LECTURE 1

What is a project?

=> A project is a temporary endeavor undertaken to create a unique product, service, or result.

Temporary

- ! a definite beginning
- ! a definite end
- " project's objectives have been achieved " it becomes clear that the project objectives will not or cannot be met" the need for the project no longer exists and the project is terminated
- ! Projects are not ongoing efforts!!

Unique Products, Services, or Results

- ! A product is quantifiable, and can be either an end item in itself or a component
- Item
- ! A capability to perform a service, such as business functions supporting production
- or distribution
- ! A result, such as outcomes or documents.

Projects vs. Operational Work

! Common features

- " Performed by people
- " Constrained by limited resources
- " Planned, executed, and controlled

! Different features

- " operations are ongoing and repetitive, while projects are temporary and unique

Software Development Projects

! Software development life cycle

- " Full life cycle projects
- " Partial life cycle projects

! Approach driven SDP

- " Fresh development (from scratch)
- " COTS product customization/implementation
- " Porting
- " Integration

! Maintenance projects

- " Defect repair
- " Functional expansion
- " Operational support
- " Software modification

SE processes vs. PM processes

SE processes

- ! Requirements
- ! Analysis&Design
- ! Construction
- ! Testing

PM processes

- ! Acquisition
- ! Initiation
- ! Execution
- ! Closure

+

- ! Measurement &
- Analysis
- ! Training
- ! Knowledge repository

<u>Projects and Strategic Planning</u>

- ! A market demand (e.g., an oil company authorizes a project to build a new refinery in
- response to chronic gasoline shortages)
- ! An organizational need (e.g., a training company authorizes a project to create a new

- course in order to increase its revenues)
- ! A customer request (e.g., an electric utility authorizes a project to build a new substation to
- serve a new industrial park)
- ! A technological advance (e.g., a software firm authorizes a new project to develop a
- new generation of video games after the introduction of new gameplaying devices)
- ! A legal requirement (e.g., a paint manufacturer authorizes a project to
- establish guidelines for the handling of a new toxic material).

What is Project Management

! Project management is the application of knowledge, skills, tools and techniques to project activities to meet project requirements.

General Management Knowledge and Skills

- ! Financial management and accounting
- ! Purchasing and procurement
- ! Sales and marketing
- ! Contracts and commercial law
- ! Manufacturing and distribution
- ! Logistics and supply chain
- ! Strategic planning, tactical planning, and operational planning
- ! Organizational structures, organizational behavior, personnel
- administration, compensation, benefits, and career paths
- ! Health and safety practices
- ! Information technology.

Interpersonal Skills

- ! Effective communication.
- ! Influencing the organization.
- ! Leadership.
- ! Motivation.
- ! Negotiation and conflict management.
- ! Problem solving.

LECTURE 2

Project Management Dimensions
! Technical
" Planning
" Monitoring and Control
! People
" Motivation
" Leadership

What is a Process? -
! A process is “a system of operations introducing something ... a series of actions, changes, or functions that achieve an end or result”. [Webster's dictionary]
•While process is often described as a leg of the process - people – technology triad, it may also be considered the “glue” that unifies the other aspects.
A process is a set of practices performed to achieve a given purpose; it may include tools, methods, materials, and/or people.
! Characteristics of process:

- " Prescribes all the major activities.
- " Possesses the set of leading principles that define
- the objectives of the individual activities.
- " The activities are organized into sequences.
- " Every activity possesses the starting and ending
- criteria.
- " Every activity uses resources in accordance with
- stated constraints and results with intermediate or final
- deliverables (products).
- " Activities may be structured.

Why Focus on Process?

- ! **Process provides a constructive, high-leverage focus as opposed to a focus on people**
- ! Your work force, on the average, is as “good” as it is trained to be.
- ! Working harder is not the answer.
- **! Working smarter, through process, is the answer. as opposed to a focus on technology**
- ! Technology applied without a suitable roadmap will not result in significant payoff.
- ! Technology provides the most benefit in the context of an appropriate process roadmap.

MATURE/IMATURE PROCESS

understood, used, and improving ! Is controlled and compliance is monitored ! Collects and uses measurement	Is not written down ! Is not enforced ! Is not followed ! Dependent on individuals ! Does not control requirements or products

MATURE/IMATURE ORGANIZATIONS:

IMATURE	MATURE
Performance driven by the competence and heroics of the people doing the work ! Schedule drives everything despite requirements ! Often the schedules unrealistic ! Testing unpredictable and often shortened ! Development starts before requirements are stable ! Unpredictable performance ! Few to no objective measures	Schedules negotiated on the basis of agreed to requirements ! Less dependent on heroics ! Measurement drives decisions ! Risks are identified and managed ! In times of crisis, relies more on process ! Predictable performance ! Early defect identification and removal ! Requirements are documented and managed

CONCEPTS OF PROCESS MATURITY

Software Process: a set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products.

Levels of process

- ! Measure the process level against a set of best practices resulted from industry experience
- ! The recognized metric is CMMI (Capability Maturity Model Integration)

CMMI Overview

! What is CMMI?

- A set of practices covering:
- " Project Management
- ! Planning, monitoring and control, risks, suppliers
- " Engineering
- ! Requirements development, technical solution, product integration, verification, validation Support
- ! Process and product quality assurance, configuration management Process Management
- ! Organizations process focus, definition, training Describes What not How
- ! Is an organized structured set of requirements for implementing industry proven practices
- ! Aids progress measurement by providing a "target"
- ! Provides a way to benchmark the maturity of your process and compare to other organizations
- ! Establishes a common vision and language for an organization

Why CMMI?

- ! CMMI provides guidance for improving organization's processes and your ability to manage the development, acquisition, and maintenance of products or services
- ! Consistently provide the right software and system, on time and on cost
- ! Gain visibility into and control of process

- ! Improve
    - " Discipline
    - " Software and systems engineering process capability
    - " Product quality
    - " Predictability/estimating capability
- ! CMMI places proven approaches into a structure that:
    - " helps the organization examine the effectiveness of your processes
    - " establishes priorities for improvement
    - " help to implement these improvements

Improvements in CMMI

- ! Incorporates additional years of learning
- ! More explicitly links best practices to business objectives
- ! Expands the scope of and visibility into the product life cycle and engineering activities
- ! Adds more best practices (e.g.: Measurement, Risk Management, Product Integration, Decision Analysis and Resolution and Supplier Management)
- ! Captures more robust high-maturity practices
- ! Addresses additional generic practices needed for institutionalization
- ! Fully complies with relevant ISO standards
- ! It is a model

What CMMI is NOT?

- ! A cookbook
- ! A complete set of practices
- ! A set of processes/procedures
- ! A set of “bolt-on processes” that last only as long as the wheel is squeaking. CMMI provides a consistent, enduring framework that accommodates new initiatives.
- ! Restricted to a single discipline (I.e. Software Engineering or Systems Engineering). CMMI focuses on
- the total software intensive system problem and integrates multiple disciplines into one process improvement framework that eliminates inconsistencies and reduces duplication

Methodologies compliance with CMMI - Gap Analysis

- ! Methodologies measure with OOSE process compliance employing gap analysis
- ! Developing a gap analysis is a long and tedious effort
- ! A gap analysis example is provided by MS in the MSF for CMMI Process improvement

Process approach vs. PM

- ! Ad-hoc
    - " Not documented and depends on involved parties
    - " PM has absolute control
- ! Process–driven
    - " Documented processes on all activities
    - " Organizational infrastructure enables successful execution of projects

Ad-hoc approach

- ! Dynamic environment
- ! Leader has absolute control
- ! Fast response to environmental changes
- ! Can be profitable
- ! Perfect for pinning the blame on one person
- ! Reduces process overhead
- ! Creates uncertainty
- ! Centralizes authority
- ! Unpredictable results
- ! Focuses on people monitoring
- ! Organizational bandwidth depends on leaders

Works good for…

- ! Small organization
- ! Small number of PMs in organization (i.e.2-3)
- ! Small number of concurrent projects (< 5)

Process-driven approach

- ! Minimizes person dependency
- ! Enables good performance for beginners
- ! Facilitates integration of project experience into process
- ! Monitors projects rather than people
- ! Involves the organization
- ! Provides uniformity
- ! Facilitates measurement
- ! Builds basis for predictability

Right choice?

- ! Financial processes (enforced by law)
- ! HR processes (enforced by law, fairness)
- ! Project management (no fairness/law enforcement!)

Software Development Methodologies
! Plan-driven
! Agile

Plan based - UP Principles

- ! Most prominent example is RUP
- ! Essentials of the UP (© RUP)
    - " Develop a Vision
    - " Manage to the Plan
    - " Mitigate Risks and Track Related Issues
    - " Examine the Business Case
    - " Design a Component Architecture
    - " Incrementally Build and Test the Product
    - " Regularly Assess Results
    - " Manage and Control Changes

- " Deploy a Usable Product
- " Adopt a Process that Fits Your Project

Project Management Definition (according to RUP)
! "Software Project Management is the art of balancing competing objectives, managing risk, and overcoming constraints to deliver, successfully, a product which meets the needs of both customers (the payers of bills) and the users. The fact that so few projects are unarguably successful is comment enough on the difficulty of the task."

The Unified Process!!!!

AGILE!!

WHY DIFFERENT?

- ! Common features
  - " Commitment to quality
  - " Communication
  - " Iterative
- ! Difference
  - " Time-span for planning

SCRUM PM Definition
! "The ScrumMaster acts as a liaison between the Product Owner and the team. The ScrumMaster does not manage the team. Instead, he or she works to remove any impediments that are obstructing the team from achieving its sprint goals. In short, this role helps the team remain creative and productive, while making sure its successes are visible to the Product Owner [leadership skills]. The ScrumMaster also works to advise the Product Owner about how to maximize ROI for the team [monitoring and control]."

Agile Values

- ! People and interactions are more important that development tools
- ! Working software is more important that complete documentation
- ! Collaboration with clients is more important that contract negotiation
- ! Response time to problems is more important than following a plan

Agile Methodologies Principles (1) –
http://www.agilemanifesto.org/principles.html

- ! An agile, scalable approach to requirements management, development and system testing
- ! Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- ! Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

- ! Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- ! Business people and developers must work together daily throughout the project.
- ! Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- ! The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- ! Working software is the primary measure of progress.
- ! Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- ! Continuous attention to technical excellence and good design enhances agility.
- ! Simplicity--the art of maximizing the amount of work not done—is essential.
- ! The best architectures, requirements, and designs emerge from self-organizing teams.
- ! At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

EssUp – The Essential Unified Process
! Tries to extract the essentials from the development process and let the people learn for themselves
! Relies on the Separation of Concerns idea, or aspect oriented thinking – you address specific concerns in the order of their priorities
EssUp foundation (1)

- ! Iterative essentials
  - " Develop incrementally in value and iterative to minimize risks
- ! Architecture essentials
  - " Create a firm foundation for the development of a robust, high-quality system
- ! Use Case essentials
  - " Employ an agile approach for Requirements Management, Development and Testing
  - " Welcome (controlled) change
- ! Component essentials
  - " Employ reuse, maintain a componentized system
- ! Model essentials
  - " Model visually, use models for communication
- ! Product essentials
  - " Manage releases, continuous integration
- ! Process essentials
  - " Keep processes simple, assign clear roles (Responsible, Accountable, Consulted, Informed)
- ! Team essentials
  - " Maintain the environment that enables the team to excel

MSF for Agile Development Principles

- ! Partner with customer
- ! Work toward a shared vision
- ! Deliver incremental value
- ! Invest in quality
- ! Empower team members
- ! Establish clear accountability

- ! Learn from all experiences
- ! Foster open communication
- ! Stay agile, adapt to change

MSF for CMMI Principles

- ! Partner with customers
- ! Foster open communication
- ! Work towards a shared vision
- ! Quality is Everyone’s Business, Every
- Day
- ! Stay Agile, adapt to change
- ! Make deployment a habit
- ! Flow of value

LECTURE 3

What is a project

- ! A unique venture with specific timeframe
- ! Conducted by people, often from various backgrounds
- ! Goal oriented, clear target
- ! Constrained by cost, schedule, resources, functionality and quality
- ! Divisible into phases and iterations
- ! Has a clear priority in the organization

PROJECT MANAGEMNT SKILLS

- TECHNIQUE
  - ! Planning the project
  - ! Implementation of the project plan
  - ! Tracking, reporting, and control of projects as they progress
  - ! Project closure and completion of the project
- TOOLS
  - PLANNING
  - Tracking
  - Reporting
- PEOPLE MANAGEMENT
  - Manage people who implement the project
  - Manage stakeholders

Project Management Processes (PMBOK)

- ! Scope Management
- ! Integration
- ! Time
- ! Cost
- ! Quality
- ! Human Resources
- ! Communication
- ! Risk
- ! Procurement

PM process types

- ! Project Management discipline perspective
  - " Core processes
  - " Support processes
- ! Core Project Management perspective
  - " Initiation Processes
  - " Planning Processes
  - " Executing Processes
  - " Controlling Processes
  - " Closing Processes

Initiation
Problem statement
Project mission
Project Objectives – SMART

- **Specific**
- **! Measurable**
- **! Attainable**
- **! Relevant**
- **! Time-bound**

Planning – schedule, budget, workers, strategy (WBS, admin strategy)

WBS

- **! Identifying the work to do**
- **! Laying out the project's schedule**
- **! Building a project team**
- **! Setting the project's budget**

COMMUNICATION MANAGEMENT

CHANGE MANAGEMENT – record changes, track changes, evaluate changes effect on the project, update documentation

RISK MANAGEMENT:

- Identifying risks
- Assessing risks
    - Risk value = f(impact, probability)
- Planning risk response
    - " Accept
    - " Avoid
    - " Control (mitigate)
    - " Transfer
    - " Plan for alternatives (contingency)

- Tracking risks – keeping logs

RUP DEVELOPMENT!!

EXAMPLE SLIDE 120-144!!!

AGILE
SCRUM
DISTRIBUTED AGILE DEVELOPMENT

If the project is small -> tailor the process and the development
LECTURE 4
PLANNING AND TAILORING THE PROCESS

-waterfall model
-Unified Process – iterative, incremental
-Agile model – SCRUM
-Waterfall – UP –AGILE
-Tailoring guidelines

- !Size. The number of control elements in the methodology. Each deliverable, standard, activity, quality
- measure, and technique description is an element of control.
- ! Ceremony. The amount of precision and the tightness of tolerance in the methodology. Greater ceremony corresponds to tighter controls.
- ! Weight. The conceptual product of size and ceremony.
- ! Precision. How much you care to say about a topic?
- ! Accuracy. How correct you are when you speak about a topic?
- ! Relevance. Whether or not you speak about a topic?
- ! Tolerance. How much variation is permitted in the execution of the methodology?

TAILORING

- Reduces cost
    - By aligning process intensity with project risk and complexity, tailoring can reduce demands for: Forms, Checklists, Processes, Procedures, Templates
    - ! Can free valuable time for engineering and testing resources that can cause small projects to proceed at a sluggish pace.
    - ! On average, tailoring can reduce process intensity by 3X-6X which can equate to more than 20% savings in project costs and other costs associated with standards, compliance and project oversight.
- Creates repeatable, proven processes
- Mitigates compliance, related risks
    - Pre-populating schedules with compliance-related processes, templates and policies based on knowledge of the compliance/standards teams, not relying solely on the project lead
    - ! Enabling the project team to conduct pre-audit runthroughs where gaps in compliance can be highlighted and addressed prior to critical and visible audits or checkpoint/Authorization-to-Proceed (ATP) meetings
    - ! Providing a Compliance Checklist that enables the team to clearly gauge progress towards compliance using a Red/Yellow/Green model
- Enables risk based testing

TAILORING STEPS-------- 182
ASSESMENT ----------181

ROLES

Process engineer

- " Tailoring the process to match the specific needs of the project.
- " Educating and mentoring project members on process related issues.
- " Ensuring that valuable project experience is harvested and fed back into the process.
- " Assisting the Project Manager in planning the project

Project Manager

- " … on small projects takes the role of process engineer
- " Validates that the tailored process fits the project
- " Fine tunes the tailoring
- " Provides feedback to the process engineer to update the process

Process Activities

! The following process activities occur in a project:

- " Tailor the process for the project
- " Develop the development case (tailored methodology)
- " Prepare templates for the project
- " Prepare guidelines
- " Launch the development process

Artefacts

- ! Development Organization Assessment (cross - project)
- ! Development Case
- ! Project Templates
- ! Project Guidelines
- ! Feedback to process (improve the organization process)

Tailor the process

- ! Goals
  - " To right-size the software development process according to the specific needs of the project
  - " To provide a relevant and accessible process description for the members of the project

- ! Activities
  - " Analyze the Project
  - " Define the Scope of the Process
  - " Extend the Process Framework (optional)
  - " Configure the Process
  - " Prepare the Process for the Project

    - " Introduce the Process to the Project Members
    - " Maintain the Process

Develop the Development Case

- ! Goals
    - " To develop a development case that describes the software-development process for a project (or projects).
    - " To relate the development case to the organization-specific process.

- ! Activities
    - " Decide How to Perform Each Discipline
    - " Tailor Artifacts per Discipline
    - " Modify Disciplines and Activities
    - " Choose Lifecycle Model
    - " Identify Stakeholders
    - " Map Roles to Job Positions
    - " Describe Sample Iterations
    - " Document the Development Case
    - " Maintain the Development Case

Prepare templates for the project

- ! Goals
    - " harvest existing or develop new templates for use by the project.
    - " prepare the templates for project use by partially instantiating them with project specific information.
    - " make the existing templates accessible to the project members when needed

- ! Activities
    - " Identify Templates for the Project
    - " Prepare Templates for Project Use
    - " Maintain Templates

Prepare guidelines for the project

- ! Goals
    - " harvest existing or develop new guidelines for use by the project.
    - " make the existing guidelines accessible for the project members when needed.

- ! Activities
    - " Identify the Project's Needs for Guidelines
    - " Prepare Guidelines for Project Use
    - " Maintain Guidelines

Launch the development process
! Goal
" make the project members use the development process tailored for the project, together with the supporting tools.

! Activities
" Make the changes public
" Educate project members
" Collect feedback

Guidelines discussion (1)
! Business Modeling Guidelines
" Describes how you should model business use cases, business workers, and business entities.
" Should be considered when the project needs to formally model the business to build a new system. The degree of business process redesign, or the complexity of the business process, dictates how comprehensive they need to be.
! Use-Case Modeling Guidelines
" Needed whenever use cases will play a significant part in capturing the behavior of the system.
" Should contain modeling conventions such as relationships to use, styles to follow for textual descriptions.
! Design Guidelines
" A product of the architecture definition. It describes the guidelines to be followed during design, architectural design, and implementation.
! Programming Guidelines
" Specific to the actual implementation language(s) and class libraries selected for the project.
" should specify
! how to present code layout and commenting,
! how to use naming conventions,
! how to use language features.
! precautions regarding certain language features.
! User-Interface Guidelines
" Should give project-specific rules and recommendations for building the user interface.
" Often reference external publications, such as The Windows Interface Guidelines for Software Design, by Microsoft® Corporation.
! Tool Guidelines
" Describe how the project makes the best use of the selected tool set. Will often include:
! Installation information, such as version, configuration parameters,
! Limitations in functionality, and functionality that the project decided not to use
! Workarounds
! Integration with other tools including procedures to follow, software to use, and principles to apply.
! Test Guidelines
" Used to record adjustments (often tactical) to the way the test process is enacted on a given project, and to capture project-specific practices discovered during the dynamic enactment of the test process.
" Examples of test guidelines are test completion criteria and defect management guidelines.

Development Case Goal
! A development case shows how the generic RUP applies to the context of your organization.
=> you modify the process and adapt the terminology.
! A development case also provides an overview of the process to be followed, something understood by everyone on the project.

Building a Development Case
! The development case should not capture the entire process ! Responsibility and decisions about the process artefacts are delegated to members of the development team ! One reason for having a project process description at all is so several people can share information ! If process does not live in projects then the cost of maintaining the process description may be too high

Lecture 5
The project structure
WBS

Planning – What, When, Who, How much?

Project Scope Management

- ! Scope Planning
  - ! Defining and managing the scope is key to project success
  - ! Detailed project scope statement process
  - ! Process for creating the WBS from the detailed statement
  - ! Formal verification specification
  - ! Change control process

- ! Scope Definition
  - ! Detailed project scope statement
  - ! Change Requests
  - ! Project Scope Management Plan update
  - ! Detailed project scope statement
    - " Project Objectives (Specific, Measurable, Attainable, Relevant, Time-limited)
    - " Requirements
    - " Boundaries
    - " Deliverables
    - " Acceptance Criteria
    - " Constraints
    - " Assumptions
  - ! Detailed project scope statement
    - " Initial Organization
    - " Initial Risk
    - " Milestones
    - " Fund limitations
    - " Cost estimate
    - " Configuration Management Requirements
    - " Specifications
    - " Approval Requirements

- ! Create WBS
  - ! Project Scope Statement Updates
  - ! Work Breakdown Structure
  - ! WBS Dictionary

    - ! Scope Baseline
    - ! Project Scope Management Plan Updates
    - ! Change Requests

- ! Scope Verification
    - ! Assures that the detailed scope definition, WBS and WBS dictionary are formally reviewed and approved by the project stakeholders
    - ! Accepted deliverables
    - ! Change Requests
    - ! Recommend corrective actions

- ! Scope Control
    - ! Assures that all Requested Changes and Corrective Actions are processed by the Integrated Change Control Management process
    - ! Project Scope Statement (Updates)
    - ! Work Breakdown Structure (Updates)
    - ! WBS Dictionary (Updates)
    - ! Scope Baseline (Updates)
    - ! Requested Changes
    - ! Recommended Corrective Action
    - ! Organizational Process Assets (Updates)
    - ! Project Management Plan (Updates)

What is the WBS?

- ! WBS is a hierarchical decomposition of work
- ! Can be action **driven or deliverable driven (verbs or nouns)**
- ! Organizes and defines the total scope of the project
- ! Each level of the WBS represents an increased level of detail

WBS Goals

- ! Improve estimates
- ! Keep the team focused
- ! Assign work to resources
- ! Keep the project on track

EXAMPLE slide 256!!!!!!

Conventional WBS Issues

- ! Prematurely structured around the product design
- ! Prematurely decomposed, planned and budgeted into either too much detail or too little detail
- ! They are project specific, and cross project comparison is impossible

Evolutionary WBS

- ! An evolutionary WBS should organize around the PROCESS framework than around the product framework
- ! Example (what the book says)

- " First-level WBS elements are Disciplines
- " Second-level elements are lifecycle phases
- " Third-level should focus on the activities that produce the artefacts of every phase/iteration

WBS tailoring criteria

- ! Project scale
- ! Organizational structure
- ! Custom development
- ! Business context
- ! Precedent experience

WBS Creation Activities

- ! Identifying the deliverables and related work
- ! Structuring and organizing the WBS
    - " Top-down approach
    - " Chronological approach
    - " Bottom-up approach
- ! Developing and assigning identification codes to the WBS components
- ! Verifying that the degree of decomposition of the work is necessary and sufficient.

WBS is the project's plan "architecture"

- " It must encapsulate change
- " It must evolve with the appropriate level of detail through the project lifecycle
- " It must cover ALL project tasks, and NO MORE

Project organization key points

- ! Organizational structures form the architecture of the teams
- ! Organizations involved in software line of business need to have an organization that supports a common process
- ! Project organizations need to allocate artefacts and responsibilities clearly and balanced across project team(s)
- ! The project organization must evolve with the WBS and the project lifecycle

LECTURE 6
TIME MANAGEMENT

WHEN
ESTIMATION
SCHEDULING

TIME MANAGEMENT PROCESS:

- ! Activity Definition
- ! Activity Sequencing
- ! Activity Resource Estimating
- ! Activity Duration Estimating
- ! Schedule Development
- ! Schedule Control

4 Activity Dependency Types

- ! Mandatory Dependencies
  - " “Hard logic” dependencies
  - " Nature of the work dictates an ordering
  - " Ex: Coding has to precede testing
  - " Ex: UI design precedes UI implementation
- ! Discretionary Dependencies
  - " “Soft logic” dependencies
  - " Determined by the project management team
  - " Process-driven
  - " Ex: Discretionary order of creating certain modules
- ! External Dependencies
  - " Outside of the project itself
  - " Ex: Release of 3rd party product; contract signoff
  - " Ex: stakeholders, suppliers, year end
- ! Resource Dependencies
  - " Two task rely on the same resource
  - " Ex: You have only one DBA but multiple DB tasks

Activity sequencing [1]

- ! Finish-to-start (FS) – the start of a task can be only after the finish of another

- ! Start-to-Start (SS) - start of one task triggers the start of another
- ! Finish-to-finish (FF) – finishing a task triggers the finish of another
- ! Start-to-Finish (SF) – start of one task triggers finish of another

Constraints

- ! As Soon As Possible
- ! As Late As Possible
- ! Start No Earlier Than
- ! Finish No Earlier Than
- ! Start No Later Than
- ! Finish No Later Than
- ! Must Start On
- ! Must Finish On

Estimations

- ! Once tasks have been identified, the time and resources necessary to accomplish them must be determined.
- ! This process is called estimating.

- ! Effort/Work = how much work will the activity need to be completed
- ! Resources = how many resources will be working on the activity
- ! Duration = how long will the activity last for

=> Estimate effort first

Effort

- ! Your best shot for providing estimations (how complex/how much work does the activity require?)
- ! Measured in man/month (3 m-m = 1 person working for 3 months; 3 people working for one 1 month, 6 people working half-time for 1 month, etc.)
- ! Communication increases the time to complete activities!

Duration

- ! Measured in (work-)hours, (work-)days, (work-)months, …
- ! Calendar time != duration: calendar time includes non-working days, holidays, …
- ! Usually:
- " A duration of 5 days = 40 hours (8 hours a day) = 1 calendar week (Sat and Sun rest time)

DURATION = EFFORT / RESOURCES

! When working with planning tools, you change one variable at a time.
! Standard characterization:

- **" Fixed Unit**. A task in which the assigned resources is a fixed value and any changes to the amount of work or the tasks duration do not affect the tasks units.
- " **Fixed Work**. A task in which the amount of work is a fixed value and any changes to the tasks duration or the number of assigned resources do not affect the tasks work.

- " **Fixed Duration**. A task in which the duration is a fixed value and any changes to the work or the assigned resources, don't affect the tasks duration

! To achieve credible results cost and schedule budgets should be done in two ways

- " Estimating top-down, starting from project level
- " Estimating bottom-up starting from task micro-analysis
- " Parametric estimating – use historical data adjusted to current project
- Rough estimation 25-50% error
- Second estimation – refine 25% error

Scheduling

- ! Once tasks (from the WBS) and effort (from estimation) are known: then schedule
- ! Primary objectives
  - " Best time
  - " Least cost
  - " Least risk
- ! Secondary objectives
  - " Evaluation of schedule alternatives
  - " Effective use of resources
  - " Communications

RULES OF THUMB

TERMINOLOGY

- PRECEDENCE
- CONCURRENCE
- LEADS AND LAG TIME
- MILESTONES
- DELIVERABLE
- SLACK AND FLOAT – slack time

Padding estimates

- ! Bloated estimates – everyone adds some padding=> totally inaccurate estimates
- ! Padding games - managers trim every estimate => more padding next time
- ! Lack of feedback – with no feedback no estimator can improve

SAFETY MARGINS: 50-75% high risk project, 5-10% low risk project

Scheduling Techniques

- ! Mathematical Analysis
  - " Network Diagrams
    - ! Critical Path Method (CPM)
    - ! PERT
- ! Bar Charts
  - " Milestone Chart
  - " Gantt Chart

Network Diagrams

- ! Two classic formats
  - " AOA: Activity on Arrow
  - " AON: Activity on Node
- ! Each task labeled with
  - " Identifier (usually a letter/code)
  - " Duration (in std. unit like days)
- ! There are other variations of labeling
- ! There is 1 start & 1 end event
- ! Time goes from left to right

Network Diagrams

- ! AOA consists of
  - " Circles representing Events
    - ! Such as 'start' or 'end' of a given task
  - " Lines representing Tasks
    - ! Thing being done 'Build UI'
  - " a.k.a. Arrow Diagramming Method (ADM)
- ! AON
  - " Tasks on Nodes
    - ! Nodes can be circles or rectangles (usually latter)
    - ! Task information written on node
  - " Arrows are dependencies between tasks
  - " a.k.a. Precedence Diagramming Method (PDM)

! Advantages

- " Show precedence well
- " Reveal interdependencies not shown in other techniques
- " Ability to calculate critical path
- " Ability to perform "what if" exercises !

Disadvantages

- " Default model assumes resources are unlimited
- ! You need to incorporate this yourself (Resource Dependencies) when determining the "real" Critical Path
- " Difficult to follow on large projects

Critical Path
! the sequence of project network activities which add up to the longest overall duration.
=> the shortest time possible to complete the project

EXAMPLE SLIDE 334 – 344

PERT

- ! Program Evaluation and Review Technique
- ! Based on idea that estimates are uncertain
- " Therefore uses duration ranges
- " And the probability of falling to a given range
- ! Uses an "expected value" (or weighted average) to determine durations
- ! Use the following methods to calculate the expected durations, then use as input to your network diagram

- ! Start with 3 estimates
  - " Optimistic
    - ! Would likely occur 1 time in 20
  - " Most likely
    - ! Modal value of the distribution
  - " Pessimistic
    - ! Would be exceeded only one time in 20

PERT FORMULA:

$$t_e = \frac{a + 4m + b}{6}$$

where

t_e = expected time
a = optimitistic time estimate
m = most likely time estimate
b = pessimistic time estimate

! Confidence Interval can be determined
! Based on a standard deviation of the expected time
! Using a bell curve (normal distribution)

$$s = \frac{b - a}{6}$$

! For the whole critical path use

$$s_{cp} = \sqrt{s_1^2 + s_2^2 + \ldots + s_n^2}$$

CPM vs. PERT
! Both use Network Diagrams
! CPM: deterministic
! PERT: probabilistic
! CPM: one estimate, PERT: three estimates
! PERT is infrequently used

EXERCISE Slide 352!!!!!!!!!!!!!

Schedule compression
! Fast-tracking
" Do critical-path activities in parallel
=> rework, increase risks, needs communication
! Crash
" Cost-schedule trade-off (decrease in schedule => increase in cost)

LECTURE 7
Monitoring and CONTROL

The goal of Project Monitoring and Control (PMC) is to provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.

Monitoring – collecting, recording, and reporting information concerning project performance that project manager and others wish to know

Controlling – uses data from monitor activity to bring actual performance to planned performance

Why, When, How, What do we monitor???

Why do we monitor? Goals

- ! Provide project manager and development team with the following:
- " An accurate assessment of the progress to date
- " Insight into the quality of the evolving software product
- " A basis for estimating cost and budgeting with increased accuracy over time

What do we monitor?

- ! Men (human resources)
- ! Machines
- ! Materials
- ! Money
- ! Space
- ! Time
- ! Tasks
- ! Quality/Technical Performance

What do we obtain?
Inputs

- ! Time
- ! Money
- ! Resources
- ! Material Usage
- ! Tasks
- ! Quality/Technical

Performance
Outputs

- ! Progress
- ! Costs
- ! Job starts
- ! Job completion
- ! Engineering / Design
- changes
- ! Variation order (VO)

PMC – Activities

- ! Continuously monitor progress
  - " Examine progress on all key dimensions of the project; goals likely to be met?
- ! Conduct team reviews
  - " Communicate status (technical activities); plan for next activities of the project
- ! Conduct formal progress reviews
  - " Monthly basis with senior management and key stakeholders
- ! Manage changes
  - " Identify, evaluate, prioritize, and control changes to the project
- ! Revise the plan
  - " Significant changes need to be reviewed and agreed to by those who originally approved
  - the plan
- ! Conduct work product reviews
  - " Walkthroughs, technical reviews and inspections, based on quality goals

PMC – Example Measures

- ! Milestone attainment
  - " Maintain the initial baseline, as well as the most recent update
  - " Report achievement and variance to both
- ! Effort spent
  - " Compare initial effort estimates for each major WBS element with actual effort spent
- ! Budget/Cost performance
  - " Compare rate of spending on the project by period (week or month) compared to the
  - planned spending
- ! Requirements change
  - " Track requirements change by period (month usually): total number of requirements,
  - number added in this period, number deleted in this period, and number changed in this
  - period
- ! EVA – Earned Value Analysis

Scope Verification

! Inputs:

- " Work results: which deliverables – fully/partially completed
- " Product documentation: describes project's products
- " WBS: used to verify work of the project; defines scope baseline
- " Scope statement: defines scope in some detail
- " Project plan

! Outputs:

" Formal acceptance: must be documented; can be conditional

! Tools & Techniques:

" Inspection: measure/examine/test results vs. requirements

PROJECT CONTROL CYCLE:

ACTION – PLAN – MONITOR - COMPARE - RE-PLAN – ACTION ….

Scope Change Control

- ! Concerned with: influencing change factors; determining occurrence of a scope change; managing changes if/when occur
- ! Must be integrated with other control processes

- ! Inputs:
  - " WBS
  - " Performance reports: e.g. which interim deliverables have been completed, which not
  - " Change requests:oral/written, direct/indirect, external/internal, legally mandated/optional
  - " Scope management plan: how scope is managed; how changes are integrated; assessment of expected stability; how changes are identified/classified

- ! Outputs:
  - " Scope
  - " Corrective changes
  - " Lessons learned
  - " Adjusted baseline
- ! Tools & Techniques:
  - " Scope change control
  - " Performance measurement
  - " Additional planning: e.g. modification to WBS

Schedule Control
! Concerned with: influencing change factors; determining occurrence of a scope change; managing changes if/when occur [same as scope change control]
! Must be integrated with other control processes

Schedule Control

- ! Inputs:
  - " Project schedule: schedule baseline
  - " Performance reports
  - " Change requests
  - " Schedule management plan
- ! Outputs:
  - " Schedule updates
  - " Corrective action
  - " Lessons learned
- ! Tools & Techniques:
  - " Schedule change control system

- " Performance measurement
- " Additional planning
- " Project management software
- " Variance analysis

Schedule Control – metrics

! Measurement done at WBS unit level:

- " CAP – Control Account Plan = work package with extra features:
  - ! Assignment of responsibility (organization/individual)
  - ! Division (if needed) in lower-level work packages
  - ! Metrics for measuring performance: milestones, %complete…
- ! EVA – Earned Value Analysis
  - " Variance analysis – not full picture
  - " Planned vs. completed work, to determine if cost, schedule, and work
  - are progressing as planned
- ! Three key values: used to derive further schedule/cost indicators:
  - " BCWS: budgeted cost of work scheduled; a.k.a. PV – Planned Value
  - " ACWP: actual cost of work performed; a.k.a. AC – Actual Cost
  - " BCWP: budgeted cost of work performed; a.k.a. EV – Earned Value
- ! Schedule indicators
  - " SV: Schedule Variance:
    - ! SV = BCWP – BCWS
    - ! < 0 => behind schedule
  - " SPI: Schedule Performance Index
    - ! SPI = BCWP/BCWS
    - ! < 1 => behind schedule

Cost Control

- ! Concerned with: same as before
- ! Must be integrated with other control processes
- ! Includes:
  - " Monitor cost performance, detect and understand variances from plan
  - " Ensure all appropriate changes – recorded accurately in cost baseline
  - " Prevent incorrect/inappropriate/ unauthorized changes to cost baseline
  - " Inform stakeholders
  - " Act to bring expected costs within acceptable limits
- ! Inputs:
  - " Cost baseline
  - " Performance reports
  - " Change requests
  - " Cost management plan
- ! Outputs
  - " Revised cost estimates
  - " Budget updates
  - " Corrective action
  - " Estimates at completion
  - " Project closeout

        - " Lessons learned
    - ! Tools & Techniques
        - " Cost change control system
        - " Performance measurement

" EVA
" Additional planning
" Software tools (e.g. MSProject) Cost Control – metrics – EVA
! Cost indicators:
" CV: Cost Variance
! CV = BCWP – ACWP
! < 0 => over the budget
" CPI: Cost Performance Index
! CPI = BCWP/ACWP
! < 1 => over the budget

EAC – Estimate At Completion
! Forecast of most likely total project costs
! EAC = actuals to date + new estimate for remaining work
" EAC = AC + ETC
" Used when original assumptions – proven flawed/no longer relevant to a change in conditions
! EAC = actuals to date + remaining budget
" EAC = AC + (BAC – BCWP);
" Used when current variances – atypical; not expected to appear in future
EAC = actual to date + remaining budget modified by a performance factor (often CPI)
" EAC = AC + (BAC – BCWP)/CPI
" Used when current variances – typical of future variances
" Once a project is 20% complete, the CPI does not vary from its current value by more than 10%.

EXAMPLE SLIDE 416-421!!!!!!!!!!!!!!!!!!

CRITICAL RATIO:

(actual progress/scheduled progress) * (budgeted cost / actual cost)

Quality Control
! Monitor and evaluate specific results against relevant quality standards; eliminate causes of unsatisfactory results.

! Often performed by a Quality Control Department

! Inputs:
" Work results: both process and product
" Quality management plan
" Operational definitions (metrics)
" Checklists

! Tools & Techniques:
" Inspection
" Control charts
" Pareto diagrams
" Statistical sampling
" Flowcharting
" Trend analysis

! Outputs:
" Quality improvement
" Acceptance decisions
" Rework
" Completed checklists
" Process adjustments

METRICS:
! Control charts = graphic display of results, over time, of a process.
! Used to monitor any type of output variable (cost/schedule variances, scope changes volume/frequency, errors in docs)

! Pareto diagrams = histogram, ordered by frequency of occurrence, that shows how many results were generated by type of cause
" Rank – used to guide corrective action
" Pareto's law: 80% of problems are due to 20% of causes

Flowcharting = show relations between system elements.
" Cause-and-effect: how various factors may be linked to potential problems/effects
" System/process flowcharts

! Trend Analysis = use math techniques to forecast future based on past:
" Technical performance: no. of errors identified; no. corrected
" Cost/Schedule performance: no. of activities/period completed with significant variances

Risk Monitoring & Control

- ! Keep track of identified risks; monitor residual risks; identify new risks; ensure execution of risk plans & evaluate their effectiveness
- ! Risks change as project matures

- ! Provides info that assists with making effective decisions in advance of the risks appearing
- ! Periodically communicate with stakeholders to agree upon level of risk in project
- ! Purpose – check if:
    - " Risk responses have been implemented as planned
    - " Risk responses are effective, or new must be implemented
    - " Project assumptions – still valid
    - " Risk exposure has changed (trend analysis)
    - " Risk trigger has occurred
    - " Proper policies and procedures are followed
    - " Risks not previously identified have occurred
- ! Inputs:
    - " Risk management plan
    - " Risk response plan
    - " Project communication: issues logs, action-item lists, jeopardy warnings, escalation notices.
    - " Additional risk identification and analysis
    - " Scope changes
- ! Tools & Techniques
    - " Project risk response audits
    - " Periodic project risk reviews
    - " Earned Value Analysis
    - " Technical performance measurement
    - " Additional risk response planning
- ! Outputs:
    - " Workaround plans
    - " Corrective action
    - " Project change requests
    - " Updates to the risk response plan
    - " Risk database
    - " Updates to risk identification checklists

- Performance Reporting
- ! Includes:
    - " Status reporting
    - " Progress reporting
    - " Forecasting
- ! Include info on scope, schedule, cost and quality; risk and procurement.
- ! Inputs:
    - " Project plan
    - " Work results
    - " Other project records (e.g. project Context)
-)! Tools & Techniques:
    - " Performance reviews
    - " Variance analysis
    - " Trend analysis
    - " EVA
    - " Information distribution tools and techniques
- ! Outputs:

- " Performance reports
- " Change requests

Integrated Change Control

- ! Concerned with: influencing change factors; determining occurrence of a change; managing changes if/when occur
- ! Original project scope and performance baseline – maintained by continuously managing changes
- ! Requirements:
  - " Maintain integrity of the performance measurements baseline
  - " Ensure changes in product scope – reflected in project scope
  - " Coordinate changes across knowledge areas (risk control, time control, schedule control…)
- ! Inputs:
  - " Project plan
  - " Performance reports
  - " Change requests
- ! Tools & Techniques:
  - " Change control system
  - ! Collection of formal, documented procedures that defines how project performance will be monitored and evaluated
  - ! Paperwork, tracking systems, processes, approval levels
- ! Evaluation groups: Configuration Control Board (CCB), Engineering Review Board (ERB), Technical Review Board (TRB), Technical Assessment Board (TAB)
- ! Tools & Techniques
  -) Configuration management: any documented procedure used to apply technical/administrative direction and surveillance to
  - ! Identify and document the functional and physical characteristics of an item or system
  - ! Control any changes to such characteristics
  - ! Record and report the change and its implementation status
  - ! Audit items and system, to verify conformance with requirements
-) Performance measurement
-) Additional planning
-) Project management information system (PMIS)
- ! Outputs:
  - " Project plan updates
  - " Corrective action
  - " Lessons learned

LECTURE 8
RISK MANAGEMENT

- ! Risk Identification
  - ! Risk Identification determines which risks might affect the project and documents their characteristics
  - ! Participants in risk identification activities can include the following :
    - " Project manager
    - " Project team members
    - " Risk management team
    - " Customers
    - " End users
  - ! Risk Identification is an iterative process
  - ! New risks may become known as the project progresses through its life cycle
  - ! Identify activities that may be affected by unexpected events
  - ! Risks can be identified in a number of ways
    - " General Risks – can happen to any task, resource or project
    - " Risk specific to the project
    - " Risks applicable to one task
    - " Resource specific risks
  - ! Review the schedule with experienced people
  - " Refer to previous projects files and archives
  - " Brainstorm likely events that may occur in your project
  - " Perform SWOT analysis (Strength, Weaknesses, Opportunities, Threats)

  - Checklist [1]
    - ! Key member of staff on critical path
    - ! Key skill on long-duration task
    - ! Project Manager on critical path
    - ! External dependencies without tight contract
    - ! Reliance on leading-edge technologies
    - ! Many internal dependencies with conflicting priorities
    - ! Ramping up resources
    - ! Assumptions about availability of people and equipments
    - ! Vacation time resource absences
    - ! Site shutdown for maintenance
    - ! Health and safety requirements and certifications
    - ! Identify risks that translate to project slippage
    - ! Identify near critical path that could become critical
    - ! Identify slack for non critical path
    - ! Identify risks on critical path that translate to slippage

  - ! Risk Assessment

    - ! Identify Probability and Impact of each risk
    - ! Key question: what event will have an influence on the project?
    - ! Probability: How likely is it to happen?

        - ! Impact: What will be the consequences?
    - Qualitative Risk Assessment
        - ! Qualitative Risk Analysis includes methods for prioritizing the identified risks for further action, such as
            - " Quantitative Risk Analysis
            - " Risk Response Planning
        - ! Qualitative Risk Analysis determines the priority of identified risks using their probability of occurring and the corresponding impact on project objectives
    - Analyzing risks
        - ! Identify uncertainties surrounding project decisions and outcomes
        - ! Identify mitigation plans for risks that are inherent in a project
        - ! Risk concept is often associated with the potential for loss in value, control, functionality, quality, or timeliness of completion of a project.
        - ! failure to maximize gain in an opportunity and uncertainties in decision making that can lead to a missed business opportunity
    - Qualitative Assessment Tools
        - ! Probability and Impact Matrix
        - " Ratings are assigned to risks based on their assessed probability and impact
        - " Evaluation of each risk's importance is conducted using a probability and impact matrix, which is used to rating the risks as low, moderate, or high priority
        - " The organization should determine which combinations of probability and impact result in a classification of high, moderate and low risk
    - Quantitative Risk Assessment [1]
        - ! Is performed on risks that have been prioritized as High-Risks by the Qualitative
        - Risk Analysis process
        - ! Analyzes the effect of those risk events and assigns a numerical rating to those risks
        - ! Uses techniques such as Monte Carlo simulation and decision tree analysis to:
        - " Quantify the possible outcomes for the project and their probabilities
        - " Identify risks requiring the most attention by quantifying their relative contribution to overall project risk
- RISK ASSESSMENT DOCUMENTS
- RISK ASSESSMENT METRICS

SLIDE 572 – 581!!!!!!!!!!!!!!

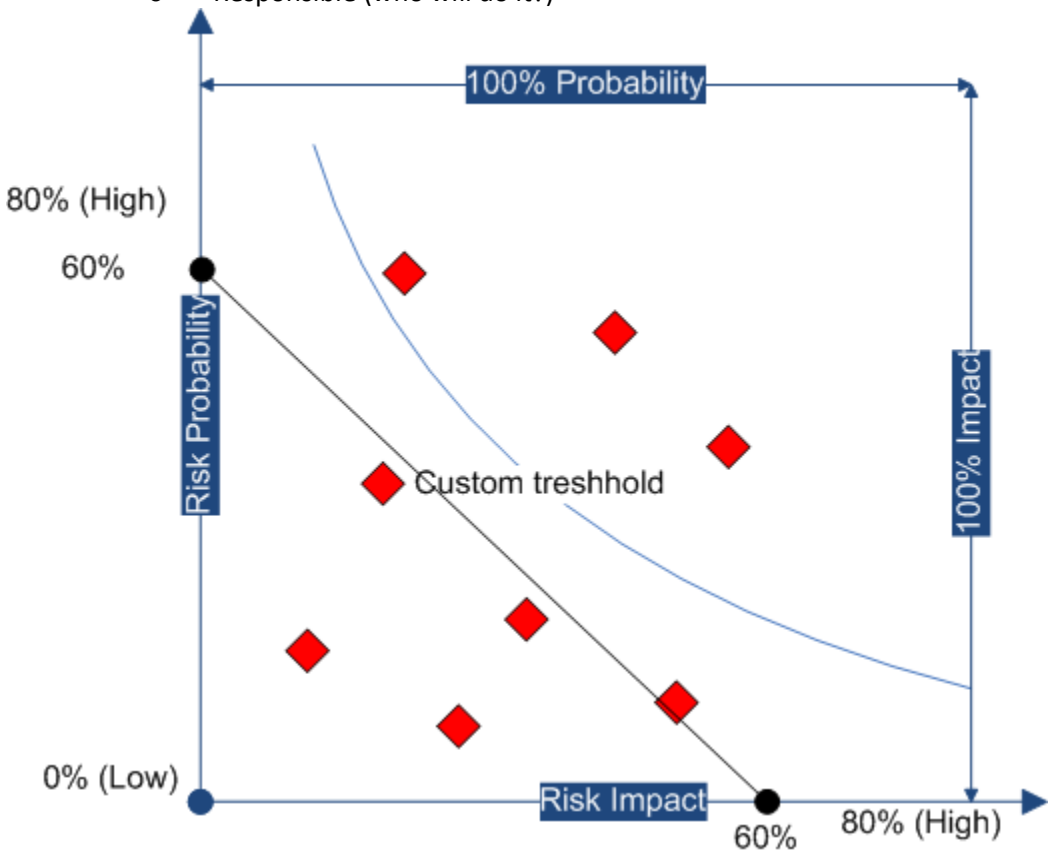
EXPOSURE = PROBABILITY * IMPACT

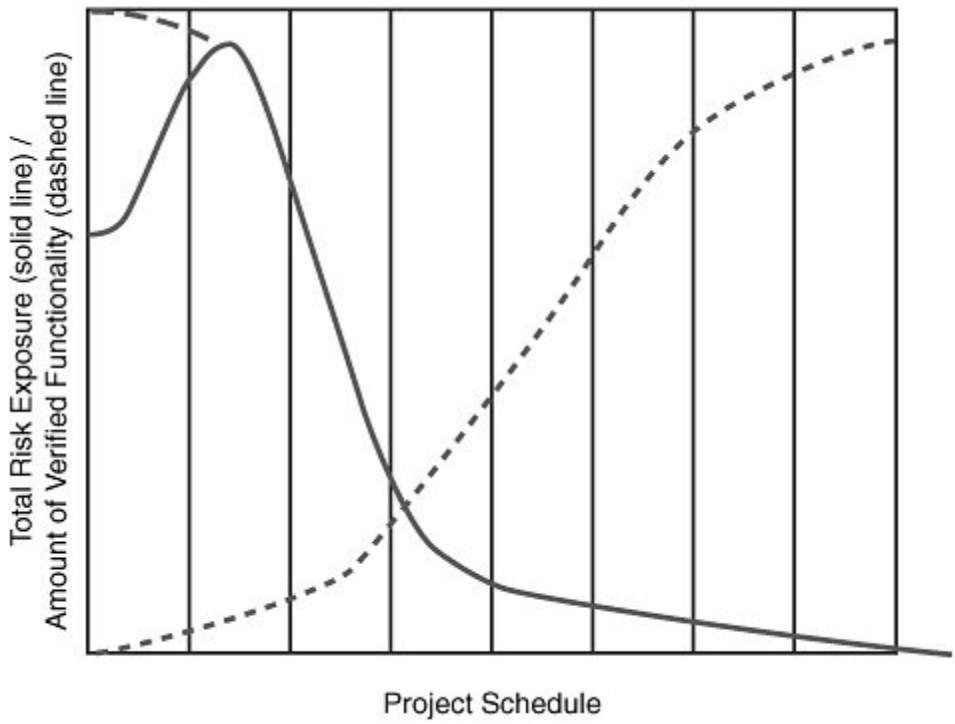
- ! Risk Evaluation

  - Risks can be placed on a graph according to probability and impact
    - ! Decide the zone on the graph that contains unacceptable risks:
    - " Depends on organization
    - " Depends on domain
    - " Depends on project

  - To move each risk out of that zone:
    - " Reduce the probability
    - " Reduce the impact

      ⇨ Reduce exposure
  - ! These actions are recorded in the risk register:
    - " Mitigation (what will be done to prevent?)
    - " Triggers (when will it be done?)
    - " Responsible (who will do it?)





Risk Metrics
! Measurements
" EP - Total Project Exposure
" Ei – Exposure for Risk i
" Ii – Impact for Risk i
" Pi – Probability for Risk i

Risk Metrics Trend

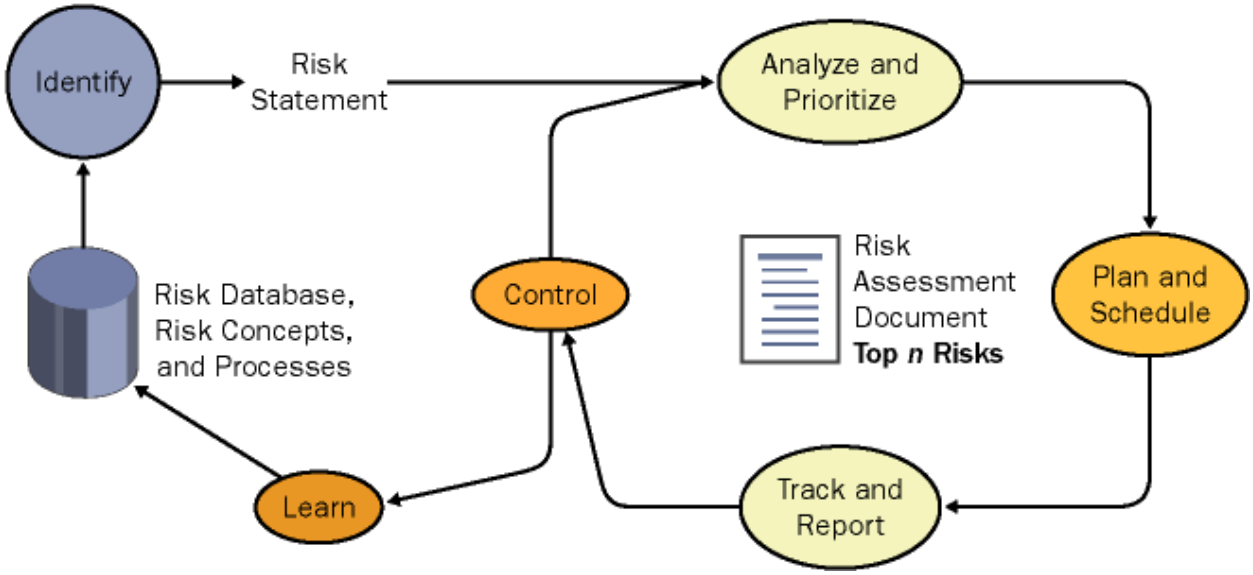
- Risk Exposure Metric (should → 0)

$$E_P = \sum_{i=1}^{n} E_i = \sum_{i=1}^{n} P_i * I_i$$

- Risk Exposure Metric Trend (should <0)

$$\overline{E}_P = \frac{dE_p}{dt}$$

- ! Risk Management
- Risk Management Steps [1]
  - ! Risk identification – allows individuals to identify risks so that the team becomes aware of any potential problems.
  - ! Risk analysis – transforms the estimates or data about specific project risks that emerges during risk identification into a form the team can use to make decisions about prioritization.
  - ! Risk planning – uses the information obtained from risk analysis to formulate strategies, plans, and actions.
  - ! Risk tracking – monitors the status of specific risks and documents the progress in their respective action plans.
  - ! Risk control – is the process of executing risk action plans and their associated status reporting.
  - ! Risk learning – formalizes the lessons learned and relevant project documents and tools, and records that knowledge in reusable form for use within the team and by the enterprise.



- Ri**sk Management Plan Development**
- ! Define risk management procedures and tools ! Create initial risk list
- ! Assign risk management team
- ! Decide strategies for managing top 10 risks
- ! Define risk indicators for top 10 risks
- ! Set schedule for risk reporting and review

- Risk Response Planning
  - ! Develop options and determine actions to enhance opportunities and minimize threats to project objectives
  - ! Assign responsibility to individuals or parties for each risk response.

- ! Risk response must be:
  - " Proportional to the severity of the risk
  - " Cost effective
  - " Timely
  - " Realistic
  - " Accepted by all parties involved
  - " Owned by a person or a party

- Risk Response Strategies (-)
  - ! Avoidance
  - ! Reduction (Mitigate)
  - ! Transfer
  - ! Acceptance

- Risk Response Strategies (+)
  - ! Exploit (opposite to Avoidance)
  - ! Enhance (opposite to Reduce)
  - ! Share (opposite to Transfer)
  - ! Acceptance

EXAMPLE SLIDE 601

Types of risks (from RUP)

- ! Resource Risks
  - Organization
  - Funding
  - People
  - Time
- ! Business Risks
  - What if a competitor reaches the market first?
  - ! What if project funding is jeopardized
  - ! Is the projected value of the system greater than the projected cost?
  - ! What if contracts cannot be made with key suppliers?
- ! Technical Risks
  - Scope risks
  - Technological risks
  - Dependency risks
- ! Schedule Risks
  - ! Are indicated by metrics
    - " complexity
    - " real-time constraints
    - " storage constraints
    - " experience
    - " availability of good tools
    - " schedule pressure

LECTURE 9
CONFIGURATION AND CHANGE MANAGEMENT

Integrated Change Control is the process necessary for controlling factors that create changes to make sure those changes are beneficial, determining whether a change has occurred, and managing the approved changes, including when they occur.
! ICC process is performed throughout the project, from project initiation through project closure.

ACTIVITY

- Identifying that a change needs to occur or has occurred.
- ! Influencing the factors that circumvent integrated change control so that only approved changes are implemented.
- ! Reviewing and approving requested changes.
- ! Managing the approved changes when and as they occur, by regulating the flow of requested changes.
- ! Maintaining the integrity of baselines by releasing only approved changes for incorporation into project products or services, and maintaining their related configuration and planning documentation.
- ! Reviewing and approving all recommended corrective and preventive actions.
- ! Controlling and updating the scope, cost, budget, schedule and quality requirements based upon approved changes, by coordinating changes across the entire project
- ! Documenting the complete impact of requested changes.
- ! Validating defect repair.
- ! Controlling project quality to standards based on quality reports

Inputs

- ! Project Management Plan
- ! Requested Changes
- ! Work Performance Information
- ! Recommended Preventive Actions
- ! Recommended Corrective Actions
- ! Recommended Defect Repair
- ! Deliverables

Tools and Techniques

- ! Project Management Methodology
- ! Project Management Information System
- ! Expert Judgment

Outputs[1]

- ! Approved Change Requests
- ! Rejected Change Requests
- ! Project Management Plan (Updates)
- ! Project Scope Statement (Updates)
- ! Approved Corrective Actions
- ! Approved Preventive Actions

CMMI – Configuration
Management
! The purpose of Configuration Management (CM) is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.

CMMI Specific Goals and Practices [1]

- ! SG 1 Establish Baselines
  - " SP 1.1 Identify Configuration Items
  - " SP 1.2 Establish a Configuration Management System
  - " SP 1.3 Create or Release Baselines

- ! SG 2 Track and Control Changes
  - " SP 2.1 Track Change Requests
  - " SP 2.2 Control Configuration Items

- ! SG 3 Establish Integrity
  - " SP 3.1 Establish Configuration Management Records
  - " SP 3.2 Perform Configuration Audits

Change request
Change attributes

Authorisation of a change by the Change Advisory Board

- ! A standard CAB agenda
  - " Failed changes
  - " Backed out changes
  - " RFCs to be assessed by CAB members
  - " RFCs that have been assessed by CAB members
  - " Implemented change are reviewed
  - " The change management process – including any amendments made to the process, as well as proposed changes to the process (as appropriate)
  - " Change management successes for the period under discussion CAB considerations for each change (prior to authorisation)

CAB considerations for each change (prior to authorisation)

- ! Impact assessment (on the business)
- ! Risk assessment (on the business)
- ! Effect upon the infrastructure and customer service, as defined in the SLA, and upon the capacity and
- performance, reliability and resilience, contingency plans, and security
- ! Impact on other services that run on the same infrastructure (or on software development projects)
- ! Resource assessment – the IT, business and other resources required to implement the change, covering the likely costs, the number and availability of people required, the elapsed time, and any new infrastructure elements required
- ! The impact on non-IT infrastructures within the organisation
- ! Effect/risk/impact of not implementing the change

- ! Technical capability and technical approval
- ! Financial approval (if required)
- ! Third party/supplier involvement in the implementation of the change
- ! Business approval (if required)
- ! Review/assessment of the change priority

Activities of change building

- ! building a new production module
- ! creating a new version of one or more software modules
- ! purchasing equipment or services externally
- ! preparing a hardware modification
- ! producing new or amended documentation showing the components of the change build
- ! devising a back out plan
- ! devising testing requirements, as appropriate
- ! documenting required resources for the change implementation

Change review

- ! The change has had the desired effect and met its objectives
- ! Users and customers are content with the results, or to identify any shortcomings
- ! There have been no unexpected or undesirable side effects to functionality, availability, capacity/performance, security, maintainability etc.
- ! The resources used to implement the change were as planned
- ! The implementation plan worked correctly (so include comments from the implementers)
- ! he change was implemented on time and to cost
- ! The backout plan functioned correctly, if the backout plan was implemented

Role: Configuration Manager[1]

- ! Manages the overall Configuration Management (CM) infrastructure and environment for the product development team
- ! Role is often shared by a Configuration (or Change) Control Board (CCB)
- ! Consists of representatives from all interested parties, including customers, developers, and users
- ! In a small project, a single team member, such as the project manager or software architect, may play this role
- ! Should understand configuration management principles
- ! Should be skilled in estimating cost and schedule impacts of change requests
- ! Should be able to communicate effectively in order to negotiate scope changes and in order to determine how each change request should be handled and by whom

Role: Change Control Manager[1]

- ! This role defines and oversees the change control process.
- ! This role is often shared by a Configuration (or Change) Control Board (CCB) and consists of representatives from all interested parties like customers, developers, and users
- ! In a small project the project manager or software architect, may play this role
- ! Should understand change management principles

- ! Should be skilled in estimating cost and schedule impacts of change requests
- ! Should be able to communicate effectively in order to negotiate scope changes and in order to determine how each change request should be handled and by whom

Plan Project Configuration & Change Control[1]

- ! Establishes an appropriate plan for managing and controlling change to the artifacts that are developed as work products of the software development process
- ! Establish Configuration Management(CM) Policies
  - " Define Configuration Identification Practices
  - " Define Baseline Practices
  - " Define Archiving Practices
  - " Define Configuration Status Reporting Requirements
- ! Write Configuration Management (CM) Plan
  - " Write the CM Plan
  - " Review and Approve the CM Plan
  - " Maintain the CM Plan
- ! Establish Change Control Process
  - " Establish the Change Request Process
  - " Establish the Change Control Board
  - " Define Change Review Notification Protocols

Create Project Configuration Management (CM) Environments[1]
! This activity establishes an environment where the overall product can be developed, built, and made available for stakeholders.
! Set up the CM Hardware Environment
" Set up the CM Hardware Environment
" Map the Architecture to the Repository
" Create Initial Set of Versioned Elements
" Define Baseline Promotion Levels
! Create Integration Workspaces
" The integration workspace is where subsystem and system integrators convince themselves that separately developed and tested components can indeed work together as a product.
" Integrators combine the elements delivered to the integration workspace to produce a Build.'

Manage Change Requests [1]

- ! This activity ensures that due consideration is given to the impact of change on the project and that approved changes are made within a project in a consistent manner.
- ! Submit Change Request
  - " Complete CR Form

        - " Submit the Change Request
        - ! Example:
        - " Sample Change Request Form.doc
        - " Change Request Form.doc
    - ! Update Change Request
        - " Retrieve the Change Request Form
        - " Update and Resubmit the Change Request Form
    - ! Review Change Requests
        - " Schedule CCB Control Meeting
        - " Retrieve Change Requests for Review
        - " Review Submitted Change Requests
    - ! Confirm Duplicated or Rejected CR
        - " Retrieve the Change Request Form
        - " Confirm Duplication or Validity
        - " Update the Change Request Status
    - ! Schedule and Assign Work
        - " Allocate Change Request to an Iteration
        - " Assign Responsibility
        - " Describe Work and Expected Outputs
        - " Budget Effort and other Resources
        - " Set Schedule
        - " Re-plan
        - " Issue Work Order
    - ! Verify Changes in Build
        - " Resolve Change Request
        - " Verify Changes in Test Build
        - " Verify Changes in Release Build

Monitor & Report Configuration Status [1]
! This activity provides visibility to configuration change activity through ongoing monitoring and reporting.
! Report on Configuration Status goals:
" Support project Configuration Status Accounting tasks.
" Facilitate product review through defect tracking and reporting tasks.
" Ensure that data is 'rolled-up' and reported for the purposes of tracking progress and trends.
! Perform Configuration Audit
" Perform Physical Configuration Audit (check if baseline contains all required work products)
" Perform Functional Configuration Audit (check that a baseline meets requirements)
" Report Findings

Change and Deliver Configuration Items[1]

- ! This activity manages project artifacts and the work involved from their initial creation as private artifacts through to their delivery and general availability to the project team and other stakeholders.
- ! Create Development Workspace
    - " A development workspace is a private development area that provides an environment in which a team member can make changes to work products without the changes becoming immediately visible to other team members

- ! Make changes
  - " Checks out the files that need to be changed.
  - " Makes the changes.
  - " Performs unit tests to verify the changes.
  - " Gets the changes approved.
  - " Checks in the changes.
  - " Promote the changes.
- ! Deliver Changes
  - " Prepare for Delivery
  - " Deliver Changes
  - " Update Work Order Status
- ! Update Workspace
  - " Ensure team members are working on the most recent versions of the project files
  - " Update the files displayed in the development view (workspace) with those in the recommended baseline
- ! Create Baselines
  - " Ensure that all developed work products are captured and archived, at given points in
  - time, as a basis for further product development
  - " Good candidates for a baseline are the sets of files and directories under version control
  - that are developed, integrated and released together
- ! Promote Baseline
  - " Establish when the creation of a Baseline is completed, and how a completed Baseline
  - must be labeled
  - " Baselines (individually tested components from various implementers, and development
  - teams, combined together to work together as a product) are 'tagged' to reflect the level
  - of software maturity, stability and quality they may have achieved

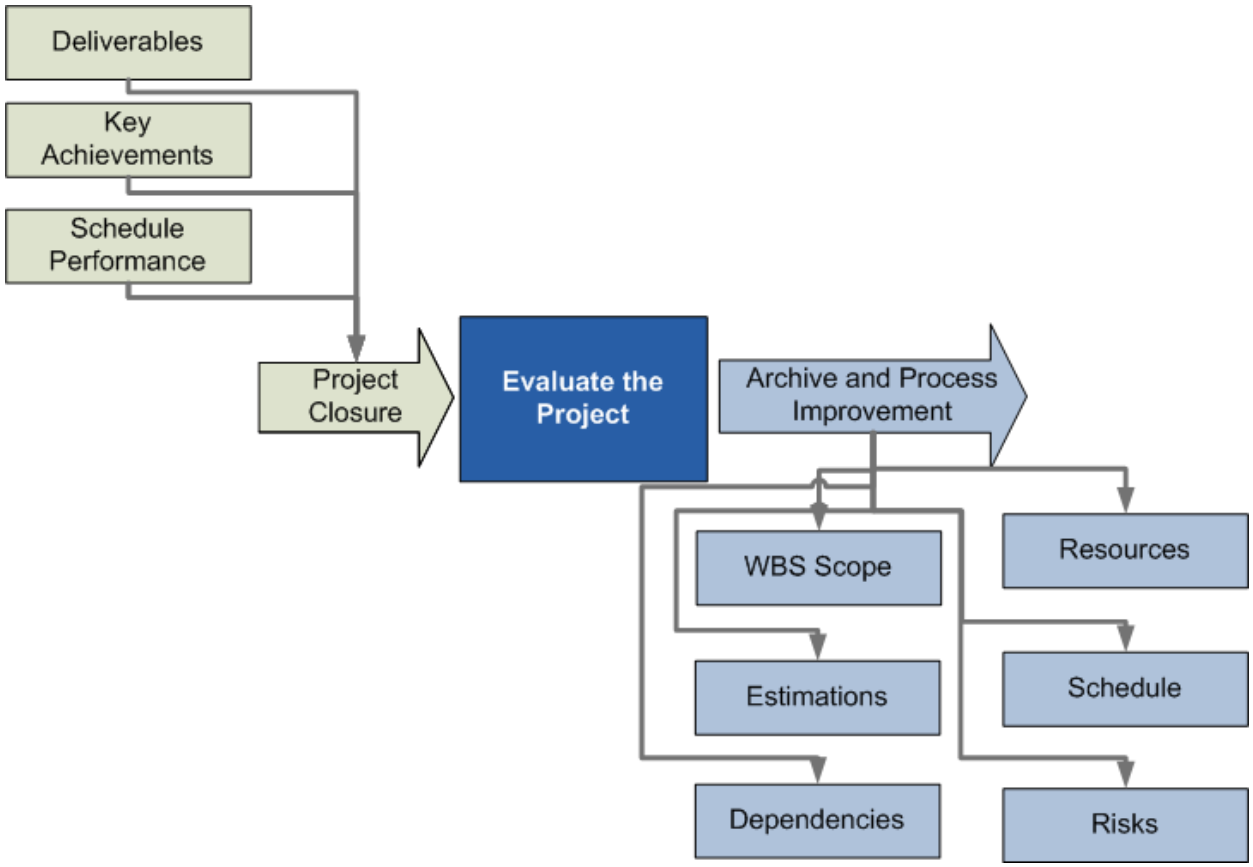
Manage Baselines & Releases [1]

- ! This activity ensures that consistent sets of related or dependent artifacts can be identified as part of a "baseline" for various purposes, such as the identification of release candidates, product versions, artifact maturity or completeness.
- ! Create Deployment Unit
  - " Focuses on the creation of the media for the solution that allows it to be deployed in the target environment
  - " The deployment unit contains all the deliverable items, and these are listed in the Bill of Materials.

LECTURE 10
PROJECT CLOSURE

The Project Closure Phase

- ! The last phase of the Project Life Cycle
- ! Determined by the completion of all Project Objectives and acceptance of the end product by the customer
- ! The outputs from Project Closure Phase is input to execute the next projects with much more efficiency and control



Closing Process Group

- ! Contains two processes:
  - " Project Closure: This is the process necessary to finalize all activities across all of the Process Groups to formally close the project or a project phase.
  - " Contract Closure: This is the process necessary for completing and settling each contract, including the resolution of any open items, and closing each contract applicable to the project or a project phase.

  - ! Administrative Project Closure includes:
    - " Integrated activities to collect project records

        - " Analyze project success or failure
        - " Gather lessons learned
        - " Archive project for future use

- ! includes:
    - " Product Verification - Work completed correctly and satisfactory
    - " Administrative Closure – Updating contract records to reflect final state and prepare them for archiving
    - ! Input for the Contract Closure Process

Project Closure Inputs

- ! Project Management Plan
- ! Contract documentation
- ! Enterprise environmental factors
- ! Organizational process assets
- ! Work performance information
- ! Deliverables

Inputs: Contract Documentation

- ! Used to perform the contract closure process
- ! Includes
- " The contract itself
- " Changes to the contract
- " Other documentation (such as the technical approach, product description, or deliverable acceptance criteria and procedures).

Inputs: Enterprise environmental factors

- ! Organizational or company culture and structure
- ! Governmental or industry standards
- ! Infrastructure
- ! Existing human resources
- ! Personnel administration
- ! Company work authorization system
- ! Marketplace info, stakeholder risk

Inputs: Organizational process assets

- ! Organization's processes and procedures for conducting work
    - " Standardized guidelines, work instructions, proposal evaluation criteria, and performance measurement criteria, templates…
- ! Organizational corporate knowledge base for storing and retrieving information
    - " Process measurement database, project files,…

Inputs: Work performance information

- ! Schedule progress
- ! Deliverables that have been completed and those not completed
- ! Schedule activities that have started and those that have been finished

- ! Extent to which quality standards are being met
- ! Costs authorized and incurred

Project Closure Outputs
- ! Administrative Closure Procedure
- ! Contract Closure Procedure
- ! Final Product, service or result
- ! Organizational process assets (updates)

Outputs: Administrative Closure Procedure
- ! All procedures and roles involved in administrative closure
- ! Procedures for transfer to production
    - " Action and activities for stakeholder approval
    - " Confirmation of goals achievement (sponsor, client, stakeholders, etc)
    - " Actions for satisfying all exit criteria

Outputs: Contract Closure Procedure
- ! A step-by-step methodology addressing:
    - " Terms and conditions of the contracts and any required completion or exit criteria for contract closure
    - " Activities and responsibilities of the project ALL team members involved in the contract
- closure process
- ! Actions performed to formally close all Contracts associated with the completed project

Output: Final Product, Service or Result
- ! Formal acceptance and handover of the final product, service, or result that the project was authorized To produce
- ! The acceptance includes receipt of a formal statement that the terms of the contract have been met

Output: Organizational Process Assets
- ! Formal Acceptance Documentation
- ! Project Files
- ! Project Closure Documents
- ! Historical Information

Close Process Tools
- ! Project Management Methodology
- ! Project Management Information System
- ! Expert Judgment

Iteration Closure
- ! Iteration is concluded by the iteration acceptance review, where the formal result of the iteration is agreed upon and recorded

- ! This involves considering the results of the iteration and analyzing their impact on the project as a whole

Iteration Closure: Measurement and Analysis
- ! Absolute progress
- ! Risk exposure
- ! Estimate to complete
- ! Effort profile
- ! Cost profile
- ! Effort Expended
- ! Find/fix rate (tolerance)
- ! Defect trends (tolerance)

Iteration Closure: Acceptance Review
- ! Results
- ! Risks
- ! Objectives
- ! Requirements
- ! Adherence to Plan
- ! Lessons Learned
- ! Rework, external changes, feedback from demo, other deliverables…

Iteration Grades
- ! Exceptional
- ! Passed
- ! Passed at risk
- ! Unfinished
- ! Failed
- ! Abandoned

Iteration recommendations
- ! For Unfinished, Failed or Abandoned we can have the following recommendations
    - " Re-plan
    - " De-scope
    - " Extend
    - " Try again
    - " Cancel

Acting on Iteration assessment results
- ! Never confuse the map with the journey
- ! Adopt an attitude that continuous planning is good
- ! Mature your process alongside your team
- ! Be prepared to cut your losses
- ! Be honest

Phase Assessment
- ! Differences from iteration assessments:
    - " Judge the project, not a single iteration

- " Always decide whether to continue or cancel the project
- " Focus on the delivered value rather than on performance
- " Assess project against the business case

Phase Assessment topics

- ! Progress
- ! Risk
- ! Scope
- ! Baseline
- ! Performance
- ! Plans
- ! Business Case

Project conclusions

- ! Approved
- ! Temporary approved
- ! Extended
- ! Paused
- ! Postponed
- ! Canceled

Project Assessments

- ! Performed not only at project closure
- ! Cover for:
  - " Handle exceptions raised by the iteration
  - assessments
  - " Phase assessments are too far apart
  - " Provide review points for the overall project (might include non-software subprojects)

Assessment objectives [1]

- ! Confirm that the need for the project has not changed
- ! Satisfy the stakeholders of the quality of the products delivered by the project
- ! Confirm that the business case is still viable
- ! Authorize the continuation or cancellation of the project
- ! Assess the benefits that the project has delivered
- ! Assess the overall effectiveness of the project

Project Closure Example

- ! In RUP we have the following activities:
- " Update Project Close-Out Plan and Schedule Activities
- " Schedule Final Configuration Audits
- " Conduct a Project Post-Mortem Review
- " Complete Acceptance Action Items
- " Close Out the Project