted/rebooted

Cosmetic → occurs due to minor mistakes in the system.

Reliability Growth Models



Jelinsky & Moranda Model



Littewood & Verralls Model