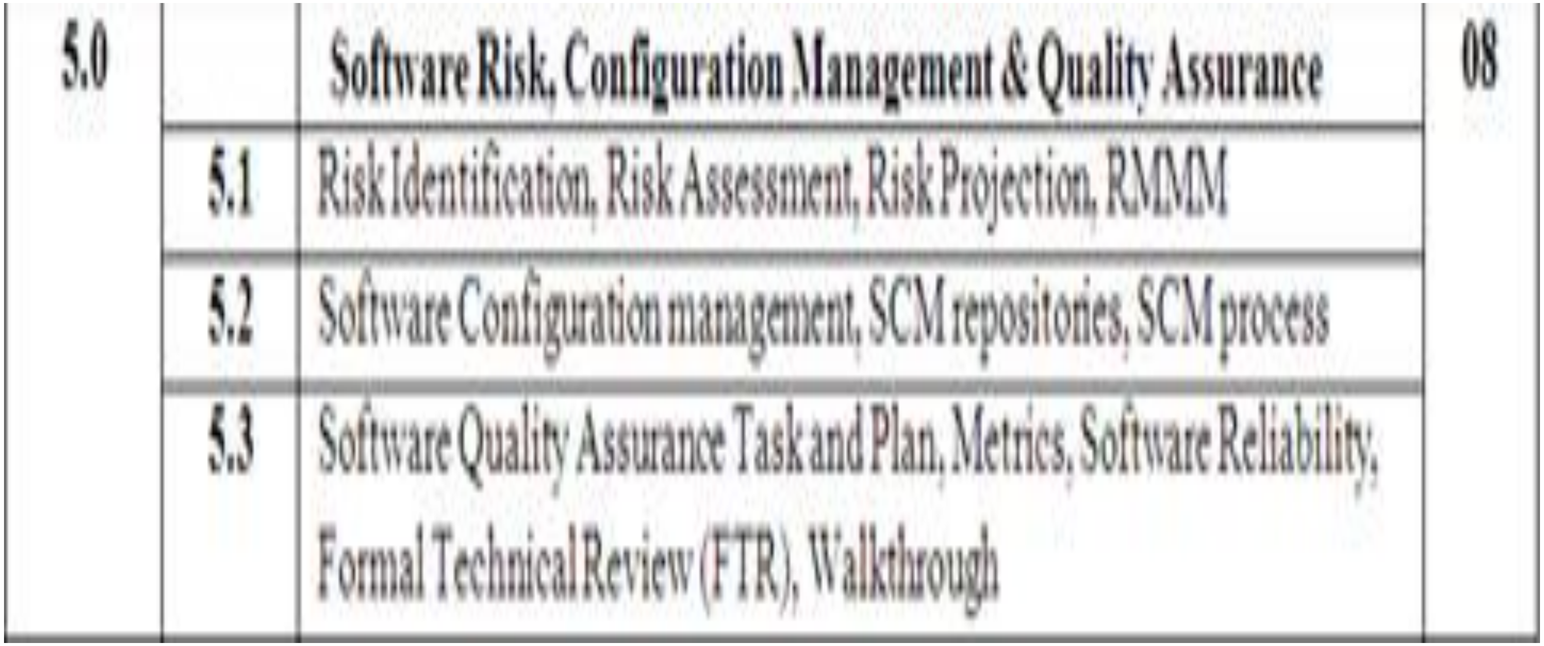
Software Engineering(SE)

CSC 601

Subject Incharge

Varsha Nagpurkar
Assistant Professor
Room No. 407
email: varshanagpurkar@sfit.ac.in

Module-5 Syllabus



5.0		**Software Risk, Configuration Management & Quality Assurance**	08
	5.1	Risk Identification, Risk Assessment, Risk Projection, RMMM	
	5.2	Software Configuration management, SCM repositories, SCM process	
	5.3	Software Quality Assurance Task and Plan, Metrics, Software Reliability, Formal Technical Review (FTR), Walkthrough	

Software Risks

- Whenever we start any business or any development process,we take into consideration the risks involved in accomplishing that task
- Similarly when software development process is started,it is main job of a software manager to look into all types of possible risks involved in the entire process
- It involves focusing on the possible risks that could affect the project development process
  - Schedule of the development process
  - Quality of the application under construction

Software Risks

- If the risk management is effective,then it becomes easier to handle all the problems and
- It is ensured that the project schedules and budget are within acceptable limits and there is no schedule slippage and ultimately no budget slippage

Software risks characteristics

- Uncertainty-the risk may or may not happen;that is,there are no 100% risks
- Loss-if the risk becomes a reality,unwanted consequences or losses will occur

Categories of risks

- Project risks
- Technical risks
- Business risks

Project risks-

- It threaten the project plan
- If project risks become real,it is likely that project schedule will slip and costs will increase
- Project risks identify potential budgetary,schedule,personnel(staffing and organization),resource,stakeholder,and requirements problems

Technical risks

- It threaten the quality and timeliness of the software to be produced
- If a technical risk becomes a reality,implementation may become difficult or impossible
- It identify potential design,implementation,interface,verification, and maintenance problems

Business risks

- Business risks threaten the viability of the software to be built
- Business risks often jeopardize(causes harm/loss) the project or the product
- Top 5 business risks are
  - Building an excellent product or system that no one really wants(market risks)
  - Building a product that no longer fits into the overall business strategy for the company(strategic risks)
  - Building a product that the sales force doesn't understand how to sell(sales risk)
  - Losing the support of senior management due to a change in focus or a change in people(management risks)
  - Losing budgetary or personnel commitment(budget risks)

General categorization of risks proposed by Charette

- Known risks- Are those that can be uncovered after careful evaluation of the project plan, the business and technical environment in which the project is being developed,and other reliable information sources(e.g.unrealistic delivery date,lack of documented requirements or software scope,poor development environment)

General categorization of risks proposed by Charette

- Predictable risks-Are extrapolated from past project experience(e.g.,staff turnover,poor communication with the customer,dilution of staff effort as ongoing maintenance requests are serviced
- Unpredictable risks-Can and do occur,but are extremely difficult to identify in advance

Risk Strategies

- Reactive
- Proactive

Reactive Risk Strategy

- It monitors the project for likely risks
- Resources are monitored to guess the possible risks and deal with them so that they do not become the problems for budget slippage and cost slippage
- Software team does nothing about risks until something goes wrong
- Then the team flies into action in an attempt to correct the problem rapidly
- This is often called a fire-fighting mode
- After failure of all these actions,the concept of “crisis management” comes into picture and the project is in deep trouble and uncertainty happens

Proactive Risk Strategy

- It is initiated when the actual technical work begins
- Potential risks are identified, and the probability and impact are assessed, so that the development team members establish an appropriate plan to manage all these risks
- The main goal to avoid the risks initially
- But all the risks cannot be avoided,therefore the development team creates a contigency plan that will control the risks in an effective manner

Risk identification

- It is a systematic attempt to specify threats to the project plan(estimates,schedule,resource loading,etc)
- By identifying known and predictable risks,the project manager takes a first step toward avoiding them when possible and controlling them when necessary

Distinct types of risks for each of the categories

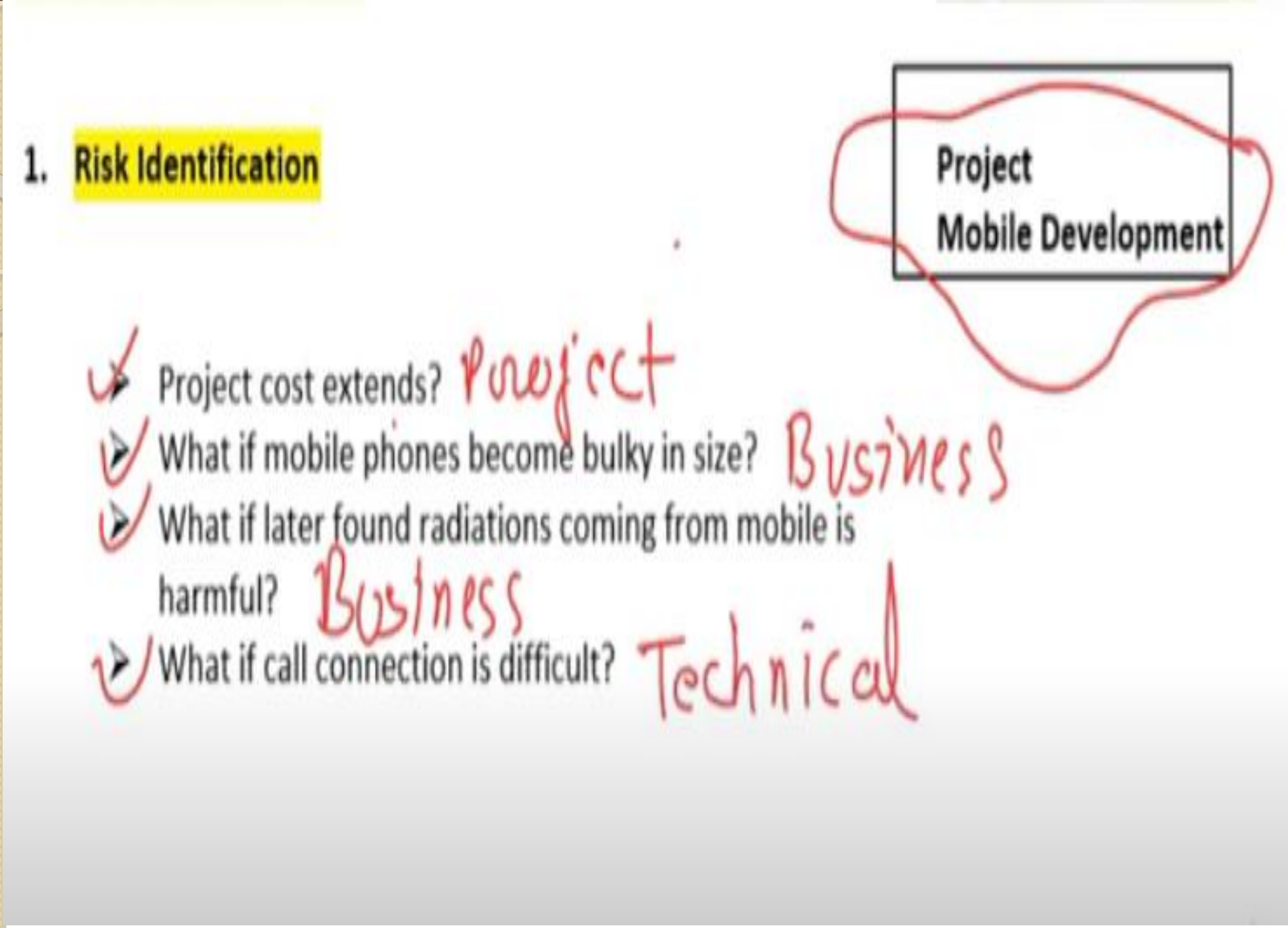
- Generic risks- Are a potential threat to every software project
- Product-specific risks-Can be identified only by those with a clear understanding of the technology,the people,and the environment that is specific to the software that is to be built
- One method for identifying risks is to create a risk item checklist
- Checklist is used for risk identification and focuses on some subset of known and predictable risks

Checklist for known and predictable risks generic subcategories

- Product size-risks associated with the overall size of the software to be built or modified
- Business impact-risks associated with constraints imposed by management or the marketplace
- Customer characteristics-risks associated with the sophistication of the customer and the developers ability to communicate with the customer in a timely manner
- Staff size and experience-risks associated with the overall technical and project experience of the software engineers who will do the work

Checklist for known and predictable risks generic subcategories

- Process definition-risks associated with the degree to which the software process has been defined and is followed by the development organization
- Development environment-risks associated with the availability and quality of the tools to be used to build the product
- Technology to be built-risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system



1. **Risk Identification**

Project
Mobile Development

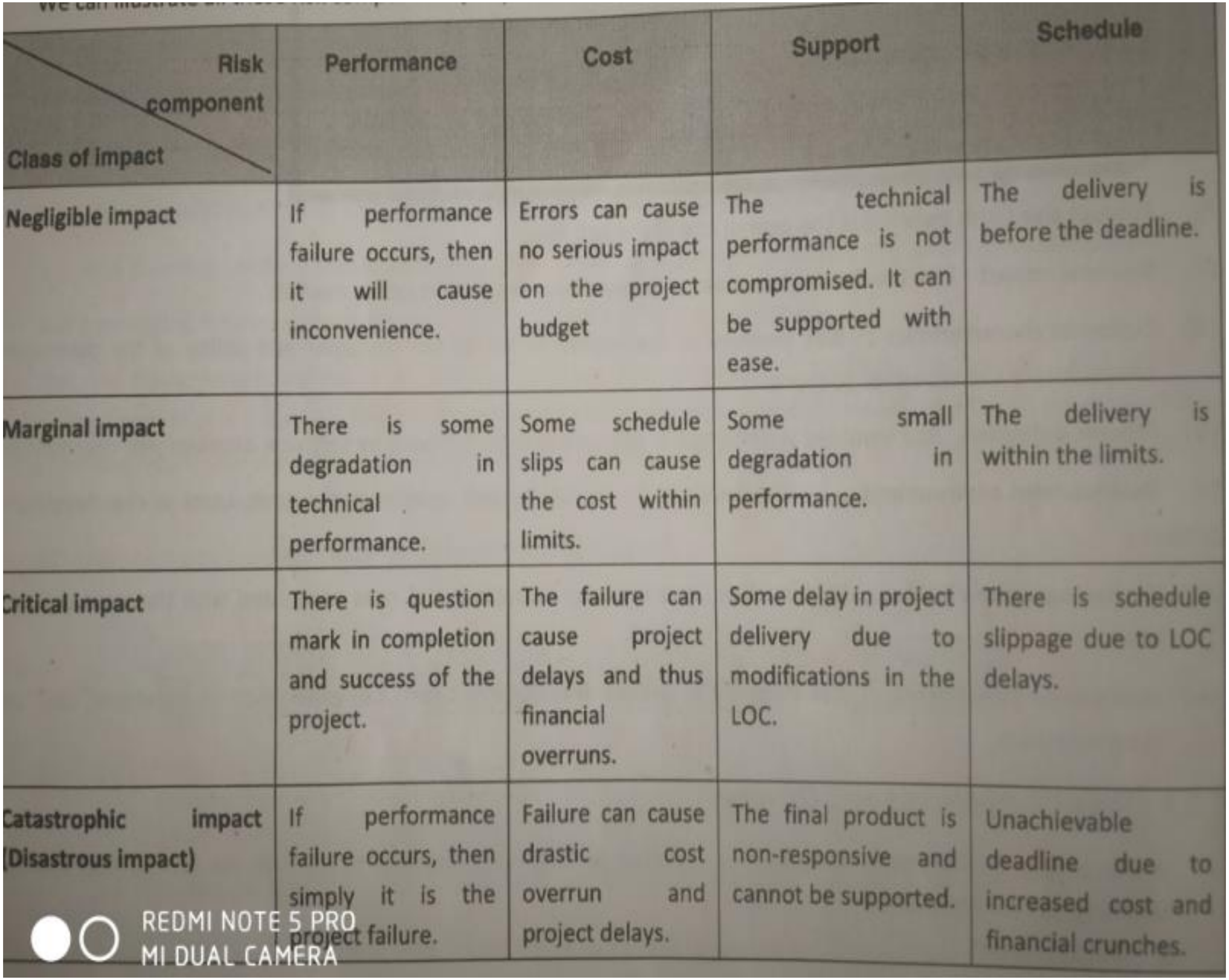
- Project cost extends? Project
- What if mobile phones become bulky in size? Business
- What if later found radiations coming from mobile is harmful? Business
- What if call connection is difficult? Technical

Risk Assessment-Risk Components and Drivers

- The factors that affect the risk components are also called as risk drivers
- Performance risk-The degree of uncertainty that the product will meet its requirements and be fit for its intended use
- Cost risk-The degree of uncertainty that the project budget will be maintained
- Support risk-The degree of uncertainty that the resultant software will be easy to correct,adapt and enhance
- Schedule risk-The degree of uncertainty that the project schedule will be maintained and that the product will be delivered on time

Impact Categories

- The impact of each risk driver on the risk component is divided into
  - Negligible
  - Marginal
  - Critical
  - Catastrophic



Risk component / Class of impact	Performance	Cost	Support	Schedule
Negligible impact	If performance failure occurs, then it will cause inconvenience.	Errors can cause no serious impact on the project budget	The technical performance is not compromised. It can be supported with ease.	The delivery is before the deadline.
Marginal impact	There is some degradation in technical performance.	Some schedule slips can cause the cost within limits.	Some small degradation in performance.	The delivery is within the limits.
Critical impact	There is question mark in completion and success of the project.	The failure can cause project delays and thus financial overruns.	Some delay in project delivery due to modifications in the LOC.	There is schedule slippage due to LOC delays.
Catastrophic impact (Disastrous impact)	If performance failure occurs, then simply it is the project failure.	Failure can cause drastic cost overrun and project delays.	The final product is non-responsive and cannot be supported.	Unachievable deadline due to increased cost and financial crunches.

Step 2) Analyze the impact of the risk occurring

Once the risk is identified ,Each risk should be classified on the basis of following two parameters,

- The **probability** of occurrence
- The **impact** on the project
- Prepare the risk table

Risk Projection & Building a Risk Table

Risk projection, also called ***risk estimation***, attempts to rate each risk in two ways

- the likelihood or probability that the risk is real and
- the consequences of the problems associated with the risk, should it occur.

The project planner, along with other managers and technical staff, performs four risk projection activities:

1. **establish a scale that reflects the perceived likelihood of a risk,**

2.**define the consequences of the risk,**

3.**estimate the impact of the risk on the project and the product, and**.

4.Note the overall accuracy of the risk projection so that there will be no misunderstandings

Building a Risk Table

Risk exposure is a measure of possible future loss (or losses) which may result from an activity or occurrence.

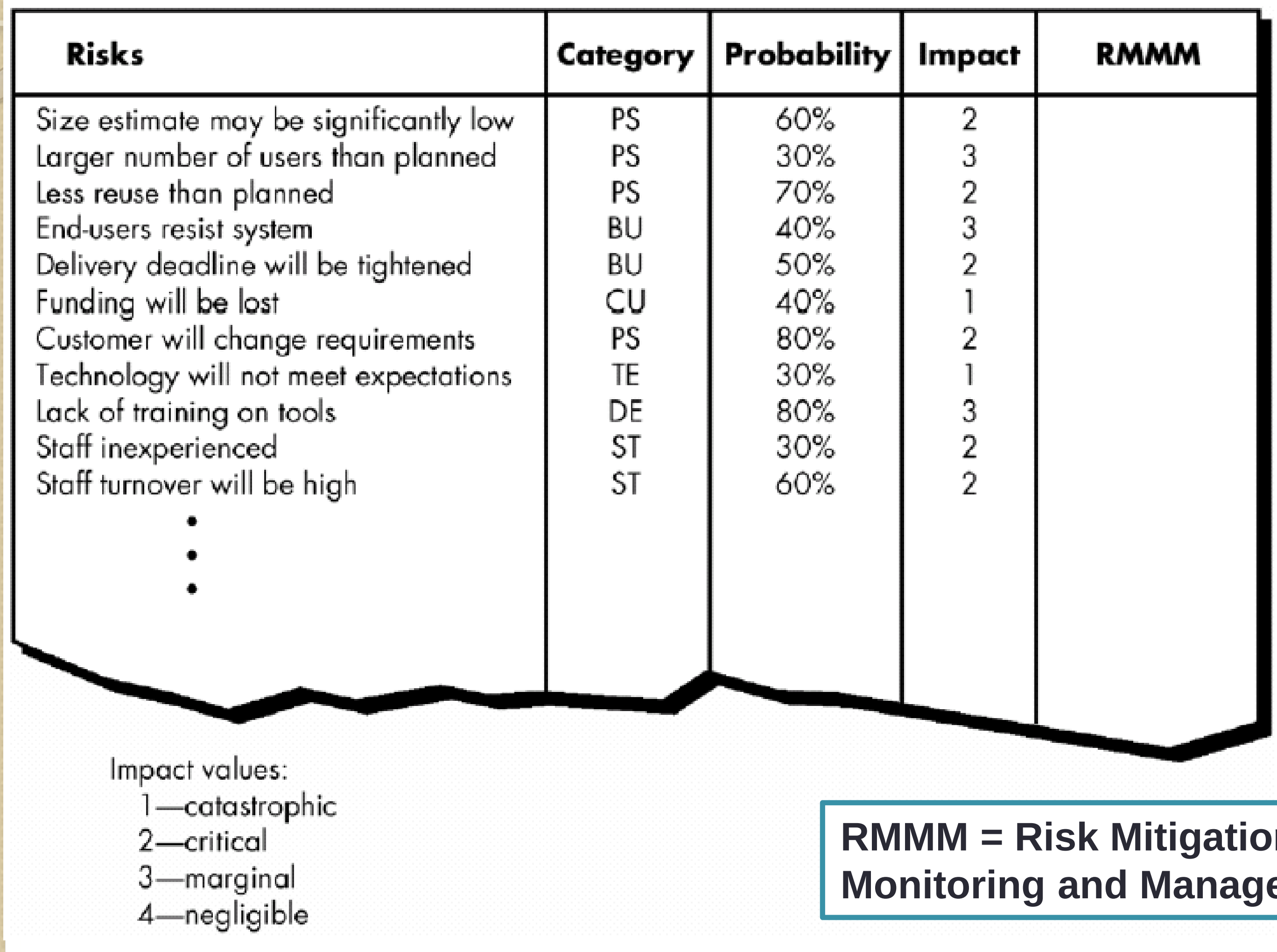
Risk	Probability	Impact	Exposure	RMMM
				Risk Mitigation Monitoring & Management

Text description of the risk

Probability of occurrence

Impact if occurs (Negligible=1…Catastrophic=4)

Building Risk Table – table



Risks	Category	Probability	Impact	RMMM
Size estimate may be significantly low	PS	60%	2	
Larger number of users than planned	PS	30%	3	
Less reuse than planned	PS	70%	2	
End-users resist system	BU	40%	3	
Delivery deadline will be tightened	BU	50%	2	
Funding will be lost	CU	40%	1	
Customer will change requirements	PS	80%	2	
Technology will not meet expectations	TE	30%	1	
Lack of training on tools	DE	80%	3	
Staff inexperienced	ST	30%	2	
Staff turnover will be high	ST	60%	2	
•				
•				
•				

Impact values:
1—catastrophic
2—critical
3—marginal
4—negligible

RMMM = Risk Mitigation, Monitoring and Management Plan

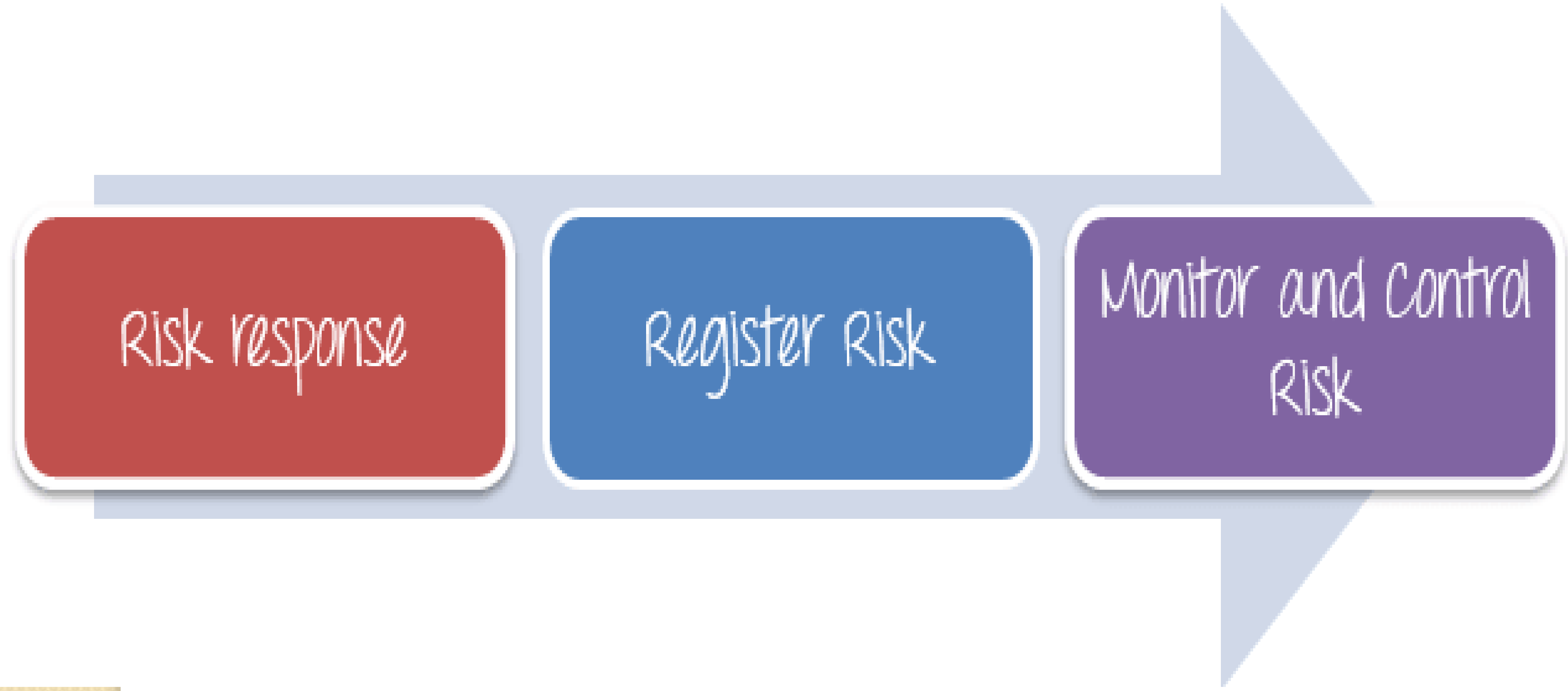
Building the Risk Table

- Estimate the probability of occurrence
- Estimate the impact on the project on a scale of 1 to 4, where
  - 4 = low impact on project success
  - 1 = catastrophic impact on project success
- Determine the exposure:
  - Risk Exposure = Probability x Impact

Risk Exposure Example

- **Risk identification.** Only 70 percent of the software components scheduled for reuse will, in fact, be integrated into the application. The remaining functionality will have to be custom developed.
- **Risk probability.** 80% (likely).
- **Risk impact.** 60 reusable software components were planned. If only 70 percent can be used, 18 components would have to be developed from scratch (in addition to other custom software that has been scheduled for development).
- Since the average component is 100 LOC and local data indicate that the software engineering cost for each LOC is $14.00, the overall cost (impact) to develop the components would be 18 x 100 x 14 = $25,200.
- **Risk exposure.** RE = 0.80 x 25,200 ~ $20,200.

Step 3) Take COUNTER MEASURES to mitigate the risk(RMMM PLAN)



Risk response

Register Risk

Monitor and Control Risk

Risk Mitigation, Monitoring, and Management (RMMM)

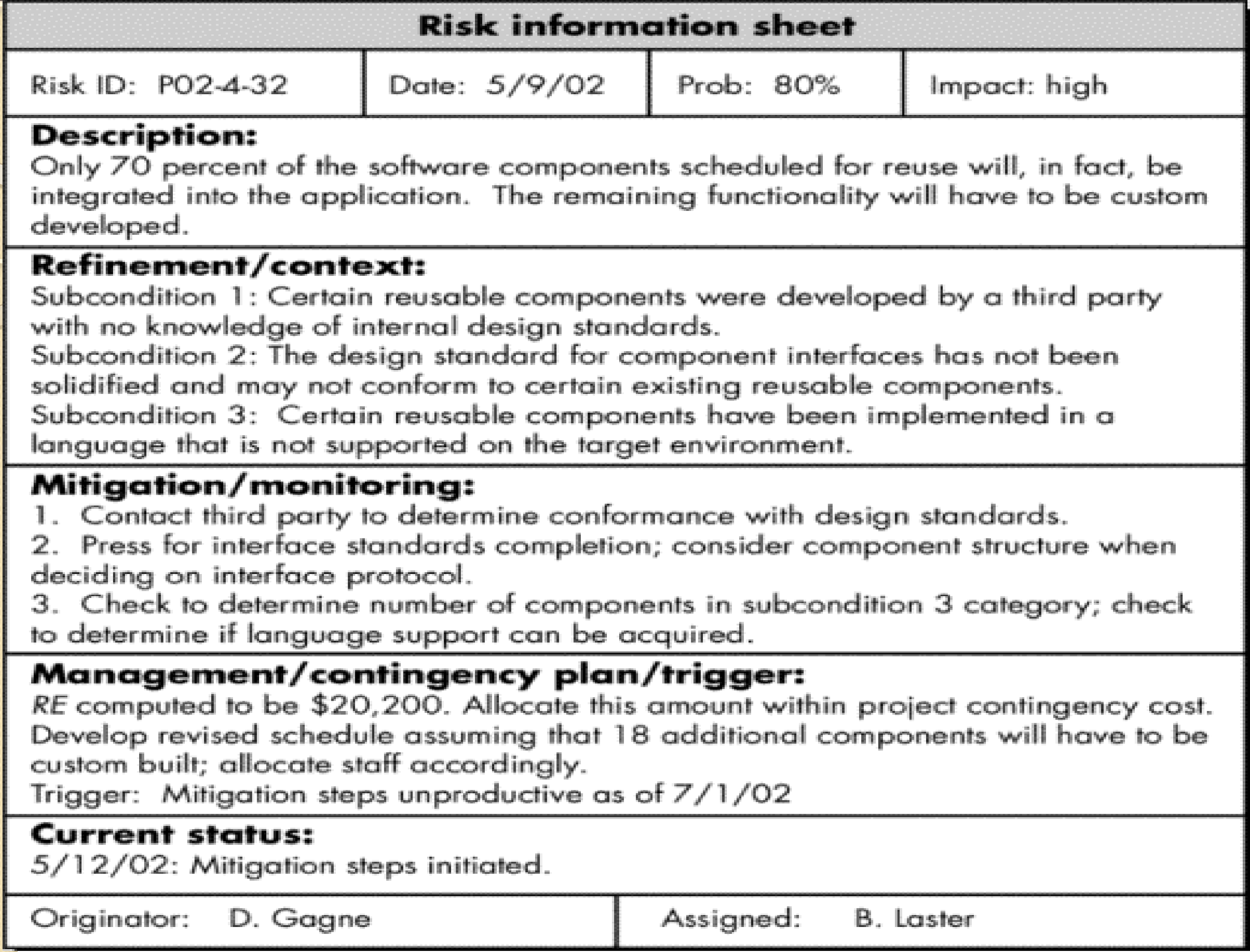
mitigation—how can we avoid the risk?It is a problem avoidance activity

monitoring—what factors can we track that will enable us to determine if the risk is becoming more or less likely?

It is a project tracking activity with 3 primary objectives

- To assess whether predicted risks do,in fact,occur
- To ensure that aversion steps defined for the risk are being properly applied; and
- To collect information that can be used for future risk analysis properly

management—what contingency plans do we have if the risk becomes a reality?



Risk information sheet

Risk ID: P02-4-32	Date: 5/9/02	Prob: 80%	Impact: high

Description:
Only 70 percent of the software components scheduled for reuse will, in fact, be integrated into the application. The remaining functionality will have to be custom developed.

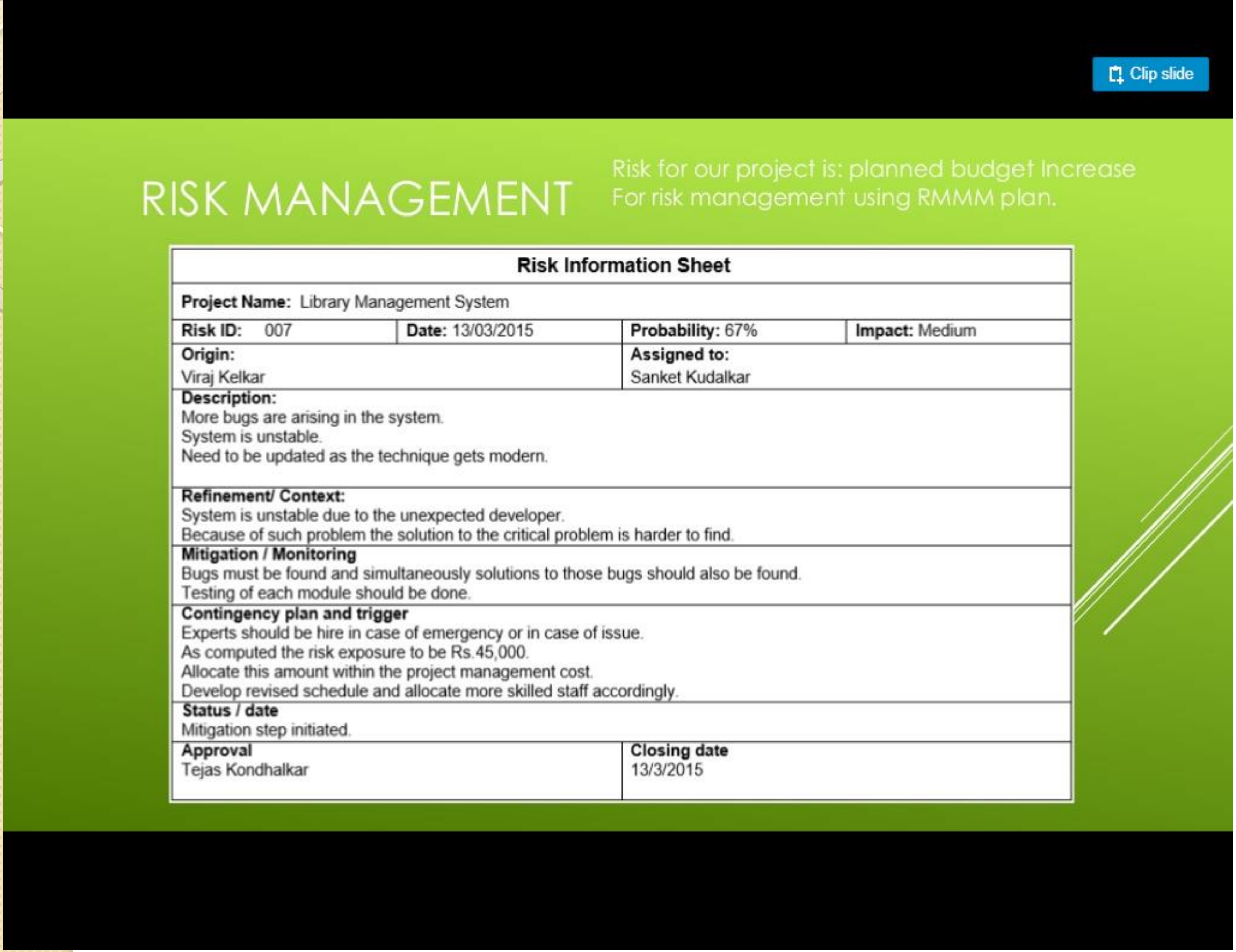
Refinement/context:
Subcondition 1: Certain reusable components were developed by a third party with no knowledge of internal design standards.
Subcondition 2: The design standard for component interfaces has not been solidified and may not conform to certain existing reusable components.
Subcondition 3: Certain reusable components have been implemented in a language that is not supported on the target environment.

Mitigation/monitoring:
1. Contact third party to determine conformance with design standards.
2. Press for interface standards completion; consider component structure when deciding on interface protocol.
3. Check to determine number of components in subcondition 3 category; check to determine if language support can be acquired.

Management/contingency plan/trigger:
RE computed to be $20,200. Allocate this amount within project contingency cost. Develop revised schedule assuming that 18 additional components will have to be custom built; allocate staff accordingly.
Trigger: Mitigation steps unproductive as of 7/1/02

Current status:
5/12/02: Mitigation steps initiated.

Originator: D. Gagne	Assigned: B. Laster



RISK MANAGEMENT

Risk for our project is: planned budget Increase
For risk management using RMMM plan.

Risk Information Sheet

Project Name: Library Management System			
Risk ID: 007	**Date:** 13/03/2015	**Probability:** 67%	**Impact:** Medium
Origin: Viraj Kelkar		**Assigned to:** Sanket Kudalkar	
Description: More bugs are arising in the system. System is unstable. Need to be updated as the technique gets modern.			
Refinement/ Context: System is unstable due to the unexpected developer. Because of such problem the solution to the critical problem is harder to find.			
Mitigation / Monitoring Bugs must be found and simultaneously solutions to those bugs should also be found. Testing of each module should be done.			
Contingency plan and trigger Experts should be hire in case of emergency or in case of issue. As computed the risk exposure to be Rs.45,000. Allocate this amount within the project management cost. Develop revised schedule and allocate more skilled staff accordingly.			
Status / date Mitigation step initiated.			
Approval Tejas Kondhalkar		**Closing date** 13/3/2015	

Video Link for RMMM

- https://www.youtube.com/watch?v=Azsze5LIrOg

Software Quality

- It is defined as conformance to explicitly stated functional and performance requirements,explicitly documented development standards,and implicit characteristics that are expected of all professionally developed software

Software Quality focuses on

- Software requirements are the foundation from which quality is measured.Lack of conformance to requirements is lack of quality
- Specified standards define a set of development criteria that guide the manner in which software is engineered.If the criteria are not followed,lack of quality will almost surely result
- A set of implicit requirements often goes unmentioned(e.g.,the desire for ease of use and good maintainability).If software conforms to its explicit requirements but fails to meet implicit requirements,software quality is suspect

Software Reviews

- Software Reviews are a “filter” for the software process.
- Reviews are applied at various points during software engineering and serve to uncover errors and defects that can be removed
- Software Reviews “purify” the software engineering activities that we have called analysis,design,and coding

Software Reviews

- Many different types of reviews can be conducted as part of software engineering
- An informal meeting around the coffee machine is a form of review,if technical problems are discussed
- A formal presentation of software design to an audience of customers,management,and technical staff is also a form of review

Formal Technical Reviews

- A *formal technical review* (FTR) is a software quality control activity performed by software engineers (and others).
- A FTR is the most effective filter from a quality assurance standpoint
- The objectives of an FTR are: (1) to uncover er-rors in function, logic, or implementation for any representation of the software;(2)to verify that the software under review meets its requirements; (3) to ensure that the software has been represented according to predeftned standards; (4) to achieve software that is developed in a uniform manner; and (5) to make projects more manageable.
- In addition, the FTR serves as a training ground, enabling ju-nior engineers to observe different approaches to software analysis, design, and implementation.
- The FTR also serves to promote backup and continuity because a number of people become familiar with parts of the software that they may not have otherwise seen.

Formal Technical Reviews

- The FTR is actually a class of reviews that includes *walkthroughs* and *inspections,round-robin reviews,and other small group technical assessments of software*
- *Every Formal Technical Review is performed as a meeting*
- *If FTR is planned,controlled,and attended in correct way then only it becomes successful*

Steps in FTR

- The review meeting
  - Every review meeting should be conducted by considering the following constraints
    - Involvement of people- Between 3 to 5 people should be involve in the review
    - Advance Preparation-Advance preparation should occur but it should be very short.At the most 2 hours of work for each person can be spent in this preparation
    - Short duration-The duration of the review meeting should be less than two hours

Steps in FTR

- Rather than attempting to review an entire design,walkthroughs are conducted for each component or small groups of components
- The focus of the FTR is on work product(a software component to be reviewed)
- The review meeting is attended by the review leader,all reviewers and the producers

Steps in FTR

- The review leader is responsible for evaluating the product for its deadline.The copies of product material are then distributed to reviewers
- The producer organizes “walkthrough” for the product,explaining the material,while the reviewers raise the issues based on their advance preparation

Review Reporting and Record Keeping

- During the FTR, a reviewer (the recorder) actively records all issues that have been raised.
- These are summarized at the end of the review meeting, and a *re- view issues list* is produced.
- In addition, a *formal technical review summary re- port* is completed.

Review Reporting and Record Keeping

- A review summary report answers three questions:
  - What was reviewed?(Which parts of the product reviewed?)
  - Who reviewed it?(Name the people who reviewed)
  - What were the findings and conclusions?(The conclusion of the meeting)

- The review summary report is a single-page form (with possible attachments).
- It becomes part of the project historical record and may be distributed to the project leader and other interested parties

Review Reporting and Record Keeping

- The review issues list serves two purposes: (1) to identify problem areas within the product and (2) to serve as an action item checklist that guides the producer as corrections are made.
- An issues list is normally attached to the summary report. You should establish a follow-up procedure to ensure that items on the issues list have been properly corrected.

Review Guidelines-The following rep-resents a minimum set of guidelines for formal technical reviews:

- During the review meeting, always review the product and do not review the producer who produced it
- The review leader should ensure healthy and friendly environment and there should not be any misconduct by any attendee of the review meeting
- The review leader should immediately call the meeting cancelled if situation goes out of control

Review Guidelines

- Set an agenda and maintain it
- For any issue raised, there should be limit in the debate
- List the problems and do not try to solve all the problems as it is impossible and people will not be agreed upon
- Always take a note of all the issues discussed
- Limit the number of participants and insist upon advance preparation.-Keep the number of people involved to the necessary minimum. However, all review team members must prepare in advance.

Review Guidelines

- Develop a checklist for each product that is to be reviewed.
- Allocate resources and schedule time for FTRs.
- Conduct meaningful training for all reviewers. To be effective all re- view participants should receive some formal training. The training should stress both process-related issues and the human psychological side of reviews.
- All early reviews must be reviewed

Walkthrough

- Walkthrough is a review meeting but it is different from Inspection as it is not a formal process
- Generally it is started by the author of code
- In walkthrough,document or code is read by author and others can write note on the defects and suggestions about it
- Walkthrough is informal way of testing so there is no need of moderator while performing walkthrough

Walkthrough

- We can call it as Open Ended discussion because preparation before meeting and creation of a list of observation is not necessary
- It is a informal way of review so it does not focus on documentation.Defect tracking is challenging task in walkthroughs

Objectives of Walkthrough

- Understand and learn the development of software product till date
- Detecting defects in the developed software product
- To explain information present in the document
- To verify and discuss about the validity of the proposed system
- Reporting the suggestions given by the others(other Employees)

Comparison between FTR and Walkthrough

Parameter	FTR	Walkthrough
Concept	FTR is a process performed to give assurance of software quality	Walkthrough is a review meeting but it is different from Inspection as it is not a formal process
Performer	This testing activity is done by software engineers and other persons	Generally it is started by the author of the code
Process	In review,a work product is inspected for defects by individuals other than the person who produced it	In a walkthroughs,the producer describes the product and asks for comments from the participants
Work Product	A work product is any significant deliverable developed through the requirements,design,coding,or testing phase of software development	Walkthroughs generally help to examine source code as opposed to design and requirements documents
Way to perform	It is usually performed as a peer review without management involvement	A walkthrough is particularly helpful for higher-level documents such as requirement specifications and architectural

University Questions

- Explain FTR(May-19,5 marks)
- Explain walkthrough(Dec-19,5 marks)
- What is FTR?Explain review guidelines considered during FTR(Dec-19,10 marks)
- Compare FTR and Walkthrough(May 16,5 marks)
- What is FTR in SQA?What are its objectives?Explain the steps in FTR
- Explain the different metrics used for software quality and reliability(Dec-17,10 marks)

- Illustrate change control and version control((May-19,10 marks)
- Illustrate SCM process(Dec-19,10 marks)

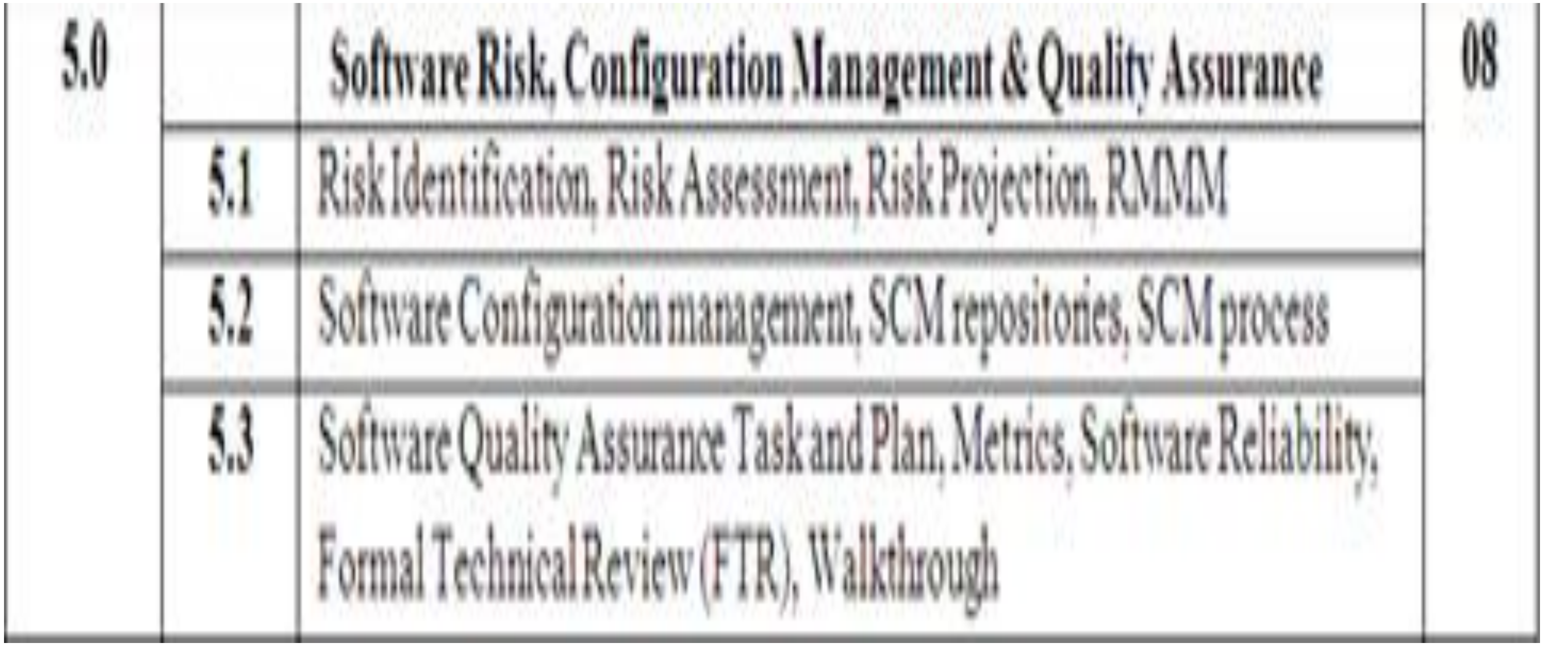
Software Engineering(SE)

CSC 601

Subject Incharge

Varsha Nagpurkar
Assistant Professor
Room No. 407
email: varshanagpurkar@sfit.ac.in

Module-5 Syllabus



5.0		**Software Risk, Configuration Management & Quality Assurance**	08
	5.1	Risk Identification, Risk Assessment, Risk Projection, RMMM	
	5.2	Software Configuration management, SCM repositories, SCM process	
	5.3	Software Quality Assurance Task and Plan, Metrics, Software Reliability, Formal Technical Review (FTR), Walkthrough	

Software Configuration Management

- The output of the software process is information that may be divided into 3 broad categories
  - Computer programs(both source level and executable forms);
  - Work products that describe the computer programs(targeted at both technical practitioners and users,and
  - Data(contained within the program or external to it.)
- The items that comprise all information produced as part of the software process are called a software configuration

software process

- A **software process** (also knows as **software** methodology) is a set of related activities that leads to the production of the **software**. These activities may involve the **development** of the **software** from the scratch, or, modifying an existing system.

The “First Law” of *System Engineering*

- No matter where you are in the system life cycle, the system will change, and the desire to change it will persist throughout the life cycle.

Fundamental sources of change:

- New business or market conditions dictate changes in product requirements or business rules.
- New customer needs demand modification of data produced by information systems, functionality delivered by products, or services delivered by a computer-based system.
- Reorganization or business growth
- Budgetary or scheduling constraints cause a redefinition of the system or product.

Software Configuration Management

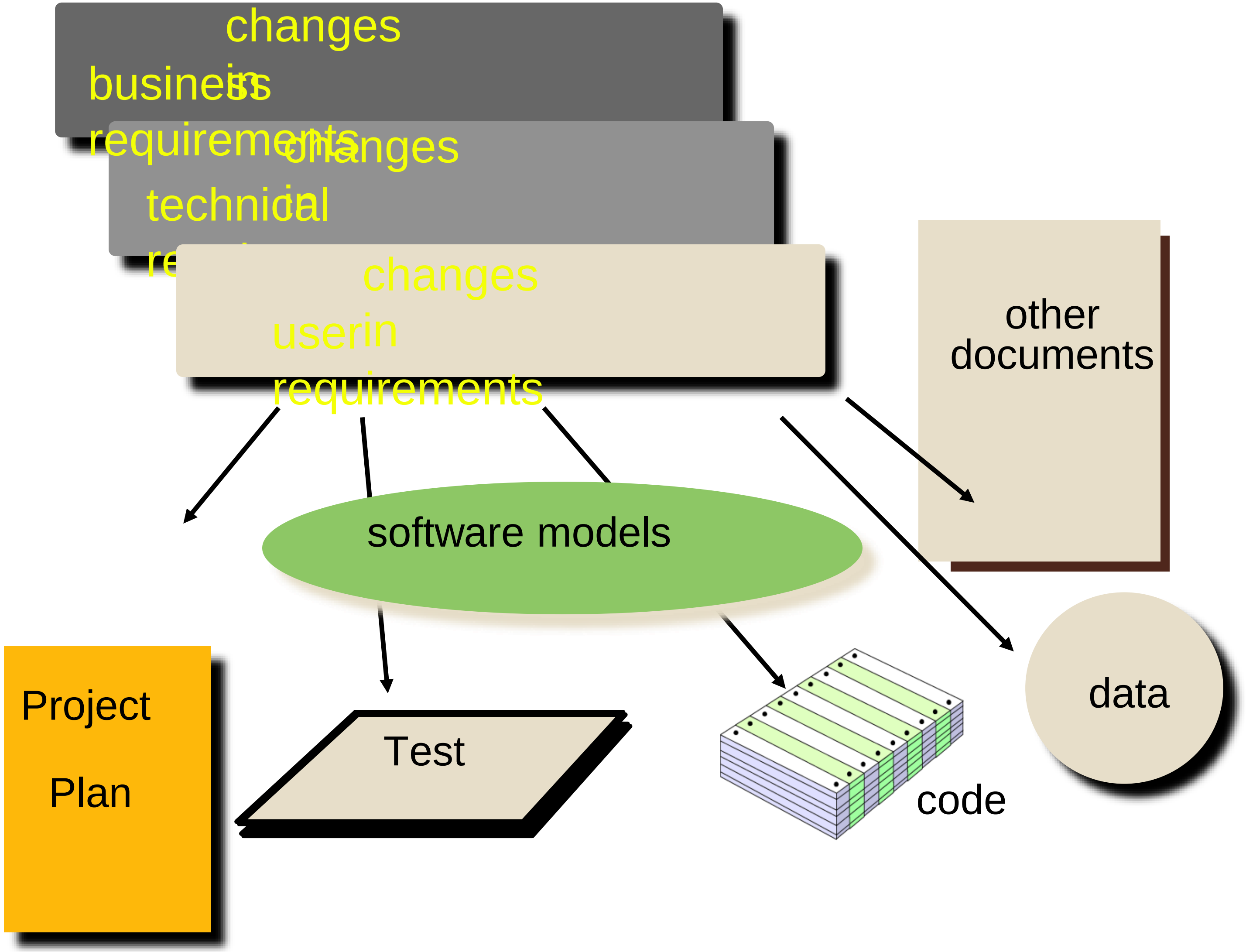
- SCM is a set of activities that have been developed to manage change throughout the life cycle of computer software.
- SCM can be viewed as a software quality assurance activity that is applied throughout the software process

Software Configuration Management (SCM)

The process of *identifying*, *organizing*, and *controlling changes to the software* during development and maintenance.

- SCM is a support activity that makes technical and managerial activities more effective
- SCM operates throughout the SW life-cycle

What Are These Changes?



changes in business requirements

changes in technical re

changes in user requirements

software models

other documents

Project Plan

Test

code

data

Causes of Change

- Evolutionary changes
  - the system evolves as it passes through various stages in the development cycle
- Revolutionary changes
  - such change is caused by the system being unable to satisfy the user's requirements or the customers or producers expectations

Example

- **Microsoft** is one of the most revolutionary companies on the planet. They usually start with an idea someone else has come up with and modify it slightly and make incremental changes over time.

Why Products change ?

- Requirements change during and after development

- Errors are found and need correction

- Variants are needed

Problems of Change

Which component ? Which version ?

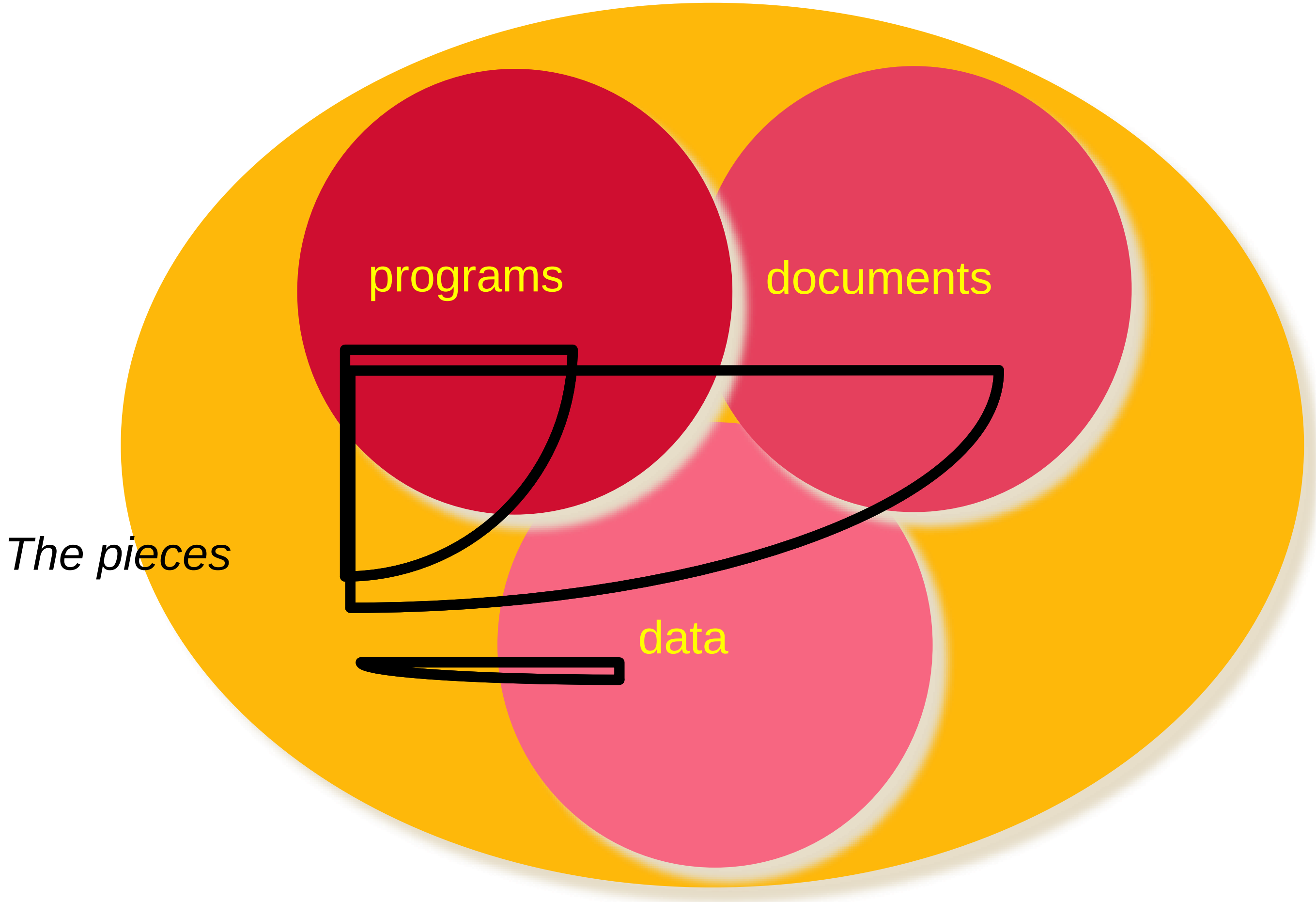
- Double (or multiple) maintenance
- Updates to shared data
- Simultaneous update

Some Definitions ...

Development item item not yet approved, can be informally changed

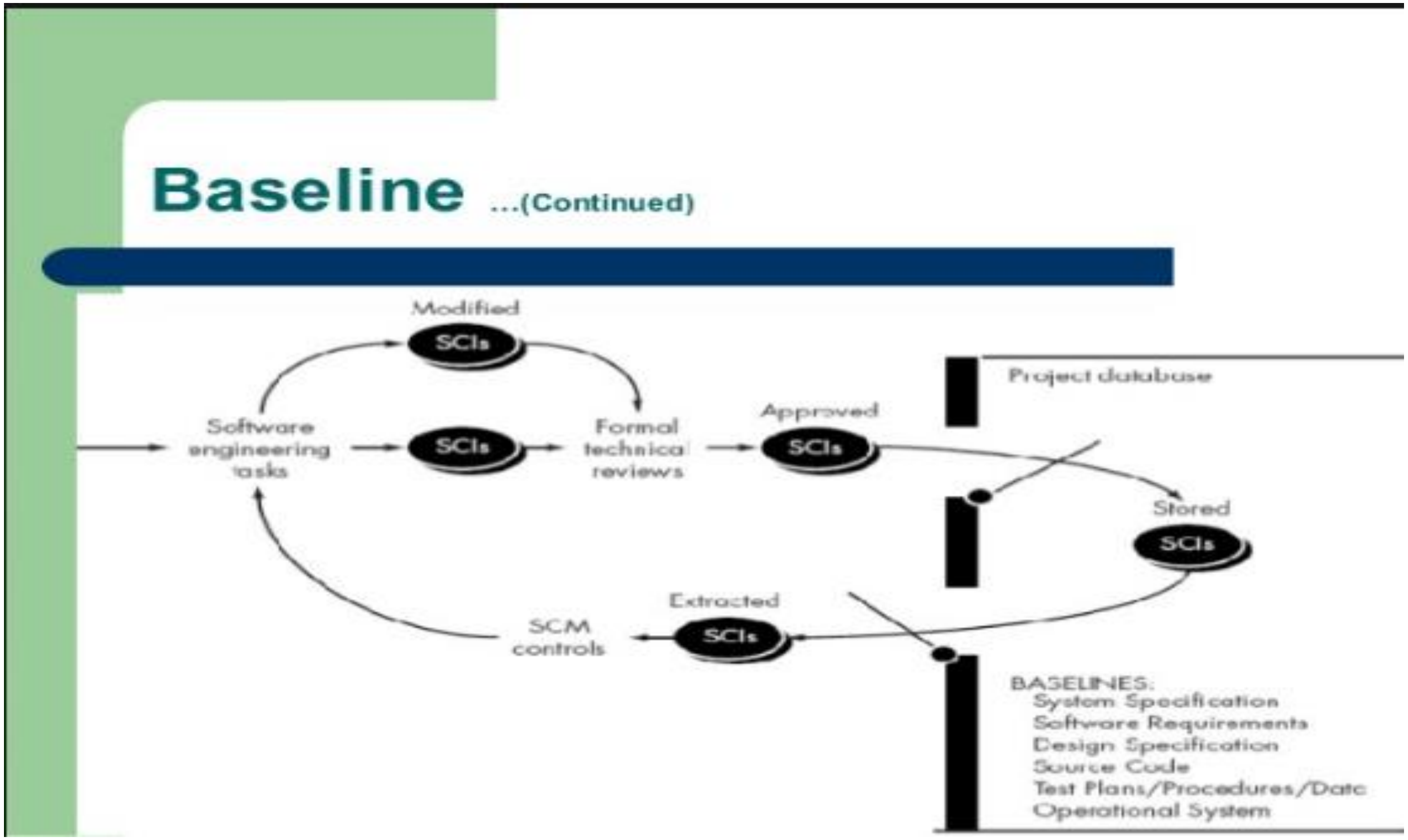
Configuration item (CI) an approved and accepted deliverable, changes done through formal change control procedures

The Software Configuration



Software Configuration Items

- A Software Configuration Item is information that is created as part of the software engineering process



Baseline ...(Continued)

Typical SW Configuration Items (CIs)

- Management plans
- Specifications (requirements, design)
- User documentation
- Test design, case and procedure specifications
- Test data and test generation procedures
- Data dictionaries and databases
- Source code, executable code
- Libraries
- Maintenance documentation
- Support software

Data dictionary

- A data dictionary contains metadata i.e data about the database. The data dictionary is very important as it contains information such as what is in the database, who is allowed to access it, where is the database physically stored etc.
- It contains information about the following −
- Names of all the database tables and their schemas.
- Details about all the tables in the database, such as their owners, their security constraints, when they were created etc.
- Physical information about the tables such as where they are stored and how.
- Table constraints such as primary key attributes, foreign key information etc.
- Information about the database views that are visible.

Milestones and Baselines

Milestone

A milestone is the end of a stage or phase of a project at which one or more deliverables are actually delivered.

Baselines

A baseline is that collection of items which when complete indicates that a milestone in the development process has been reached.

Baselines

- The IEEE (IEEE Std. No. 610.12-1990) defines a baseline as:
  - A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.
- a baseline is a milestone in the development of software that is marked by the delivery of one or more software configuration items and the approval of these SCIs that is obtained through a formal technical review

Baselines

- Baselines serve as the basis for further development
- Baselines can be changed only through formal change control procedures
- Only items that have been approved and obtained through a formal technical review are accepted into the baseline.

SCM Terminology

Version

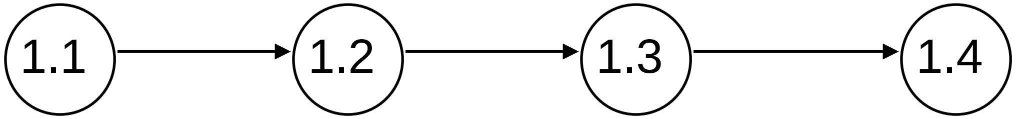
A SW having a defined set of functional capabilities.

Variants

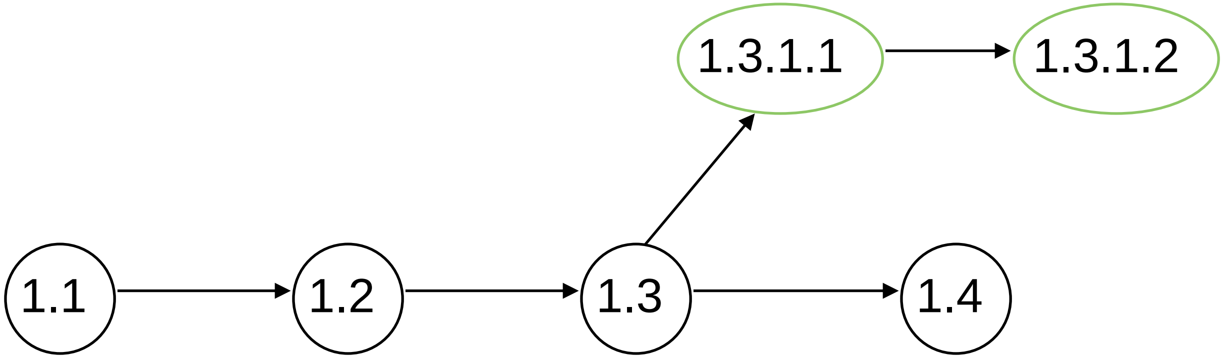
a variation of a version developed to run on different types of HW, or to provide slightly different facilities for different users.

Examples

successive versions



branching versions (variants)



Merging

- Two diverging versions may be merged to create a single new version combining both set of change requests.

- Merge operations are typically done interactively with tool assistance

Configuration Control

- Enforces a rigorous change control mechanism

- Requires formal procedures to
  - request changes
  - carry out impact analysis
  - approve changes
  - carry out changes

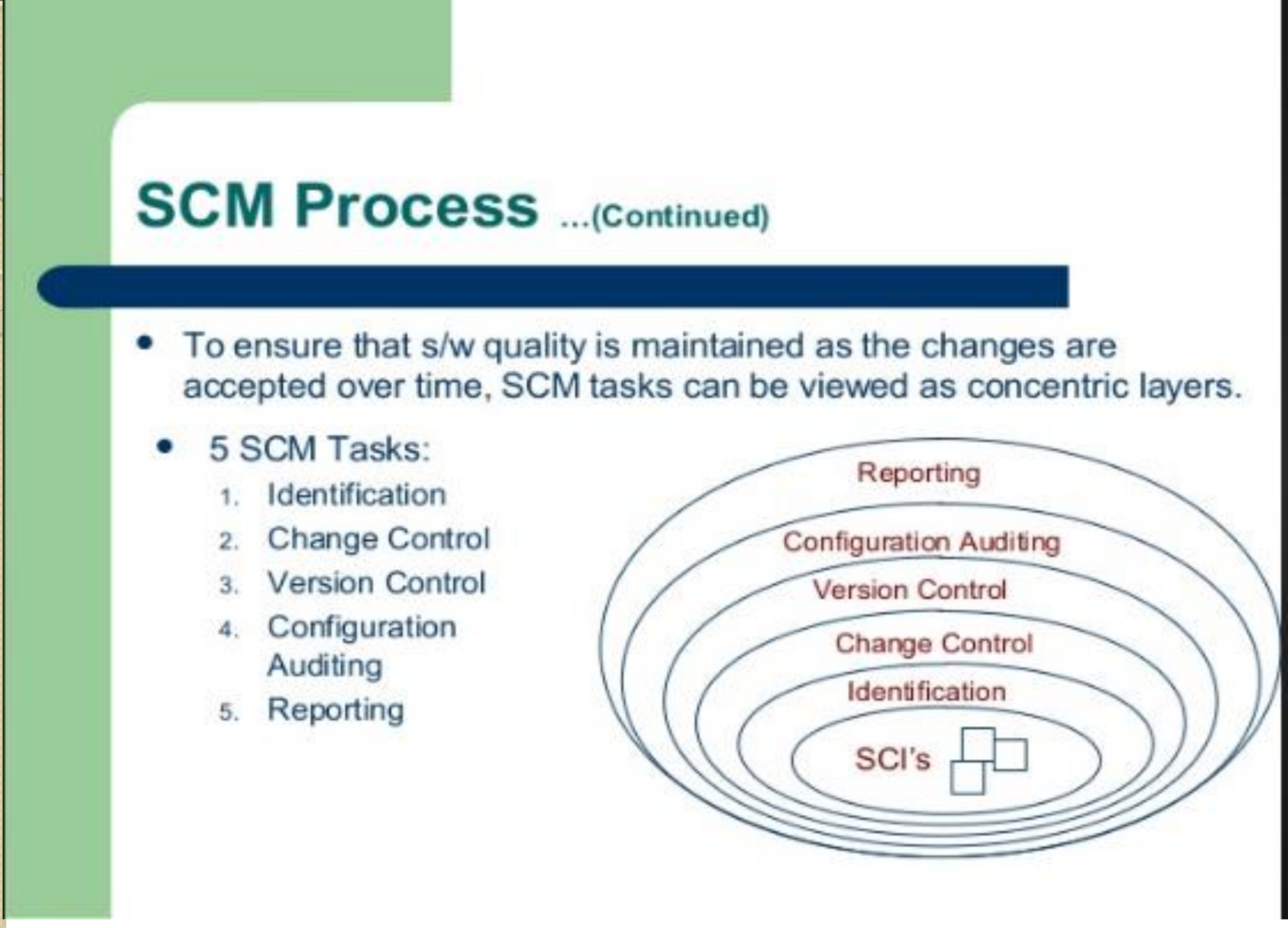
The SCM Process

- The software configuration management process defines a series of tasks that have four primary objectives: (1) to identify all items that collectively define the software configuration, (2) to manage changes to one or more of these items, (3) to facilitate the construction of different versions of an application, and (4) to ensure that software quality is maintained as the configuration evolves over time.

The SCM Process

Addresses the following questions …

- How does a software team identify the discrete elements of a software configuration?
- How does an organization manage the many existing versions of a program (and its documentation) in a manner that will enable change to be accommodated efficiently?
- How does an organization control changes before and after software is released to a customer?
- Who has responsibility for approving and ranking changes?
- How can we ensure that changes have been made properly?
- What mechanism is used to appraise others of changes that are made?



SCM Process ...(Continued)

- To ensure that s/w quality is maintained as the changes are accepted over time, SCM tasks can be viewed as concentric layers.
- 5 SCM Tasks:
  1. Identification
  2. Change Control
  3. Version Control
  4. Configuration Auditing
  5. Reporting

Identification of Objects in the Software Configuration

- To control and manage software configuration items, each should be separately named and then organized using an object-oriented approach.

Change Control Board/Authority(CCB/CCA)

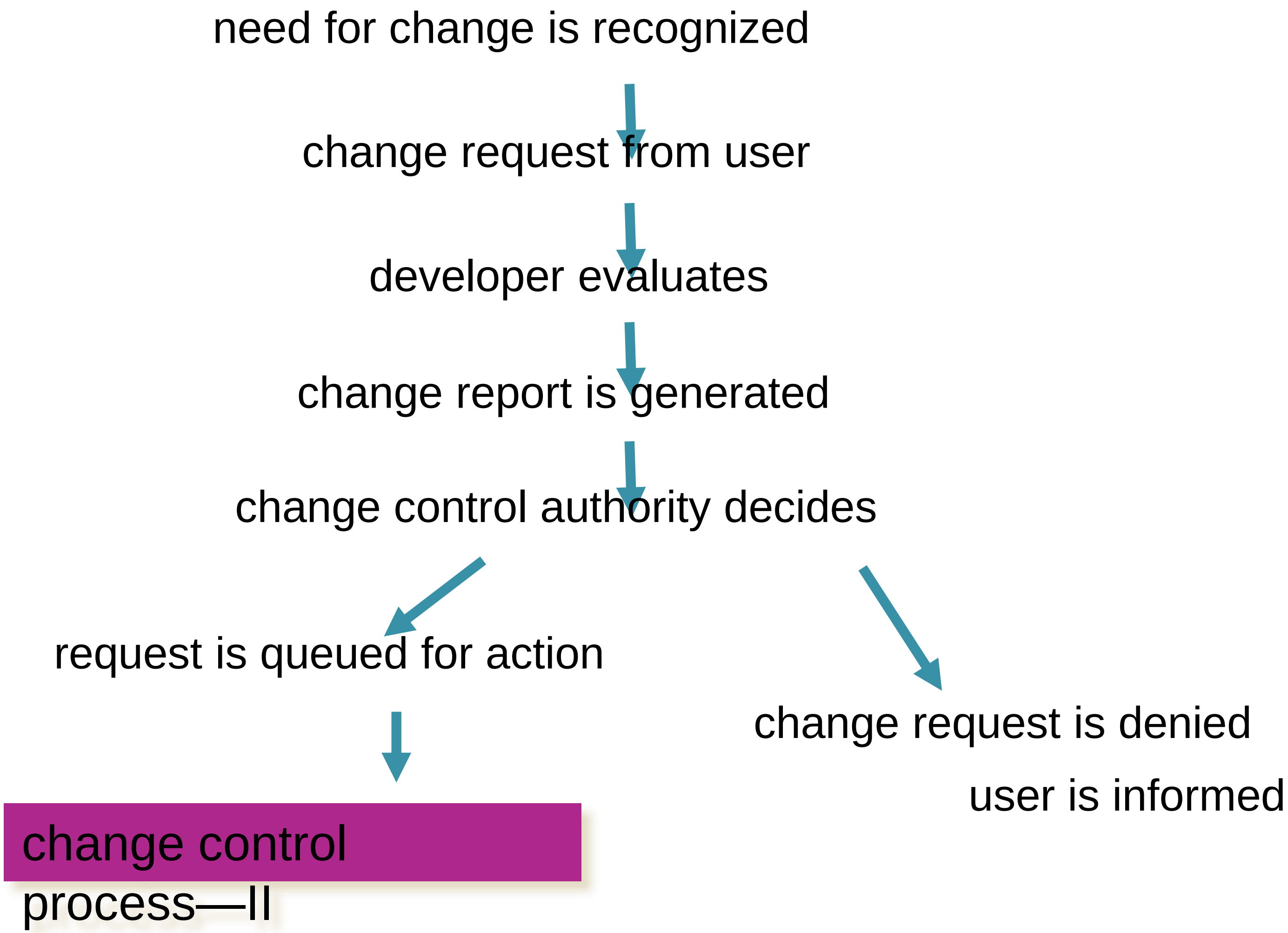
- A group consisting of representatives of user, customer, producer.

- Responsibilities:
  - to approve, monitor and control baselines
  - to approve, monitor, and control changes
  - to authorise changes
- CCB concerns in change approval
  - Is the solution technically ok, cost, schedule, configuration of the whole system, user satisfaction

Change Management Methodology

- Submission of Change Request (CR)
- Technical and business evaluation and impact analysis
- Approval by Change Control Board
- Engineering Change Order (ECO) states
  - changes to be made
  - The constraints that must be respected
  - criteria for review and audit
- CI's checked out
- Changes made and reviewed
- CI's checked in

Change Control Process—I



need for change is recognized

change request from user

developer evaluates

change report is generated

change control authority decides

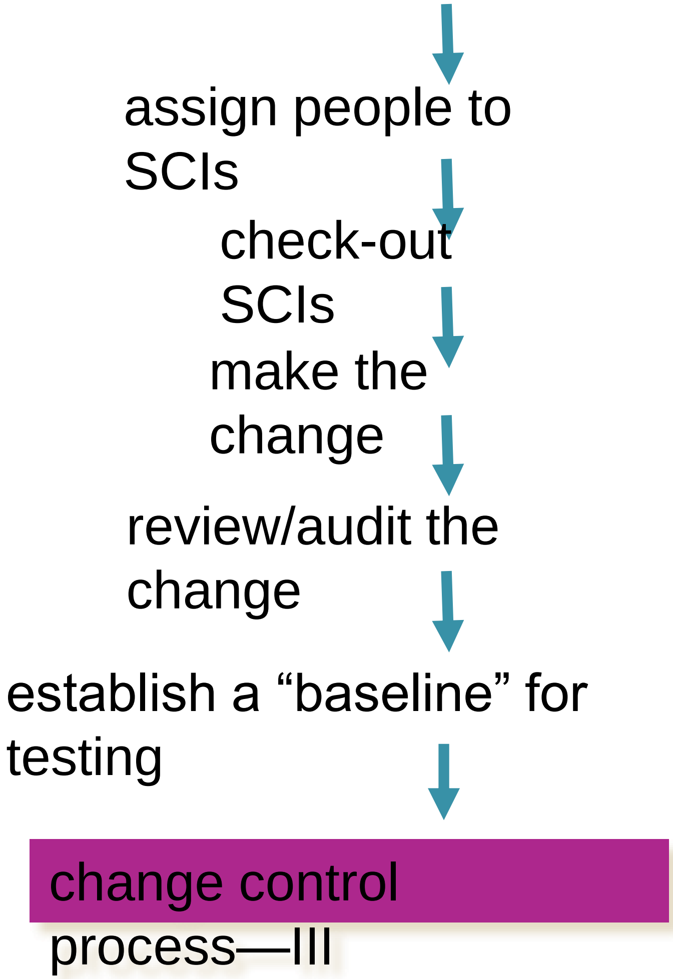
request is queued for action

change request is denied

user is informed

change control process—II

Change Control Process-II



assign people to SCIs

check-out SCIs

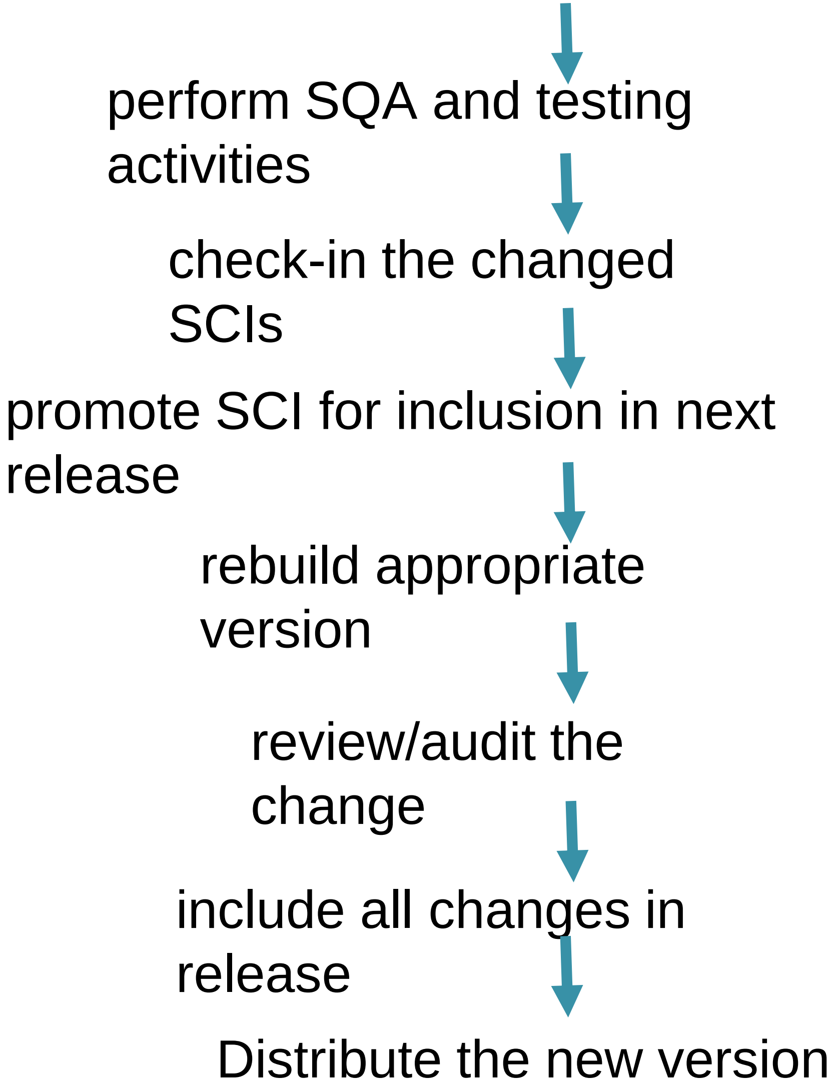
make the change

review/audit the change

establish a “baseline” for testing

change control process—III

Change Control Process-III



perform SQA and testing activities

check-in the changed SCIs

promote SCI for inclusion in next release

rebuild appropriate version

review/audit the change

include all changes in release

Distribute the new version

Version Control

- Version control combines procedures and tools to manage different versions of configuration objects that are created during the software process

- A version control system implements or is directly integrated with four major capabilities:
  - **a *project database (repository)*** that stores all relevant configuration objects
  - **a *version management* capability** that stores all versions of a configuration object

Version Control

- a ***make facility*** that enables the software engineer to collect all relevant configuration objects and construct a specific version of the software.

- an ***issues tracking*** (also called *bug tracking*) capability that enables the team to record and track the status of all outstanding issues associated with each configuration object.

Version Control

- Most of the version control systems create a change set of all the respective changes which are necessary to build a particular version of the software
- It is possible to identify various named change sets for an application or system
- This helps in building a version of the software by mentioning the change sets(by name) which must be applied to the baseline configuration
- A system modeling approach is used to accomplish it

Version Control-System Model contains

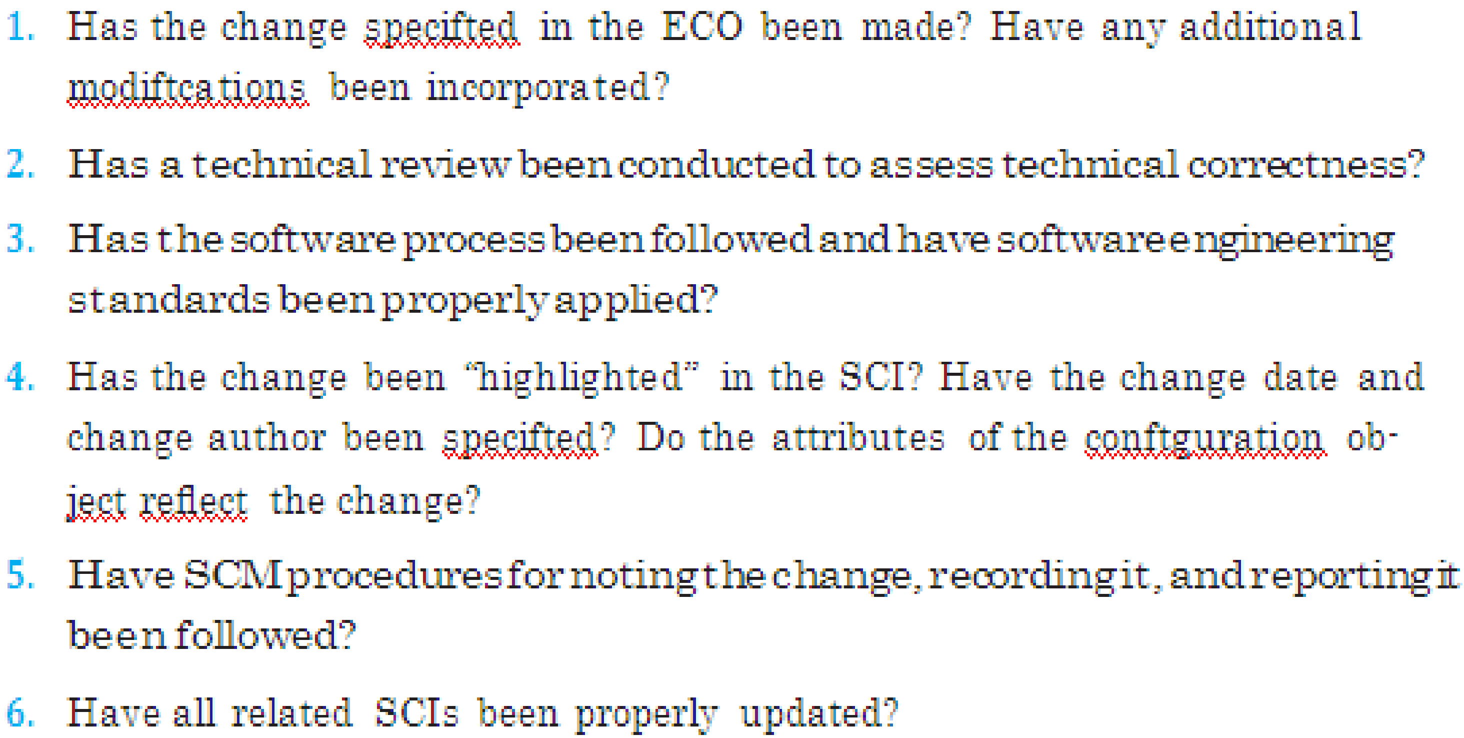
- The system model contains:
  - A template which contains a component hierarchy and a “build order” for the components that describes how the system must be constructed,
  - Contruction rules,and
  - Verification rule
- A number of different automated approaches to version control have been proposed
- Primary difference in approaches is the sophistication of the attributes that are used to construct speciftc versions and variants of a system

Configuration Audit

- To ensure that the change has been properly implemented ,we are conducting FTR and the software configuration audit
- The FTR focuses on the technical correctness of the configuration object that has been modified.
- The reviewers assess the SCI to determine consistency with other SCIs, omissions, or potential side effects.

Configuration Audit

- A *software configuration audit* complements the technical review(FTR) by assessing a configuration object for characteristics that are generally not considered during review. The audit asks and answers the following questions:



1. Has the change specifted in the ECO been made? Have any additional modiftcations been incorporated?
2. Has a technical review been conducted to assess technical correctness?
3. Has the software process been followed and have software engineering standards been properly applied?
4. Has the change been “highlighted” in the SCI? Have the change date and change author been specifted? Do the attributes of the conftguration object reflect the change?
5. Have SCM procedures for noting the change, recording it, and reporting it been followed?
6. Have all related SCIs been properly updated?

Status Reporting

- *Configuration status reporting* (sometimes called *status accounting*) is an SCM task that answers the following questions:
- (1) What happened?
- (2) Who did it?
- (3)When did it happen?
- (4) What else will be affected?

Status Reporting

- The flow of information for configuration status reporting (CSR) is illustrated in change control process.
- Each time an SCI is assigned new or updated identification, a CSR entry is made.
- Each time a change is approved by the CCA (i.e., an ECO is issued), a CSR entry is made. Each time a configuration audit is conducted, the results are reported as part of the CSR task.
- Output from CSR may be placed in an online database or website, so that software developers or support staff can access change information by keyword category.

University Questions

- Illustrate SCM process(Dec-19,10 marks)
- Illustrate change control and version control(May-19,10 marks)

Project Management

- https://www.youtube.com/watch?v=oiVnWX-J5Mo