UNIT-1 : Software Process Maturity

- Software maturity Framework
- Principles of Software Process change
- Sofw Process Assessment
- the initial Process
- the Repeatable Process
- the Defined Process
- the Managed Process
- the Optimizing Process
- Process Reference Model Capability Maturity Model (CMM)
  - CMMI
  - PCMM
  - PSP
  - TSP

S/w Process

→ It is a set of related activities
- that leads to the Production of a S/w product.

→ these activities may involve the development of S/w
- from scratch in a standard programming language
- like Java or C

→ [redacted] by

→ there are many different S/w processess
- but all must include 4 activities that are fundamental to engineering.

~~1. S/w Spec~~

1. S/w Specification

→ the functionality of the S/w & the constraints on its operations must be defined.

2. S/w Design & Implementation

→ the S/w to meet the specification must be Produces.

3. S/w Validation

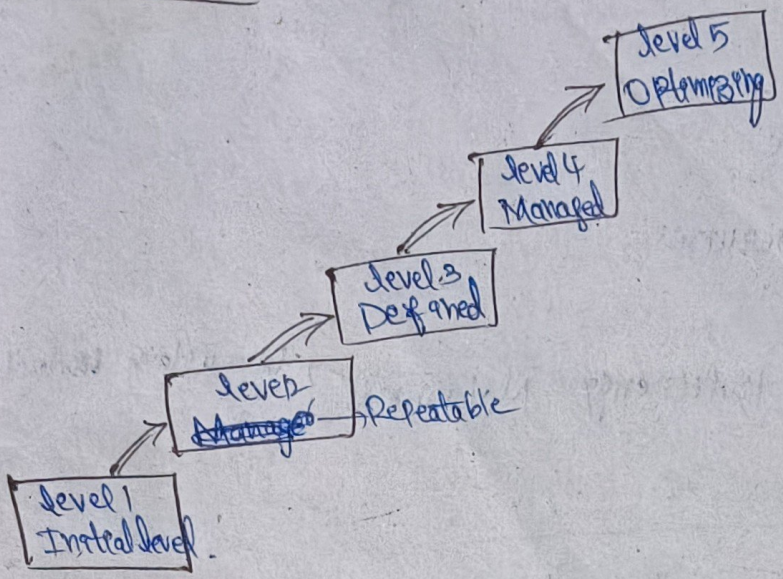
→ the S/w must be validated
- to ensure that it does what the customer want!

4. S/w Process Evolution

→ the S/w must be evolve
- to meet the customer need.

S/w Process Maturity Framework

Characteristics



Level-1: Initial

→ Adhoc, Chaotic → not in proper way, not in plan
→ Unpredictable, cost, time, Poorly controlled, reactive
→ Individual effort

level-2: Repeatable

→ Basic Project mgmt.
→ track cost, schedule, functionality
→ Repeat old experience or earlier success

Level-3: Defined

→ Documentation, Standardization
→ Measure S/w quality

level-4: Managed

→ Process & Product quality collected.

Level-5: Optimizing

→ Deployment, feedback, update.

Software Process Assessment

→ evaluating identify

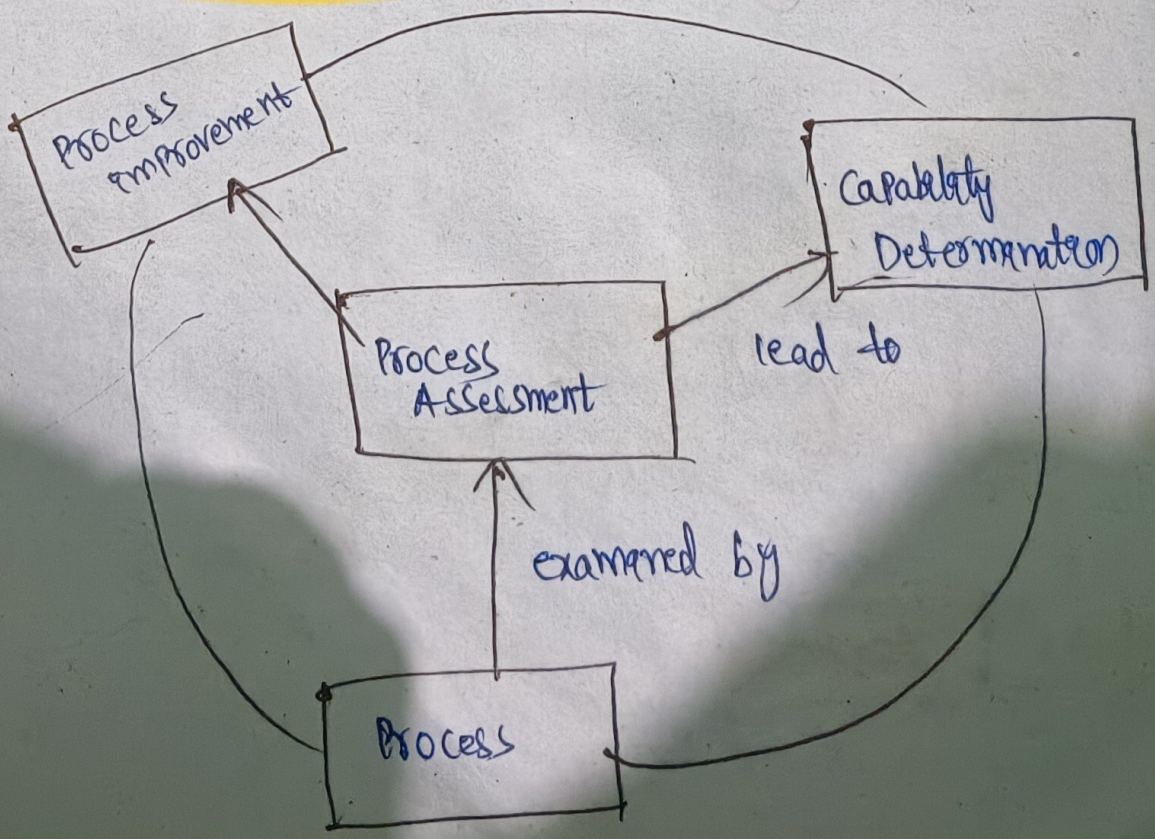
→ identify the failures, lackness

→ If a problem occurs, how it will be solved.

- rectifies problem

⇒ Paulk & Colleagues (1995)

dhintlo kuda 5 levels
initial process,repeatable process,defined process,managed process, optimizing process



CMM

① CMM - Capability Maturity Model

② older model
→ measure maturity of a company

③ Provides only 5 maturity levels
→ gives rating & (tells failure)

④ Measures only maturity
→ benchmark

⑤ No integration of multiple models

⑥ Less focus on continuous improvement.

⑦ Less flexible

⑧ Limited coverage

⑨ Narrow scope

Levels

1. Initial
2. ~~Managed~~ Repeatable
3. Defined
4. ~~Quantitatively~~ managed
5. ~~Maturity~~ Optimizing

CMMI

① CMMI - CMM Integration

② Modern model
→ Same.

③ Provides 5 maturity levels + capability levels.
→ gave reason (where we ~~are~~ lacking)

④ Measures maturity & capability
→ benchmark.

⑤ Integrates all CMM models

⑥ Strong focus on continuous improvement.

⑦ Highly flexible.

⑧ Broader coverage

⑨ Wider scope.

Levels

1. Initial
2. Managed
3. Defined
4. Quantitatively managed
5. Optimizing.

PSP (Personal S/w Process)

1). Self-improvement process
2). helps a developer improve personal skills.
3). Bottom-up approach
4). applies to one person, not the whole team.
5). Focus on personal planning
6). PSP is for individuals
7). improves personal work quality.
8). handles personal tasks
9). Focus on improving the individual work.
10). focus on one developer.
11). Self-managed
12). requires Personal training
13). Includes:
- Personal planning
- Time tracking
- code review
- defect tracking

14). Benefits:
- Better personal improvement

TSP (Team S/w Process)

1). Team-level process.
2). Helps the entire team to work together
3). Top-down approach
4). applies to whole team.
5). Focus on Team Planning.
6). TSP is for Teams.
7). improves team work quality.
8). handles Project-level tasks
9). Focus on improving the team work.
10). focus a full team.
11). managed by team leader.
12). requires team training.
13). Includes:
- team planning
- To Schedule
- Team review
- roles

14)
- Team improvement
- Team communication

Six Basic Principles of Software Process Change

1. **Major changes to the software process must start at the top**
   - Major changes requires leadership. Managers must provide good leadership, even though they may not do the work; they must set priorities: furnish resources and provide continuing support.
2. **Ultimately, everyone must be involved.**
   - With an immature software process, software professionals are forced to improvise solutions. in a mature process, these individual actions are more structured, efficient and reinforcing. People are the most important aspect. It's necessary to focus on repairing the process and not the people.
3. **Effective changes require the team to have common goals and knowledge of the current process.**
   - An effective change program requires a reasonable understanding of the current status. An assessment is an effective way to gain this understanding. Software professionals generally need most help in controlling requirements., coordinating changes, making plans. managing interdependencies and coping with system design issues.

4. **Change is continuous**
   - One of the most difficult things for a management team to recognize is that human interactive processes are never static. Both problems and people are in constant flux, and this fluidity all for periodic adjustment of tasks and relationships. In dealing with these dynamics, 3 points are important.
   - Relative changes generally make things worse
   -Every defect is an improvement opportunity
   -Crisis prevention is more important than crisis recovery
5. **Software process changes will not be retained without conscious effort and periodic reinforcements**
   - Precise and accurate work is hard. Its rarely sustained for long without reinforcement
   Human adoption of new process methods involves 4 stages
   - Installation, Practice , Proficiency , Naturalness
6. **Software process improvement requires investment**
   - While the need for dedicating resources to improvement seems self- surprising how often managers rely on exhorting their people to try harder