Unit 5

Lecture 1. Software Configuration Management: Software Configuration Items and tasks, Baselines,

Why Software Config Management?

- The problem:
- Multiple people have to work on software that is changing
- More than one version of the software has to be supported:
- Released systems
- Custom configured systems (different functionality)
- System(s) under development
- Software must run on different machines and operating systems

- Need for coordination
- Software Configuration Management
- manages evolving software systems
- controls the costs involved in making changes to a system

What is Software Configuration Management?

- Definition:
- A set of management disciplines within the software engineering process to develop a baseline.
- Description:
- Software Configuration Management encompasses the disciplines and techniques of initiating, evaluating and controlling change to software products during and after the software engineering process.
- Standards (approved by ANSI)
- IEEE 828: Software Configuration Management Plans
- IEEE 1042: Guide to Software Configuration Management

Software Configuration Management is a Project Function

- SCM is a Project Function (as defined in the SPMP) with the goal to make technical and managerial activities more effective.
- Software Configuration Management can be administered in several ways:
- A single software configuration management team for the whole organization
- A separate configuration management team for each project
- Software Configuration Management distributed among the project members
- Mixture of all of the above

Configuration Management Activities

- Software Configuration Management Activities:
- Configuration item identification
- Promotion management
- Release management
- Branch management
- Variant management
- Change management
- No fixed rules:

Activities are usually performed in different ways (formally, informally) depending on the project type and life-cycle phase (research, development, maintenance).

Baseline

"A specification or product that has been formally reviewed and agreed to by responsible management, that thereafter serves as the basis for further development, and can be changed only through formal change control procedures."

Examples:

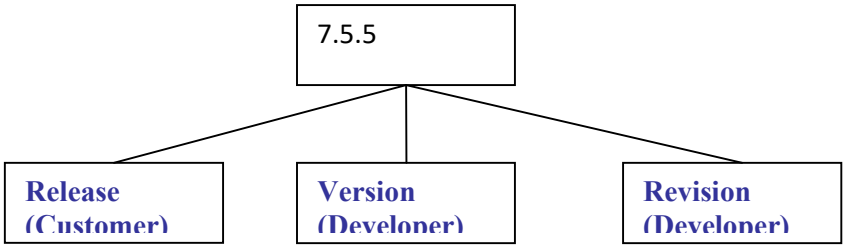
Baseline A: All t**he API have completely been defined; the bodies of the methods are empty**.

Baseline B: **All data access methods are implemented and tested**

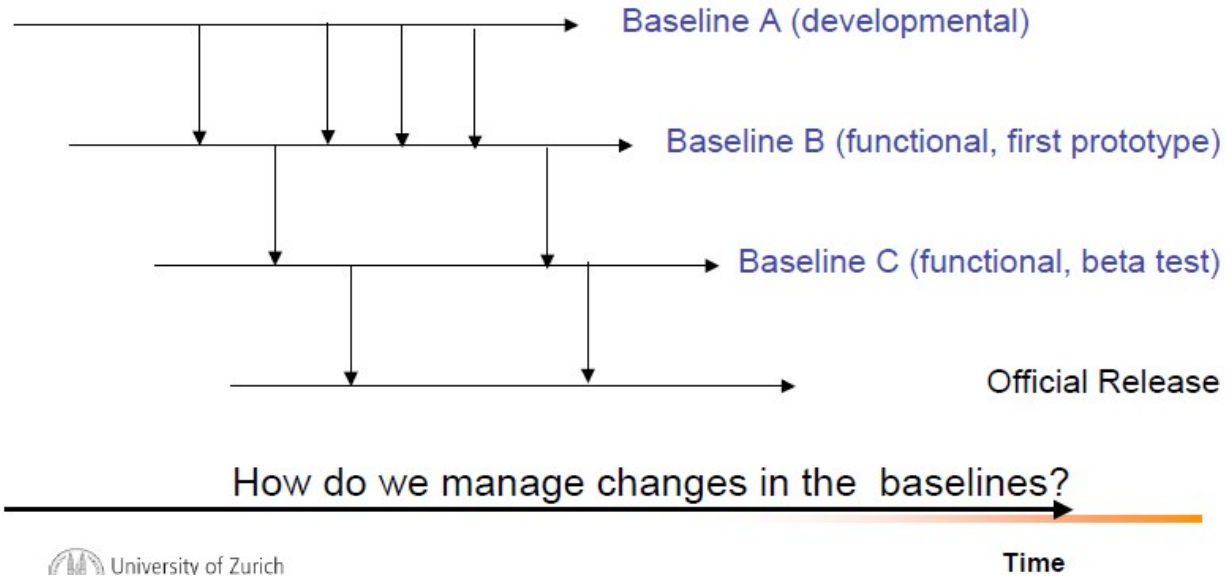
Baseline C: **The GUI is implemented**

More on Baselines

- As systems are developed, a series of baselines is developed, usually after a review (analysis review, design review, code review, system testing, client acceptance, ...)
- *Developmental baseline* (RAD, SDD, Integration Test, ...)
- Goal: Coordinate engineering activities.
- *Functional baseline* (first prototype, alpha release, beta release)
- Goal: Get first customer experiences with functional system.
- *Product baseline* (product)
- Goal: Coordinate sales and customer support.
- Many naming scheme for baselines exist (1.0, 6.01a, ...)
- A 3 digit scheme is quite common:



Baselines in SCM:



Lecture 2: Plan for Change, Change Control, Change Requests Management,

Change management

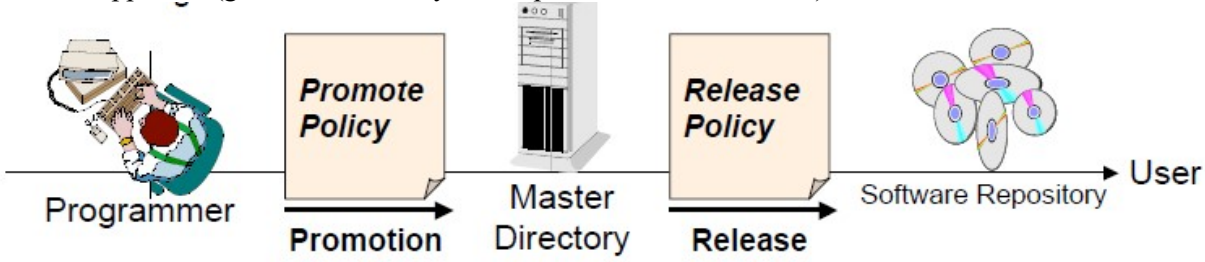
- Change management is the handling of change requests
- A change request leads to the creation of a new release
- General change process
- The change is requested (this can be done by anyone including users and developers)
- The change request is assessed against project goals
- Following the assessment, the change is accepted or rejected
- If it is accepted, the change is assigned to a developer and implemented
- The implemented change is audited.
- The complexity of the change management process varies with the project.

Small projects can perform change requests informally and fast while complex projects require detailed change request forms and the official approval by one more managers

Controlling Changes

Two types of controlling change:

- *Promotion:* The internal development state of a software is changed.
- *Release:* A changed software system is made visible outside the development organization.
- Approaches for controlling change (Change Policy)
- Informal (good for research type environments and promotions)
- Formal approach (good for externally developed CIs and for releases)



Configuration Control

- Defines the following steps
- 3.2.1 How to identify the need for a change (layout of change request form)
- 3.2.2 Analysis and evaluation of a change request
- 3.2.3 Approval or disapproval of a request
- 3.2.4 Verification, implementation and release of a change

Change Request

- Specifies the procedures for requesting a change to a baselined CI and the information to be documented:
- Name(s) and version(s) of the CI(s) where the problem appears
- Originator's name and address
- Date of request
- Indication of urgency
- The need for the change
- Description of the requested change

Evaluation of a Change

- Specifies the analysis required to determine the impact of proposed changes and the procedure for reviewing the results of the analysis.

Change Approval or Disapproval

- This section of the SCMP describes the organiztion of the configuration control board (CCB).
- Configuration Control Board (CCB)
- Can be an individual or a group.
- Multiple levels of CCBs are also possible, depending on the complexity of the project
- Multiple levels of CCBs may be specified.
- In small development efforts one CCB level is sufficient.
- This section of the SCMP also indicates the level of authority of the CCB and its responsibility.
- In particular, the SCMP must specify when the CCB is invoked.

Implementing Change

- This section of the SCMP specifies the activities for verifying and implementing an approved change.
- A completed change request must contain the following information:
- The original change request(s)
- The names and versions of the affected configuration items
- Verification date and responsible party
- Identifier of the new version
- Release or installation date and responsible party
- This section must also specify activities for
- Archiving completed change requests
- Planning and control of releases
- How to coordinate multiple changes
- How to add new CIs to the configuration
- How to deliver a new baseline

Lecture 3: Version Control, Risk Management: Risks and risk types, Risk Breakdown Structure (RBS),

Version vs. Revision vs. Release

Version
- An initial release or re-release of a configuration item associated with a complete compilation or recompilation of the item. Different versions have different functionality.
- Revision
- Change to a version that corrects only errors in the design/code, but does not affect the documented functionality.
- Release
- The formal distribution of an approved version

Risk Management:

Project Risk Management Objectives

The Project Risk Management Handbook has been designed to:

- Be simple and easy to use
- Be scalable to project size and complexity
- Pull communication of risks across project milestones and phases
- Actively manage risk to enhance project success
- Integrate into the current project delivery process, and
- Involve all functional units in the management of risks.

Project Risk Management Values

Identifying, communicating, and managing project risks requires a *risk management* culture. This culture is defined by the *values* in which we operate. The following attributes depict PRM values required for the development of a successful risk management culture.

- Risk decision-making based on balancing project values such as cost, schedule, and quality
- Stewardship
- Efficiency
- Teamwork
- Joint ownership of risks and responsibilities
- Accountability

Risk and issue are two words that are often confused when it comes to their usage. Actually there is some difference between them.A risk is an uncertain event that has a probability associated with it. An issuedoes not have this attribute. Issues are problems right now that the project team has to do something about.Think of risk management as a proactive activity, while issue management is reactive

The Project Risk Management Process

The Basic Process

All approaches to project risk management strive to maximize both efficiency and effectiveness. Although the details of risk processes may differ depending on the project, risk management has three important parts: identification, analysis, and action. Before risk can be properly managed, it

must first be identified, described, understood, and assessed. Analysis is a necessary step, but it is not sufficient; it must be followed by action. A risk process which does not lead to implementation of actions to deal with identified risks is incomplete and useless. The ultimate aim is to manage risk, not simply to analyze it.

FIGURE 1 – PROJECT RISK MANAGEMENT PROCESS

The project risk management process (Figure 1) is not difficult. It simply offers a structured way to think about risk and how to deal with it. A full project risk management endeavor includes these processes:

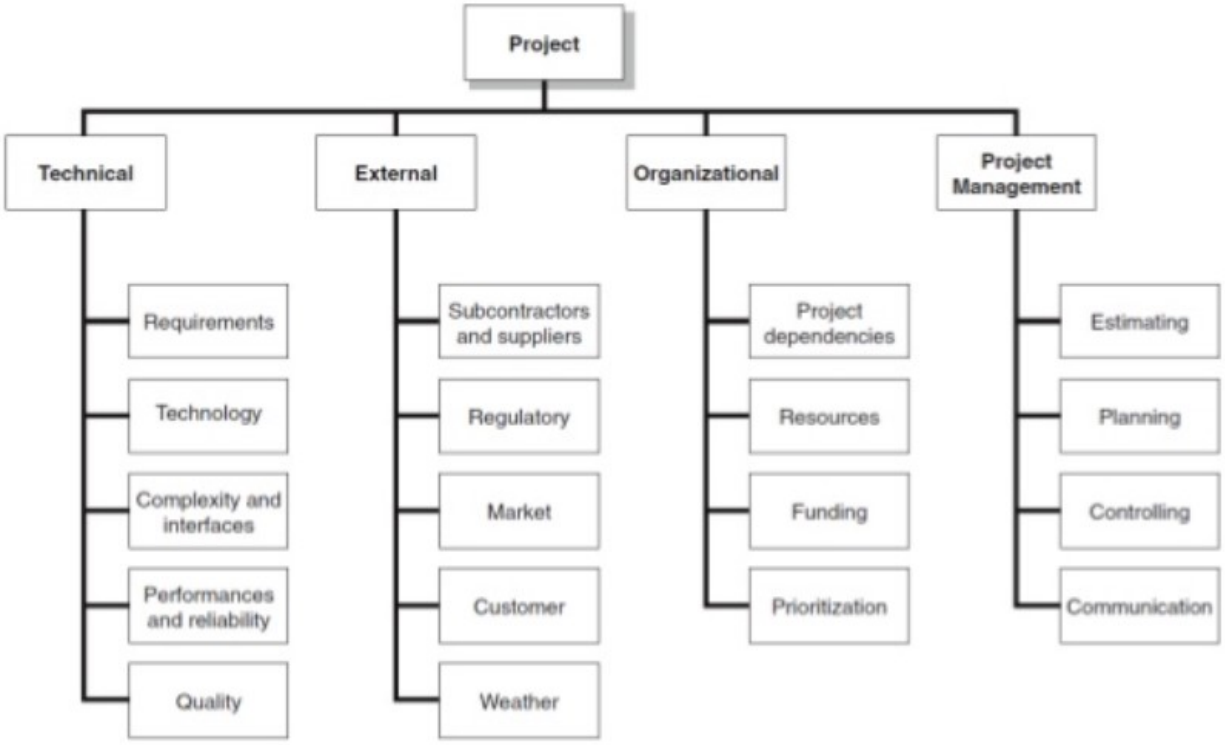
1. *Risk Management Planning* – Deciding how to approach, plan, and execute the risk management activities for a project.
2. *Risk Identification* – Determining which risks might affect the project and documenting their characteristics.
3. *Qualitative Risk Analysis* – Prioritizing risks for subsequent further analysis or action by assessing and combining their probability of occurrence and impact.
4. *Quantitative Risk Analysis* – Analyzing probabilistically the effect of identified risks on overall project objectives.
5. *Risk Response* – Developing options and actions to enhance opportunities and to reduce threats to project objectives.
6. *Risk Monitoring* – Tracking identified risks, monitoring residual risks, identifying new risks, executing risk response plans, and evaluating their effectiveness throughout the project life cycle.

At its foundation, project risk management involves asking and answering a few simple questions:

- What risks might negatively (threats) or positively (opportunities) affect achieving the project objectives? (***Risk identification***)
- Which of these are most important? (***Qualitative risk analysis***)

- How could these affect the overall outcome of the project in probabilistic terms of cost and schedule? (***Quantitative risk analysis***)
- What can be done about it? (***Risk response***)
- Having taken action, how did the responses effect change, and where is the project now? (***Risk monitoring***)
- Who needs to know about this? (***Communication***)

Risk Breakdown Structure (RBS):



Risk Identification

Risk identification determines what might happen that could affect the objectives of the project and how those things might happen. It produces a deliverable — the project risk register – that documents the risks and their characteristics. The risk register is subsequently amended by the qualitative or quantitative risk analysis, risk response, and risk monitoring processes. Risk identification is an iterative process because new risks may become known as the project progresses through its life cycle, previously-identified risks may drop out, and other risks may be updated.

"Risk" Includes Threats and Opportunities

The concept of risk can include positive and negative impacts. This means that the word "risk" can be used to describe uncertainties that, if they occurred, would have a negative or harmful effect. The same word can also describe uncertainties that, if they occurred, would be helpful. In short, there are two sides to risk: threats and opportunities. Projects in design have the greatest potential for opportunities because the project is still open to changes. Risk reduction and avoidance are opportunities, as are value analyses, constructability reviews, and innovations in design,

construction methods, and materials. Once a project enters construction, the project objectives (scope, time, and cost) are fixed contractually, so opportunities to save money and time are fewer. Any changes must be made using a contract change order (CCO), and only a negative CCO such as one resulting from a Value Engineering Change Proposal by the contractor would still afford an opportunity to save money and time. Otherwise, CCOs add cost and/or time to the project. So, the risk management focus during construction is on reducing or eliminating risks.

The Risk Register

What it is

A risk register is a tool that project teams can use to address and document project risks throughout the project life cycle. It is a living document – a comprehensive listing of risks and the manner in which they are being addressed as part of the project risk management process. The risk register is maintained as part of the project file that also includes information related to uncertainties in the cost estimate and schedule.

Why use it

A new project team is formed for every project and disbanded when the project is complete. Although not desirable, project team members sometimes change, and the project experiences change over the course of the project. Communication among project team members about the project objectives, costs, risks, etc., is vital. The risk register communicates project risks and helps the team members understand the status of the risks as a project moves from inception toward completion. Managers should view the risk register as a management tool through a review and updating process that identifies, assesses, manages, and reduces risks to acceptable levels.

How to use it

A risk register is best used as a living document throughout the project's entire life cycle, from PID through construction, to record the evolution of project risks. There is no prescription for how extensive a project's risk register should be. The project team decides the most beneficial use of the risk register, with the objective of minimizing the risk impact.

The team members identify the potential risks (threats and opportunities) using any combination of:

- Brainstorming,
- Challenging of assumptions,
- Looking for "newness" (e.g. new materials, technology, or processes),
- Their knowledge of the project or similar projects,
- Consultation with others who have significant knowledge of the project or its environment,
- Consultation with others who have significant knowledge of similar projects, and
- The experience of project stakeholders or others in the organization

Qualitative Risk Analysis

Qualitative risk analysis includes methods for prioritizing the identified risks for further action, such as risk response. The PRMT can improve the project's performance effectively by focusing on high-priority risks. Team members revisit qualitative risk analysis during the project's lifecycle. When the team repeats qualitative analysis for individual risks, trends may emerge in the results.

These trends can indicate the need for more or less risk management action on particular risks or even show whether a risk mitigation plan is working.

Planning Project Risk Management

1. Creating the Project's Risk Management Plan

The Risk Management Plan (RMP) defines the level at which risk management will be performed for the project and the frequency of risk management meetings and risk register updates. It lists the members of the Project Risk Management Team by the various disciplines involved in the project and sets a budget for the risk management activities. The RMP should be completed early in project planning, since it is crucial to successfully performing the other processes described herein.

steps:

- Determine the scalability level for the project.
- Download the risk register for the scalability level from: http://onramp/riskmanagement.
- Determine the frequency of risk meetings for the project and the applicable communication and accountability checkpoints.
- Decide who will be on the Project Risk Management Team.
- If significant effort or outside consultants will be involved, include estimates for project risk management activities in work plans.
- If applicable, obtain the necessary approvals for the written RMP.

2.The Project Risk Management Team

The Project Risk Management Team (PRMT) is the core group performing, updating, and reviewing risk management activities under the direction of the project risk manager. The PRMT will include members of the PDT, but not necessarily all members.

3. Incorporating Project Risk Management Activities into the Project Schedule

The project schedule (work plan) should incorporate the following:

- Dates for project risk management meetings
- Time to allow team members to prepare for review of the risk register and risk responses
- Milestones for communication and accountability checkpoints

4. The First Project Risk Management Meeting

The first time that the PRMT meets, the project manager should brief the team about the following:

- The importance and objectives of the project risk management process
- The process itself
- The roles and responsibilities
- The risk register

- The communication and accountability check points
- Risk management activities in the project schedule
- Time charge codes for risk management activities
- The expectation that risk will be managed, documented, and reported

At this first meeting, elicit risks from the team members. If working to Level 2 scalability, determine the impact and probability definitions so that the team has the same understanding of the meaning of the word descriptions.

Risk Monitoring

Continuous monitoring by the project risk manager and the project team ensures that new and changing risks are detected and managed and that risk response actions are implemented and effective. Risk monitoring continues for the life of the project. Risk monitoring and control keeps track of the identified risks, residual risks, and new risks. It also monitors the execution of planned strategies for the
identified risks and evaluates their effectiveness. Risk monitoring and control continues for the life of the project. The list of project risks changes as the project matures, new risks develop, or anticipated risks disappear. Risk ratings and prioritizations can also change during the project lifecycle. Typically, during project execution, risk meetings should be held regularly to update the status of risks in the risk register, and add new risks. Periodic project risk reviews repeat the process of identification, analysis, and response planning. If an unanticipated risk emerges, or a risk's impact is greater than expected, the planned response may not be adequate. The project manager and the PRMT should perform additional responses to control the risk. Monitoring also determines whether:

- The PRMT is performing periodic risk review and updating
- Risk management policies and procedures are being followed
- The remaining contingency reserves for cost and schedule are adequate

And it may involve recommending:

- Alternative risk responses
- Implementing a contingency plan
- Taking corrective actions
- Changing the project objectives

Risk Review and Updating

Periodically, the PRMT will convene to review the project's risk register and risk response actions, and to update project risk information.
The review tasks of the PRMT include the following:

- Identify, analyze, and plan response actions for newly arising risks, and add them to the risk register.

- Review the execution of risk response actions, and evaluate their effectiveness.
- Re-assess existing risks, verify that the assumptions are still valid, and modify the previous assessments as necessary.
- Assign additional risk response actions to the Risk Owner.
- Retire risks whose opportunity to impact the project has elapsed, or whose residual impact on the project is deemed to have reached an acceptable level.

The PRMT should discuss any risks for which response actions are not being carried out effectively or whose risk impact is increasing. If these cannot be resolved within the PRMT, they should be escalated to the project manager with recommendations for action

Updating the Risk Register

Make any changes and additions to the risks and enter the revision date into the "Updated" column.

Lessons Learned

When a risk is retired, the PRMT will review the history of the risk to record any lessons learned regarding the risk management processes used. The team is essentially asking itself: "What, if anything, would we have done differently and why?" The project risk manager will conduct a periodic review of all lessons learned with the PRMT.

Cost Benefit Analysis:

Cost-bene t analysis The comparison of costs and bene ts of public goods projects to decide if they should be undertaken.

Measuring the Costs of Public Projects:

Example-

■ TABLE 8-1

Cost-Benefit Analysis of Highway Construction Project

		Quantity	Price / Value	Total
Costs	Asphalt	1 million bags		
	Labor	1 million hours		
	Maintenance	$10 million/year		
			First-year cost:	
			Total cost over time:	
Benefits	Driving time saved	500,000 hours/year		
	Lives saved	5 lives/year		
			First-year benefit:	
			Total benefit over time:	
			Benefit over time minus cost over time:	

The renovation of the turnpike in your state has three costs: asphalt, labor, and future maintenance. There are two associated benefits: reduced travel time and reduced fatalities. The goal of cost-benefit analysis is to quantify these costs and benefits.

MEASURING CURRENT COSTS

Cash-flow accounting:

Accounting method that calculates costs solely by adding up what the government pays for inputs to a project, and calculates bene ts solely by adding up income or government revenues generated by the project.

Opportunity cost:

The social marginal cost of any resource is the value of that resource in its next best use.

General rule: Economic costs are only those costs associated with diverting the resource from its next best use

Perfectly Competitive Markets

Social Cost = Price (true for labor and material)

Imperfectly Competitive Markets

A. Monopoly: (suppose asphalt is produced by monopoly)

Price = Marginal cost + Monopoly Marginal Profit > Marginal cost

On efficiency grounds, Social cost = Marginal cost

Profit is a transfer from govt (taxpayers) to monopoly (this matters for redistribution but not efficiency)

B. Labor market with unemployment : Suppose a minimum wage set at $10 creates involuntary unemployment The unemployed would be willing to work for $6 on average but cannot find jobs Govt provides jobs paying $10/hour.

Social Cost = $6 = $10 (wage) - $4 (surplus value of jobs for workers)

MEASURING FUTURE COSTS

Present discounted value (PDV): A dollar next year is worth 1+r times less than a dollar now because the dollar could earn r in interest if invested.

Government uses public debt (Treasury Bills) with interest r to borrow (example: r = 6% nominal, r-ination=3%)

Social discount rate: The appropriate value of r to use in computing PDV for social investments.

Problematic predictions for the long-run: r=3%) $100 in 100 years = 1=(1 + r)100 = $5.2 today) Long-run costs (such as global warming) are heavily discounted

Measuring the Costs of Public Projects

Measuring Future Costs

■ TABLE 8-2

Cost-Benefit Analysis of Highway Construction Project

		Quantity	Price / Value	Total
Costs	Asphalt	1 million bags	$100/bag	$100 million
	Labor	1 million hours	½ at $20/hour and ½ at $10/hour	$15 million
	Maintenance	$10 million/year	7% discount rate	$143 million
			First-year cost:	**$115 million**
			Total cost over time (7% discount rate):	**$258 million**
Benefits	Driving time saved	500,000 hours/year		
	Lives saved	5 lives/year		
			First-year benefit:	
			Total benefit over time:	
			Benefit over time minus cost over time:	

The cost of the asphalt for this project is dictated by the market price for asphalt, $100 per bag. The cost of labor depends not on the wage but on the full opportunity cost of the labor, which incorporates the current unemployment of any workers who will be used on the project. The cost of future maintenance is the present discounted value of these projected expenditures.

Measuring the Benefits of Public Projects
Valuing Saved Lives
Government Revealed Preference

■ TABLE 8-3

Costs Per Life Saved of Various Regulations

Regulation concerning . . .	Year	Agency	Cost Per Life Saved (millions of 2005 $)
Childproof lighters	1993	CPSC	$0.1
Food labeling	1993	FDA	0.4
Reflective devices for heavy trucks	1999	NHTSA	1.0
Children's sleepwear flammability	1973	CPSC	2.4
Rear/up/shoulder seatbelts in cars	1989	NHTSA	4.8
Asbestos	1972	OSHA	6.0
VALUE OF STATISTICAL LIFE			**7.6**
Benezene	1987	OSHA	24
Asbestos ban	1989	EPA	85
Cattle feed	1979	FDA	185
Solid waste disposal facilities	1991	EPA	109,000

Morrall (2003), Table 2, updated to 2005 dollars.

Government safety regulations increase costs and save lives, and these costs and benefits can be compared to compute an implicit cost per life saved. These values range from a low of $110,000 per life saved for childproof lighters to a high of over $109 billion per life saved for solid waste disposal facility regulations.

Putting It All Together

■ TABLE 8-4

Cost-Benefit Analysis of Highway Construction Project

		Quantity	Price / Value	Total
Costs	Asphalt	1 million bags	$100/bag	$100 million
	Labor	1 million hours	½ at $20/hour and ½ at $10/hour	$15 million
	Maintenance	$10 million/year	7% discount rate	$143 million
			First-year cost:	**$115 million**
			Total cost over time (7% discount rate):	**$258 million**
Benefits	Driving time saved	500,000 hours/year	$17/hour	$8.5 million
	Lives saved	5 lives/year	$7 million/life	$35 million
			First-year benefit:	**$43.5 million**
			Total benefit over time (7% discount rate):	**$621.4 million**
			Benefit over time minus cost over time:	**$363.4 million**

The time savings from this project is most appropriately valued by the revealed preference valuation of time, which is $17/hour. The life savings is most appropriately valued by the revealed preference value of life, which averages $7 million. The present discounted value of costs for this renovation project is $258 million, while the *PDV* of benefits for this project is $621.4 million. Because benefits exceed costs by $363.4 million, the project should clearly be undertaken.

OTHER ISSUES IN COST-BENEFIT ANALYSIS

Common Counting Mistakes: When analyzing costs and benefits, a number of common mistakes arise, such as:

-Counting secondary bene_ts (e.g., more commerce activity around new highway comes at the expense of other places)

-Counting labor as a bene_t (e.g., labor is a cost, jobs created means those workers do not produce something else)

-Double-counting bene_ts (e.g., rise in house values due to reduced com-muting cost)

Distributional Concerns: The costs and bene_ts of a public project do not necessarily accrue to the same individuals.

Uncertainty: The costs and bene_ts of public projects are often highly uncertain.

CONCLUSION

Government analysts at all levels face a major challenge in attempting to turn the abstract notions of social costs and benfits into practical implications for public project choice.

What at _rst seems to be a simple accounting exercise be-comes quite complicated when resources cannot be valued in competitive markets.

Nevertheless, economists have developed a set of tools that can take analysts a long way toward a complete accounting of the costs and bene_ts of public projects.

Lecture 4: Software Project Management Tools: CASE Tools, Planning and Scheduling Tools, MS-Project.

CASE stands for **C**omputer **A**ided **S**oftware **E**ngineering. It means, development and maintenance of software projects with help of various automated software tools.

CASE Tools

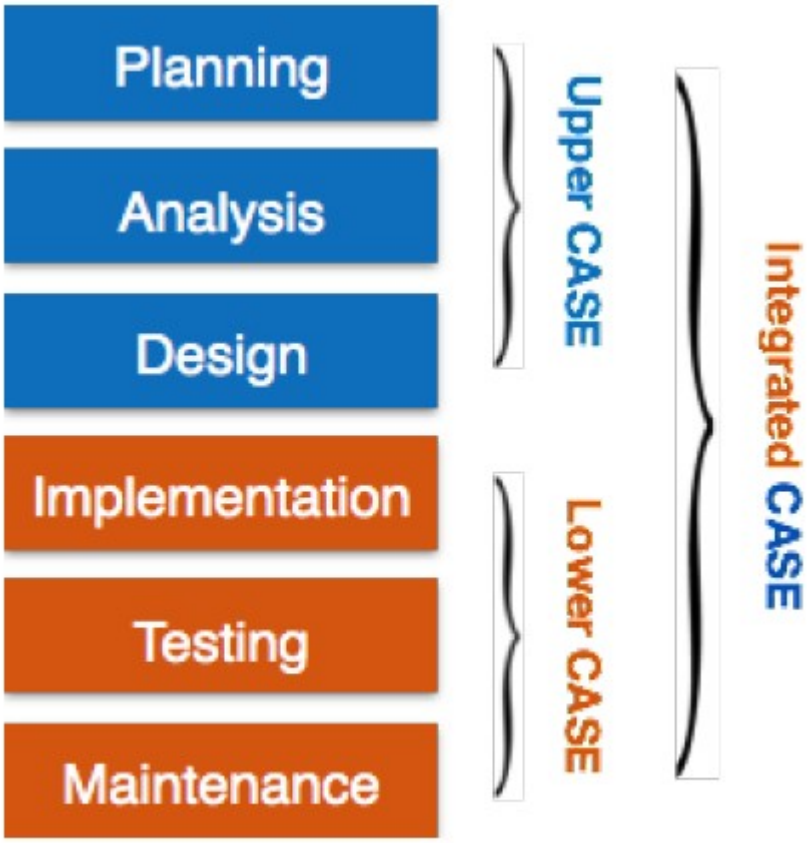
CASE tools are set of software application programs, which are used to automate SDLC activities. CASE tools are used by software project managers, analysts and engineers to develop software system.

There are number of CASE tools available to simplify various stages of Software Development Life Cycle such as Analysis tools, Design tools, Project management tools, Database Management tools, Documentation tools are to name a few. Use of CASE tools accelerates the development of project to produce desired result and helps to uncover flaws before moving ahead with next stage in software development.

Components of CASE Tools

CASE tools can be broadly divided into the following parts based on their use at a particular SDLC stage:

Central Repository - CASE tools require a central repository, which can serve as a source of common, integrated and consistent information. Central repository is a central place of storage where product specifications, requirement documents, related reports and diagrams, other useful information regarding management is stored. Central repository also serves as data dictionary.



Upper Case Tools - Upper CASE tools are used in planning, analysis and design stages of SDLC.

Lower Case Tools - Lower CASE tools are used in implementation, testing and maintenance.

Integrated Case Tools - Integrated CASE tools are helpful in all the stages of SDLC, from Requirement gathering to Testing and documentation. CASE tools can be grouped together if they have similar functionality, process activities and capability of getting integrated with other tools.

Scope of Case Tools

The scopeof CASE tools goes throughout the SDLC.

Case Tools Types

Now we briefly go through various CASE tools

Diagram tools

These tools are used to represent system components, data and control flow among various software components and system structure in a graphical form. For example, Flow Chart Maker tool for creating state-of-the-art flowcharts.

Process Modeling Tools

Process modeling is method to create software process model, which is used to develop the software. Process modeling tools help the managers to choose a process model or modify it as per the requirement of software product. For example, EPF Composer

Project Management Tools

These tools are used for project planning, cost and effort estimation, project scheduling and resource planning. Managers have to strictly comply project execution with every mentioned step in software project management. Project management tools help in storing and sharing project information in real-time throughout the organization. For example, Creative Pro Office, Trac Project, Basecamp.

Documentation Tools

Documentation in a software project starts prior to the software process, goes throughout all phases of SDLC and after the completion of the project. Documentation tools generate documents for technical users and end users. Technical users are mostly in-house professionals of the development team who refer to system manual, reference manual, training manual, installation manuals etc. The end user documents describe the functioning and how-to of the system such as user manual. For example, Doxygen, DrExplain, Adobe RoboHelp for documentation.

Analysis Tools

These tools help to gather requirements, automatically check for any inconsistency, inaccuracy in the diagrams, data redundancies or erroneous omissions. For example, Accept 360, Accompa, CaseComplete for requirement analysis, Visible Analyst for total analysis.

Design Tools

These tools help software designers to design the block structure of the software, which may further be broken down in smaller modules using refinement techniques. These tools provides detailing of each module and interconnections among modules. For example, Animated Software Design

Configuration Management Tools

An instance of software is released under one version. Configuration Management tools deal with –
Version and revision management Baseline configuration management Change control management
CASE tools help in this by automatic tracking, version management and release management. For example, Fossil, Git, Accu REV.

Change Control Tools

These tools are considered as a part of configuration management tools. They deal with changes made to the software after its baseline is fixed or when the software is first released. CASE tools automate change tracking, file management, code management and more. It also helps in enforcing change policy of the organization.

Programming Tools

These tools consist of programming environments like IDE (Integrated Development Environment), in-built modules library and simulation tools. These tools provide comprehensive aid in building software product and include features for simulation and testing. For example, Cscope to search code in C, Eclipse.

Prototyping Tools

Software prototype is simulated version of the intended software product. Prototype provides initial look and feel of the product and simulates few aspect of actual product. Prototyping CASE tools essentially come with graphical libraries. They can create hardware independent user interfaces and design. These

tools help us to build rapid prototypes based on existing information. In addition, they provide simulation of software prototype. For example, Serena prototype composer, Mockup Builder.

Web Development Tools

These tools assist in designing web pages with all allied elements like forms, text, script, graphic and so on. Web tools also provide live preview of what is being developed and how will it look after completion. For example, Fontello, Adobe Edge Inspect, Foundation 3, Brackets.

Quality Assurance Tools

Quality assurance in a software organization is monitoring the engineering process and methods adopted to develop the software product in order to ensure conformance of quality as per organization standards. QA tools consist of configuration and change control tools and software testing tools. For example, SoapTest, AppsWatch, JMeter.

Maintenance Tools

Software maintenance includes modifications in the software product after it is delivered. Automatic logging and error reporting techniques, automatic error ticket generation and root cause Analysis are few CASE tools, which help software organization in maintenance phase of SDLC. For example, Bugzilla for defect tracking, HP Quality Center.