23CS712: NR23: Software Process & Project Management: 23CS712

Dr. P. Dileep Kumar Reddy

Professor-Dean-R&D,IPR, IIC

CSE Department

Narsimha Reddy Engineering College (Autonomous)

Secunderabad, Telangana State, India- 500100.

Ph.No: 09959845657

Unit 1

SOFTWARE PROCESS MATURITY

Course Objectives:

The main goal of software development projects is to create a software system with a predetermined functionality and quality in a given time frame and with given costs. For achieving this goal, models are required for determining target values and for continuously controlling these values.

This course focuses on principles, techniques, methods & tools for model-based management of software projects, assurance of product quality and process adherence (quality assurance), as well as experience-based creation & improvement of models (process management).

- The goals of the course can be characterized as follows:
- Understanding the specific roles within a software organization as related to project and process management
- Describe the principles, techniques, methods & tools for model-based management of software projects, assurance of product quality and process adherence (quality assurance), as well as experience-based creation & improvement of models (process management).

- Understanding the basic infrastructure competences (e.g., process modeling and measurement)
- Understanding the basic steps of project planning, project management, quality assurance, and process management and their relationships
- **Course Outcomes:**
- Describe and determine the purpose and importance of project management from the perspectives of planning, tracking and completion of project.

- ImplemCompare and differentiate organization structures and project structures
- ent a project to manage project schedule, expenses and resources with the application of suitable project management tools

Introduction:

- **IMPORTANT QUOTES:**
  If you don't know where you are going, any road will do." Chinese Proverb
- "If you don’t know where you are, a map won't help." Watts Humphrey
- "If you don't know where you are going, a map won't get you there any faster." Anonymous
  "You can't expect to be a functional employee in a dysfunctional environment" Watts Humphrey

- **WHY SHOULD WE MANAGE THE SOFTWARE PROCESS?**
  Individuals, Teams, and Armies:
  History of software is one of increasing scale
- Initially a few people could craft small programs
- Today large projects require the coordinated work of many teams
- The increase in scale requires a more structured approach to software process management
- People and the Software Process
- Talented people are the most important element in a software organization
- Successful organizations provide a structured and disciplined environment to do cooperative work

MYTH OF THE SUPER PROGRAMMERS:

- Common view: First-class people intuitively know how to do first-class work
  - Implication: No orderly process framework is needed
  - Conclusion: Organizations with the best people should not suffer from software quality and productivity problems
- However, studies show that companies with top graduates from leading universities are still plagued with the same problems

- **MYTH OF TOOLS AND TECHNOLOGY:**
- Common View: Some technically advanced tool or method will provide a magic answer to the software crisis
- Reality: Technology is vital, but unthinking reliance on an undefined "silver bullet" will divert attention from the need for better process management
- **MAJOR CONCERNS OF SOFTWARE PROFESSIONALS:**
- Open-ended requirements
- Uncontrolled change
- Arbitrary schedules
- Insufficient test time
- Inadequate training
- Unmanaged system standards

- **LIMITING FACTORS IN USING SOFTWARE TECHNOLOGY:**
- Poorly-defined process
- Inconsistent implementation
- Poor process management
- **FOCUSING ON SOFTWARE PROCESS MANAGEMENT:**
- Software process: the set of actions required to efficiently transform a user's need into an effective software solution
- Many software organizations have trouble defining and controlling this process

A SOFTWARE MATURITY FRAMEWORK:

- Software maturity Framework: Fundamentally, software development must be predictable. The software process is the set of tools, methods, and practices we use to produce a software product. The objectives of software process management are to produce products according to plan while simultaneously improving the organization's capability to produce better products. The basic principles are those of statistical process control

- •Effective change requires great knowledge of the current process
- •Change is continuous
- •Software process changes will not be retained without conscious effort and periodic reinforcement
- •Software process improvement requires investment.
- Continuous Change:
- •Reactive changes generally make things worse
- •Every defect is an improvement opportunity
- •Crisis prevention is more important than crisis recovery

SOFTWARE PROCESS ASSESSMENT

- Process assessments help software organizations improve themselves by identifying their
- crucial problems and establishing improvement priorities. The basic assessment objectives

- **ASSESSMENT OVERVIEW:**
- A software assessment is not an audit. Audits are conducted for senior managers who suspect problems and send in experts to uncover them. A software process assessment is a review of a software organization to advise its management and professionals on how they can improve their operation.
- The phases of assessment are:
- •Preparation - Senior management agrees to participate in the process and to take actions on the resulting recommendations or explain why not. Concludes with a training program for the assessment team

- **THE INITIAL PROCESS(LEVEL1)**
- Usually ad hoc and chaotic - Organization operates without formalized procedures, cost estimates, and project plans. Tools are neither well integrated with the process nor uniformly applied. Change control is lax, and there is little senior management exposure or understanding of the problems and issues.
- **THE REPEATABLE PROCESS (LEVEL 2)**
- This level provides control over the way the organization establishes plans and commitments. This control provides such an improvement over Level 1 that the people in the organization tend to believe they have mastered the software problem. This strength, however, stems from their prior experience in doing similar work. Level 2 organizations face major risks when presented with new challenges.

THE DEFINED PROCESS (LEVEL 3)

- The organization has the foundation for major and continuing change. When faced with a crisis, the software teams will continue to use the same process that has been defined.
- However, the process is still only qualitative; there is little data to indicate how much is accomplished or how effective the process is. There is considerable debate about the value of software process measurements and the best one to use.

THE DEFINED PROCESS (LEVEL 3)

- The organization has the foundation for major and continuing change. When faced with a crisis, the software teams will continue to use the same process that has been defined.
- However, the process is still only qualitative; there is little data to indicate how much is accomplished or how effective the process is. There is considerable debate about the value of software process measurements and the best one to use.

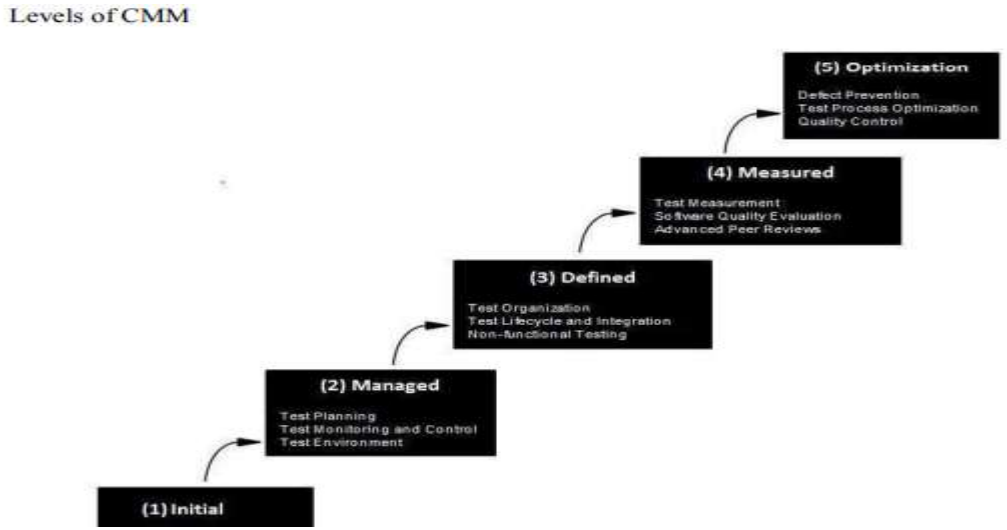
THE MANAGED PROCESS (LEVEL 4)

- Largest problem at Level 4 is the cost of gathering data. There are many sources of potentially valuable measure of the software process, but such data are expensive to collect and maintain.
- Productivity data are meaningless unless explicitly defined. For example, the simple measure of lines of source code per expended development month can vary by 100 times or more,

THE OPTIMIZING PROCESS (LEVEL 5)

- To this point software development managers have largely focused on their products and will typically gather and analyze only data that directly relates to product improvement. In the Optimizing Process, the data are available to tune the process itself.

- **CAPABILITY MATURITY MODEL (CMM):**



- **WHAT IS CMMI ?**
- CMM Integration project was formed to sort out the problem of using multiple CMMs.
- CMMI Product Team's mission was to combine three Source Models into a single
- improvement framework to be used by the organizations pursuing enterprise-wide process
- improvement. These three Source Models are :
  - Capability Maturity Model for Software (SW-CMM) - v2.0 Draft C
  - Electronic Industries Alliance Interim Standard (EIA/IS) - 731 Systems Engineering
  - Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98

PSP

- The Personal Software Process (PSP) is a structured software development process that is designed to help software engineers better understand and improve their performance by bringing discipline to the way they develop software and tracking their predicted and actual development of the code.

TSP

- The team software process (TSP) provides a defined operational process framework that is designed to help teams of managers and engineers organize projects and produce software the principles products that range in size from small projects of several thousand lines of code (KLOC) to very large projects greater than half a million lines of code

Thank You..