Software Process Improvement (3C05/D22)

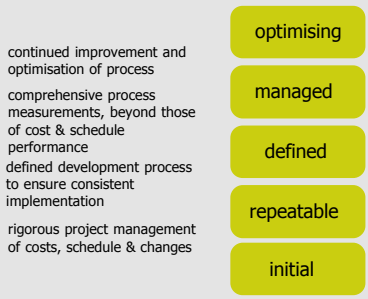
Unit 9: Software Process Improvement

- Objectives
  - To provide a framework for software process assessment and improvement
  - To introduce the Capability Maturity Model (CMM) of the Software Engineering Institute

Background

- A framework to help the US DoD pick software vendors more cleverly and hence obtain a quick and relatively inexpensive "productivity boost".
- Work at the CMU Software Engineering Institute, 1986 onwards.
  - Humphrey, W.S. (1988); Characterizing the Software Process: a maturity framework; IEEE Software; 5, 2, pp 73-79.
  - Humphrey W.S.; Kitson D.H. & Kasse T.C. (1989); The State of Software Engineering Practice: a preliminary report; Proc. IEEE 11th International Conference on Software Engineering; pp 277-288, IEEE CS Press.

5 Level Process Maturity Framework



Approach

- An effective software process is predictable, cost estimates and schedule commitments are met with reasonable consistency. The resulting products generally meet user needs.
- The key requirements for this to be achieved are:
  - measurement
  - statistical control

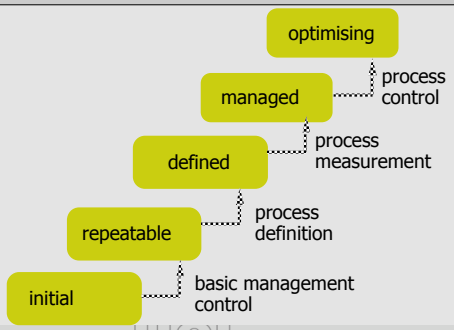
Development Process Improvement

- FIVE STEPS
  (1) understand the current status of their development process or processes;
  (2) develop a vision of the desired process;
  (3) establish a list of required process improvement actions in order of priority;
  (4) produce a plan to accomplish these actions;
  (5) commit the resources to execute the plan.

Levels

- Levels are chosen because:
  - reasonably represent historical phases of evolutionary improvement of real software development organisations;
  - represent a measure of improvement that it is reasonable to achieve from the prior level;
  - suggest interim improvement goals and progress measures;
  - makes obvious a set of immediate improvement priorities.

How to Move Between Levels



Initial to Repeatable

- To get from initial to repeatable, you need:
  - project management;
  - management oversight;
  - quality assurance;
  - change control.
- You need "commitment control".

Repeatable to Defined

- To get from repeatable to defined, you need:
  - to establish a "process group";
  - a software development process architecture;
  - to introduce a family of software engineering methods and technologies.

UCL ComputerScience

Defined to Managed

- To get from defined to managed, you need:
  - to establish a minimum, basic set of process measurements;
  - to establish a process database with the resources to manage and maintain it;
  - to provide sufficient process resources to gather and maintain this data;
  - assess the relative quality of each product and inform management where quality targets are not being met.

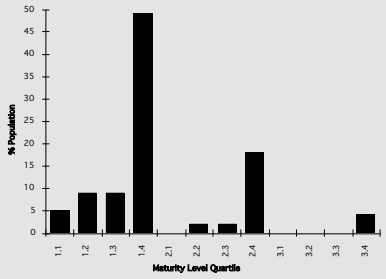
UCL ComputerScience

Managed to Optimising

- To get from managed to optimising, you need:
  - to support automatic gathering of process data;
  - to use this data to both analyse and modify the process to prevent problems and improve efficiency.

UCL ComputerScience

Results of SEI Assisted Assessment



Key Questions for Level 2

- Does the Software Quality Assurance (SQA) function have a management reporting channel separate from the software development project management?
- Is there a software configuration control function for each project that involves software development?
- Is a formal procedure used in the management review of each software development prior to making contractual commitments?
- Is a formal procedure used to make estimates of software size?

Key Questions for Level 2

- Is a formal procedure used to produce software development schedules?
- Are formal procedures applied to estimating software development cost?
- Are profiles of software size maintained for each software configuration item, over time?
- Are statistics on software code and test errors gathered?
- Does senior management have a mechanism for the regular review of the status of software development projects?

Key Questions for Level 3

- Is there a software engineering process group function?
- Is there a required software engineering training programme for software developers?
- Is a formal training programme required for design and code review leaders?
- Does the software development organisation use a standardised software development process?

UCL ComputerScience

Key Questions for Level 3

- Does the software development organisation use a standardised and documented software development process on each project?
- Are statistics on software design errors gathered?
- Are internal software design reviews conducted?
- Is a mechanism used for controlling changes to the software design?
- Are software code reviews conducted?

UCL ComputerScience

Key Questions for Level 3

- Are the action items resulting from design reviews tracked to closure?
- Are the action items resulting from code reviews tracked to closure?
- Is a mechanism for used for ensuring compliance with software engineering standards?
- Is a mechanism used for verifying that the samples examined by Software Quality Assurance are truly representative of the work performed?

UCL ComputerScience

LEVEL	CHARACTERISTIC	KEY PROBLEM AREAS
optimising	improvement fed back into process	automation
managed	quantitative measured process	changing technology problem analysis problem prevention
defined	process defined & institutionalised	process measurement process analysis quantitative quality plans
repeatable	process dependent on individuals	training technical practices process focus
initial	ad hoc/chaotic	project management project planning configuration management software quality assurance

Key Points

- In order to improve the software process in an organisation you must first understand it. Your improvements must be appropriate to the maturity level you discover. You cannot have a technical fix to fundamental management problems. You cannot skip a stage, any attempt to move directly from 1 to 5 is doomed to failure.