Unit - 5

CCPDS-R case study and Future software project Management practices :-

- CCPDS-R stands Command Center Processing and Display system Replacement (CCPDS-R).
- CCPDS-R project was performed for USA Air force by TRW space and defence in California.
- The entire project includes software engineering, hardware and software development.
- The schedule spanned 1987 through 1994.
- This appendix presents a detailed case study of successful software process that followed many of the techniques presented in the book.
- successful means on budget, on schedule and satisfa-ction to the customer.
- The process of CCPDS-R has two phases
    1. Concept Definition phase (CD)
    2. A Full Scale Developement phase (FSD).

1. Concept Definition phase :-

Fundamental products are

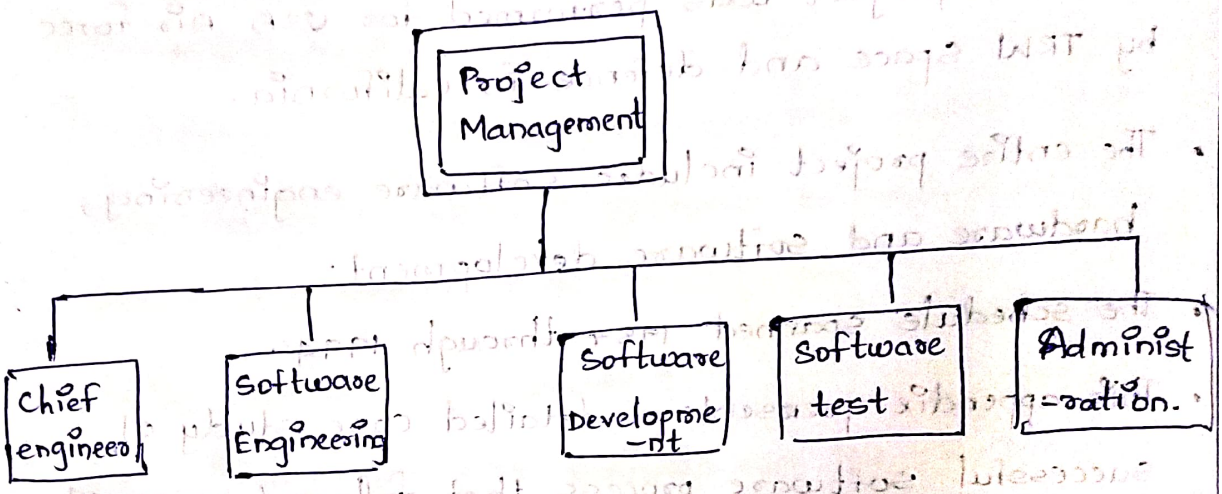
- A system specification which is planned document.
- A software development plan.
- A review of system design.
- Contract deliverable document.
- The meeting with Government stake holders.

2. Full scale development phase :-

Fundamental products are :

- A distributed software Architecture
- A flexible user interface.
- Proposed FSD software development plan.

Responsibilities :



a) Responsible for Chief Engineer :

- providing the software specification.
- Defining the stake holder interface.

b) Responsible for Software Engineering :

- Defining a software process.
- Developing a software tool.

c) Responsible for software development :

- Maintaining Components
- Developing system service components.

d) Responsible for software test :

- Building Integration testing.
- Software Development testing.

e) Responsibilities of Administration:

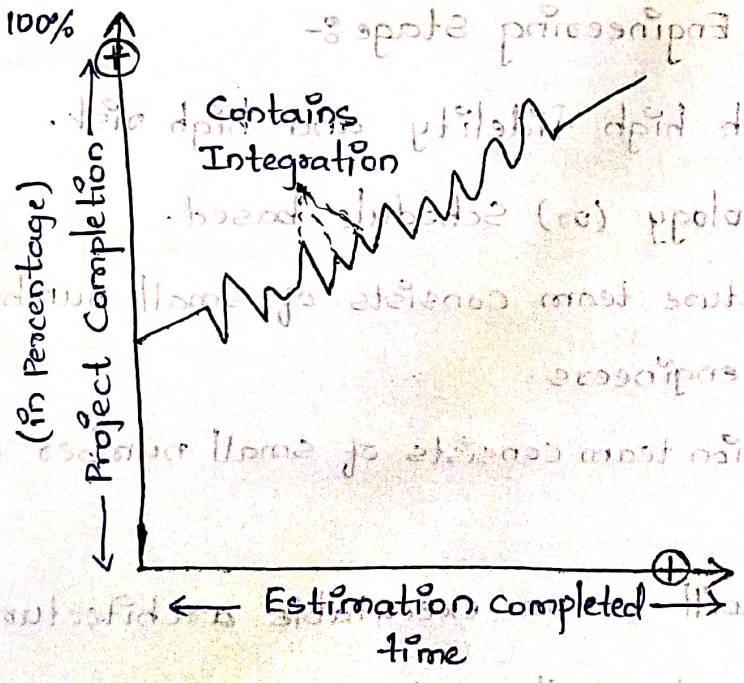
- Quality Assuring
- Cost and schedule estimating.

Responsibilities of Concept definition phase :-

- To define and generate top-level architecture:
- Organise the process and development environment.

Modern Project Profiles :-

- Modern project profile source is Unique project.
- This could-be one time project like website building.
- It typically includes an overview of project and outline of the project goals.
- It may also includes timelines, cost estimates and other relevant information.
- Modern Project profiles are more flexible.



- It produces feasible and manageable design by delaying the "design Breakage".

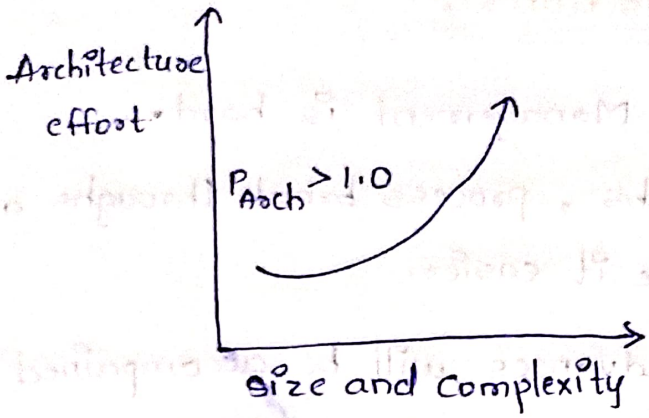
- This can be done by making use of project demonstra -tion.

Next Generation Software Economics :-

- Next Generation Software economics is being practiced by some advanced software organisation.
- It introduces several provocative hypothesis about the future of software economics.
- The general structure is proposed for cost estima -tion model.
- Some improvements will enabled by advanced in Software Development environments.
- Many of the techniques, process, models are described in the books.
- The next Generation cost model developed on the basis of Architecture as shown below.

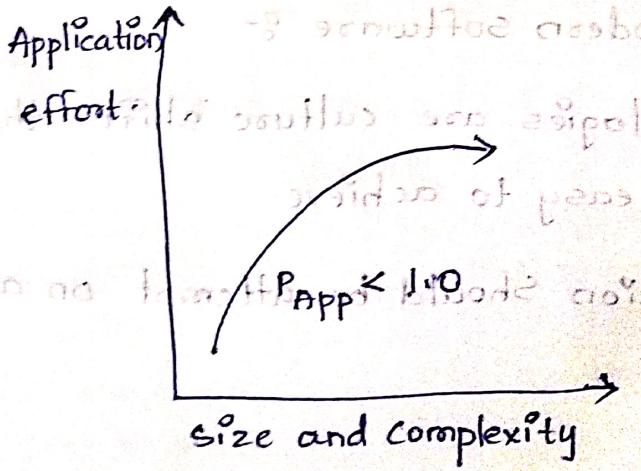
Architecture Engineering Stage :-

- A plan with high fidelity and high risk.
- It is technology (or) schedule based.
- The architecture team consists of small number of software engineers.
- The application team consists of small number of domain engineers.
- The output will be an executable architecture, production and requirements.



Application Production stage :-

- A plan with low fidelity and low risk.
- It is cost based.
- The architecture team consists of small number of software engineers.
- The Application team consists may have any no. of domains.
- The output will be function, which is deliverable and useful.



Modern Process Transitions :-

- Successful Software Management is hard.
- Technical breakthroughs, process break throughs and new tools will make it easier.
- New technological advances will be accompained by new oppurtunities for software applications, new dimensions of complexity, new avenues of automation and new customers with different priorities.
- Some of these changes will be restricted by certain stakeholders within a project. (or) Organisation.
- It is not always easy to seperate cultural resistance from objective resistance.

Key-points :-

The transition to modern software :-

- Processes and technologies are culture shifts that will not always be easy to achieve.
- A significant transition should be attempt on a significant project.