## CH 1 Software Maturity Framework

- **Company SW problems:** Open-ended requirements, Uncontrolled change, Arbitrary schedules, Insufficient testing time, Inadequate training, Unmanaged system standards
- Effective use of software technology is limited by ill-defined process, inconsistent implementation, & poor process management.
- SEI Software Capability Maturity Model (CMM) is useful for evaluating strengths & weaknesses, which has proved valuable for assessing other software organizations.
- **Software process:** set of tools, methods, & practices we use to produce a software product. **Objectives of software process management:** produce products according to plan while simultaneously improving the organization's capability to produce better products. **Characteristics of truly effective software process:** Predictable [cost estimates, schedule commitments met with reasonable consistency], resulting products should meet users' functional & quality expectations.
- **Software Maturity Framework:** statistical process control [process is under statistical control if its future performance is predictable within established statistical limits]
- Basic principle behind statistical control is measurement **Factors for measuring the process:** Numbers must properly represent the process being controlled, numbers must be sufficiently well defined and verified to provide a reliable basis for action, measuring human processes changes them.
- To address problems in the software process, treat the entire software task as a process that can be controlled, measured, and improved.
- **Improvement of software development organizations:** 1. Understand current status of development process. 2. Develop vision of desired process. 3. Establish list of required process improvement actions in order of priority. 4. Produce a plan to accomplish required actions. 5. Commit resources to execute the plan. 6. Start over at step 1.
- **Process maturity levels:** represent 1. actual historical phases of evolutionary improvement of real SW organizations 2. measure of improvement that is reasonable to achieve from the prior level 3. Suggest interim improvement goals and progress measures 4. Make obvious a set of immediate improvement priorities.
- **Initial Process (Level 1):** Ad hoc, chaotic, org. operates without formalized procedures, cost estimates, and project plans. Tools not integrated or uniformly applied. Change control-lax, little senior management exposure or understanding of problems. No management mechanism to ensure formal procedures for planning & tracking used. Test to observe- how org. behaves in crisis. **Performance improvement:** *PROJECT MANAGEMENT* * Ensure effective control of commitments * Starts with understanding of job's magnitude. *MANAGEMENT OVERSIGHT* * Review & approval of major development plans prior to official commitment. *QUALITY ASSURANCE* * assuring management SW work being done the way it's supposed to be done. * Effective- independent reporting line to senior management and sufficient resources to monitor performance * 3-6% of SW org. *CHANGE CONTROL* * requirements must be established and maintained with reasonable stability * changes in req., design, code etc. must be controlled.
- **Repeatable Process (Level 2):** Control over the way the organization establishes its plans and commitment. Face major risks when they are presented with new challenges. **Performance improvement:** *ESTABLISH PROCESS GROUP* * technical resource that focuses exclusively on improving SW process. *ESTABLISH SW DEVELOPMENT PROCESS ARCHITECTURE* * structural decomposition of the development life cycle into tasks is a, each of which has a defined set of prerequisites, functional descriptions, verification procedures, and task completion specifications. * technical & management activities for proper execution. *INTRODUCE FAMILY OF SE METHODS/TECH* * design and code inspections, formal design methods, library control systems, and comprehensive testing methods, prototyping
- **Defined process (Level 3):** achieved the foundations for major and continuing progress **Performance Improvement:** *Establish minimum basic set of process measurements to identify quality and cost parameters of each process step. *Establish a process database and resources to manage and maintain it. *Provide sufficient process resources to gather and maintain this process data and to advise project members on its use. *Assess relative quality of each product and inform management where quality targets are not being met.
- **Managed Process (Level 4):** org. should expect to make substantial quality improvement, problem: cost of gathering data **Performance improvement:** *automatic gathering of process data *process data both to analyse and to modify the process
- **Optimizing Process (Level 5):** *data is available to tune process itself * process optimization major quality, productivity benefits * identify weakest elements of process, fix them

## CH 2 The Principles of Software Process Change

- **Process:** assessing SW problems, + people + design
- **People:** *Better people, better work * best people short supply * probably have best team right now * proper support & leadership better performance
- **Design:** * superior product -superior design *successes designed by people who understood application * program – executable knowledge *application knowledge * orderly process
- **Principles of Software Process Change:** * Major changes to SW process must start at top. [Senior management launch change effort, provide resources] * Everyone must be involved. [team effort, non-participants miss benefits, inhibit progress] *Effective change requires a goal & knowledge of current process. [understand current status, conduct assessment] * Change is continuous. [Reactive changes generally make things worse. Every defect- improvement opportunity. Crisis prevention >> crisis recovery] *SW process changes will not be retained without conscious effort and periodic reinforcement. [Four stages: • Installation involves initial installation of the method, training in their proper use • Practice people learn to perform the methods as instructed • Proficiency With traditional learning curve, people gradually improve their efficiency. • Naturalness methods so ingrained they are performed without intellectual effort.] *SW process improvement requires investment [planning, dedicated people, management time, and capital investment]
- To Improve the SW Process, Someone Must Work on It. Unplanned Process Improvement Is Wishful Thinking. Automation of a Poorly Defined Process Will Produce Poorly Defined Results. Improvements Should Be Made in Small Steps. Train.
- **Misconceptions about the Software Process:** We must start with firm requirements. [req. change as job progresses, gradually increase the level of detail] *If it passes test, it must be OK. *Software quality cannot be measured. *The problems are technical. *We need better people. [true cause of errors is the process, not the people] *Software management is different.
- **Implementing software process change:** effective change process has three phases: – Unfreezing – Moving– Refreezing * unfreezing is best initiated by an effort to understand the true problems in the organization and to make people aware of the opportunities for change. *Assessment
- **Champions** are the ones who initiate the change process. **Senior management role**: someone in authority needs to recognize the value of the work and sponsor it. Provide both resources and official backing. **Change agents** lead change planning and implementation.
- Three key elements of effective change are • Planning• Implementation • Communication [maintain a continuous stream of actions and successes] [Public plans, Periodic progress reports, Early demonstration of success- help to ensure continued executive support.]
- Refreezing: ensure that an achieved capability is retained in general practice [Retain the management team that instituted the change • Modify the organization's procedure • Establish measurements and incentives • Set up a dedicated staff to monitor and support performance • Establish an education and training program]

## CHr 3:Software Process

**Assessment:Assessment Overview**: Process assessment helps software organisations improve themselves by identifying their critical problems and establishing improvement priorities.Objective:how the organisation works ,to identify its major problems,To enrol its opinion leaders in the change process **Assessment Phases**:**Preparation**:Senior management becomes committed to the process, agrees to participate personally, and commits to take action on the resulting recommendations,brief one-to-one training for the team. **Assessment**: takes upto 2 or more weeks and concludes with a preliminary report of the findings.**Recommendation**: the findings and action recommendations are presented to the local managers.local action team is assembled to plan and implement **Five Assessment Principles:The need for a process model as a basis for the assessment:** reviewed in comparison with some vision of how the process should be performed,**The requirement for confidentiality:**support the organisation's improvement program and not to report its problems to higher management. If such data is discovered, they cannot speak in confidence and support the organisation's improvement program and not report its problems to higher management,**Senior management involvement:** sets the priorities and must be personally involved in the assessment,**An attitude of respect for the views of the people in the organisation being assessed:**In Spite of supportive team,people will be resentful and not cooperate,**An action orientation:** directed towards improvement.either focus on taking action or not do an assessment.**The Assessment Process:**Forming an Assessment Team,Guidelines for Selecting Assessment Team Members, Self-Assessment Considerations,Assessment Ground Rules, Assessment Team Training, The On-Site Period,**Assessment Conduct:**what the projects actually do, how they do it, the problems encountered, and the results obtained.Meetings with expertise in various facets of S/W process,**Probing Questions:** hard to obtain really accurate information,**Assessment Conclusions:** the team prepares a report on its initial finding,a composite summary of site status and review it with the project managers,**Assessment Conclusions:** review the composite findings with the site manager and staff, **Assessment Report:** presentation of a written final report and recommendations, **Action Plans**,**Reassessments**,**Implementation Considerations:**assessment risk is that no significant improvement actions will be taken.Long term goals and then sub goals are decided.**Risk:Schedule Conflicts,Inadequate support,lack of follow-through.**

## CHr 4: The Initial Process:

**Levels: Initial**: Chaotic-unpredictable cost, schedule, and quality performance,**Repeatable**:Intuitive-– cost and quality highly variable, reasonable control of schedules,informal and ad hoc process methods and procedures,**Defined:**Qualitative – reliable costs and schedules, improving but unpredictable quality performance,**Managed:**Quantitative – reasonable statistical control over product quality,**Optimising-** Quantitative basis for continued capital investment in process automation and improvement.**Needed actions:Initial:**Planning performance tracking, change control, commitment management, Quality Assurance,**Repeatable:**Develop process standards and definitions, assign process resources, establish methods,**Defined:**Establish process measurements and quantitative quality goals, plans, measurements, and tracking,**Managed:**Quantitative productivity plans and tracking, instrumented process environment, economically justified technology investments,**Optimising-**Continued emphasis on process measurement and process methods for error prevention.

**Chaotic Project:** inadequate project management,Without a plan destroys technical work.Chaotic project is seriously late,The schedule does not match actual,No plan even with a good technical shape–Lack of documented planning– A small amount of additional function expected turns out to be hundred thousands of code.– No idea how big the project really is – No basis to know where they stand– No way to justify added resources when they get into trouble– Even though the technical work would be sound, the customer was convinced the project would be a disaster. **Why Software Organizations Are Chaotic:** often make a guess rather than a plan,temptation to believe in magic,project follows an escalating cycle,new technical challenges put pressure on the process. **Chaotic Forces:Unplanned Commitments:**insufficient time and resources,hidden trap,**Gurus,Magic:**need effective management,**Problems of Scale:** software products become larger, they are much more difficult to understand and prototypes or multiple releases are needed because of those new complications.– The total function cannot be implemented in time.– Some needs cannot be understood without operational experience on a partial system.– Some design issues cannot be resolved until a preliminary system has been built and run.– A release discipline helps sort out user priorities.– No one successfully builds large software systems in one shot anyway.**The Implications of Software Scale:a**ffect the individual, the management system, and the technical methods and tools that we use,**Software Process Entropy:** many forces on the software process that push us toward disorganisation i.e.Dynamic requirements,Increasing system size ,Human nature.**The Way Out:** Plan the work.• Track and maintain the plan.• Divide the work into independent parts.• Precisely define the requirements for each part.• Rigorously control the relationships among the parts.• Treat software development as a learning process.• Recognize what you don’t know• When the gap between your knowledge and the task is severe, fix it before proceeding.• Manage, audit, and review the work to ensure it is done as planned.• Commit to your work and work to meet your commitments.• Refine the plan as your knowledge of the job improves.

CH 5: Managing Software Organizations

- **Role:** successfully completed, organization-wide agreement, requires continuing management focus,
- **Basic principles:** plan based on commitments, commitment is by one person to do something for other, resolves conflict between line and staff, Oversights, and review system tracks progress against the plan
- **Commitment Discipline:** supported by plans, estimates, reviews, met by committed by people
- **Making a Commitment:** a planned completion date and some consideration or payment, when the coordinated efforts then mutual commitments are essential
- **elements of an effective commitment:** willingly, not lightly, agreement on what has to be done, by whom, when, publicly stated, advance notice if commitment not met
- **The Commitment Hierarchy:** done by committed individual not done in vacuum, a management team making commitments,
- **The Software Commitment Process:** top of the organization, made by senior exec, made only after formal review, enforcement mechanism to review and concurrence is maintained
- **Establishing a Commitment Process:** required planning before commitment, training courses, Estimating, review and approval procedure
- **The Management System Goals:** technical and business strategy, provide quality products, provide quality products, improve continually
- **Product and Period Plans:** To resolve the inherent conflict -> annual operating plans, specify the task, skilled resource allocation,
- **Management Oversight:** reviews and a contention system to resolve conflicts, each staff and line annual plan and reviews, separate plan is consolidated into org plan
- **The Contention Process:** requires a parallel contention system to encourage openness, best decision based on full understanding,
- **The Principles of the Contention System:** Reviewed and agreed by all parties, all parties present for decision, when no disagreement SM decides if knowledgeable agreement is there, if any decision is deferred,
- **The Quarterly Review:** forum for resolving conflicts and monitoring progress, assessment of progress on plan, and against goals,
- **Project Phase Reviews:** assess project progress periodically, review at key points, DoD standards MIL-STD-2167, MIL-STD-498, ISO/IEC 12207, ISO/IEC 15504(spice), ISO/IEC 15288, ISO 9000 Serie
- **Establishing a Project Management System:** if **project** management is not aware of then system can’t be effective.

CH 6: The Project Plan

- **The Project Plan**: The Work and how it’s done, benchmark to compare with actual performance.
- **Project Planning Principles:** quality program, precise of user needs, conceptual design, subsequent requirements, detailed design and implementation
  Strategy, implementation details, framework for negotiating the time and resources
- **Planning Considerations:** initial resource estimates unexpectable, early agreement on the essential functions, harmonious teamwork between the users and the software engineers
- **The Planning Cycle:** initial requirements, work Breakdown Structure (WBS), Product size estimates, Resource estimates, Project schedule
- **Goals and Objectives:** negotiation period, **WBS:** estimate of size of the product, Size **Measures:** Early estimates to actual, **Function points** from users’ perspective determines the number and complexity of inputs, summed to produce an overall function point rating,
- **Estimating:** Wideband Delphi Technique: - Group of experts given estimate forms, meet and anonymously complete the form, any method can be inaccurate,
- **Estimating Inaccuracies:** Normal statistical variations, estimating bias caused by the project stage,
- **Estimating Contingencies:** inflate the resource estimates, add to the schedule, and increase
- **Productivity Factors:** number of programmer months and time required, orderly procedure, – Organization productivity data, developing productivity data, Using productivity data, Resource estimating
- **Developing Productivity Data:** Identify a number of recently completed programs that are as similar, get data on the size, in LOC, count of the programmer months
- **Using Productivity Data:** derive base productivity numbers,
- **Scheduling:** total resource needs, organization’s historical experience, the overall project schedule objective, a staffing plan is developed, preliminary schedule, preliminary plan,
- **Project Tracking:** ability to determine project status, checkpoints, earned-value project scheduling,
- **The Development Plan:** submitted to management for review and approval, Project purpose and scope, Project goals and objectives, Organization and responsibilities, Management and technical controls, Work definition and flow, Development environment, Software development methodology, Configuration management, Verification and validation, Quality assurance provisions
- **Planning Models:** COCOMO, SLIM, error by 500 percent or more, used to augment, used effectively to check the estimates for errors or oversights,
- Prereq to a SW cost estimate the establishment of estimating group, rarely improve their estimating accuracy to this degree w/O the support of an orderly planning process

AGILE

• Iterative development:an approach to building software in iterations and Each iteration is a Self-contained mini-project composed of activities such as requirements analysis, design, programming, and test.The goal for the end of an iteration is an iteration release, a stable, integrated and tested partially complete system.• Risk-driven and client-driven : riskiest, most difficult elements,client – whatever they perceive as the highest business value • Time-boxing:Fixed time for each iteration.• Evolutionary development:the requirements, plan, estimates, and solution evolve or are refined rather than fully defined and “frozen” in a major up-front specification effort. • Adaptive planning:elements adapt in response to feedback. • Incremental delivery:repeatedly delivering a system into production (or the marketplace) in a series of expanding capabilities.• Evolutionary delivery:refinement of the practice of incremental delivery in which there is a vigorous attempt to capture feedback regarding the installed product, and use this to guide the next delivery. **The Agile Manifesto:**• Individuals and interactions over processes and tools• Working software over comprehensive documentation• Customer collaboration over contract negotiation• Responding to change over following a plan.**Principles behind the Agile Manifesto:** satisfy the customer,Welcome changing requirements,Deliver working software frequently,Business people and developers must work together daily,Build projects around motivated individuals,face-to-face conversation,Working software,promote sustainable development,Continuous attention to technical excellence and good design enhances agility,Simplicity,he best architectures, requirements, and designs emerge from self-organizing teams.