RISK MANAGEMENT

Risk Analysis and Management

- Risks are potential problems that might affect the successful completion of a software project.
- Risks involve **uncertainty and potential losses.**
- Risk **analysis** and **management** are intended to help a software team understand and manage uncertainty during the development process.
- The important thing is to remember that things can go wrong and to make plans to minimize their impact when they do. The work product is called a **Risk Mitigation, Monitoring, and Management Plan (RMMM).**

Reactive Vs Proactive Risk

- **Reactive: “Indiana Jones School of Risk Management” — project team reacts to risks when they occur**

  1. **Mitigation—plan for additional resources in anticipation of fire fighting**
  2. **Fix on failure —resources are found and applied when the risk strikes**
  3. **Crisis management— if failure does not respond to applied resources then the project is in jeopardy**

Reactive Vs Proactive Risk

Proactive:

- Formal risk analysis is performed.

1. Risk management begins long before technical work starts
2. Potential risks are identified.
3. Their probability and impact are assessed.
4. They are ranked by importance.
5. Project team establishes a plan for managing these risks.

Example

Reactive: A company faces a data breach.

A tech company neglects to regularly update its security systems and does not perform frequent audits on its networks. As a result, hackers exploit a vulnerability in the system, and sensitive customer data is compromised.

- The company only reacts after the data breach occurs.
- They scramble to contain the damage,
- Notify affected users,
- address the breach after it has happened.
- The company incurs reputational damage, legal fees, and customer distrust, all due to not acting sooner to prevent the breach.

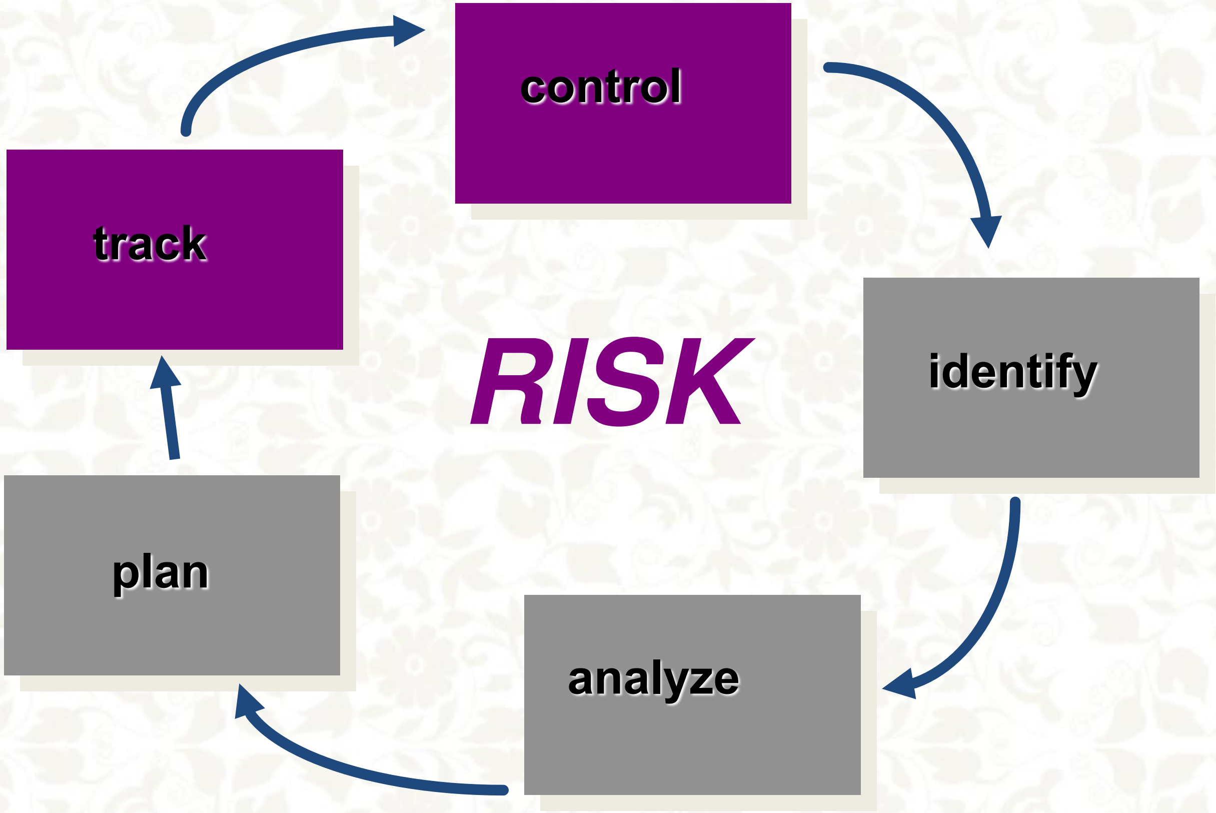
Example

Proactive: **A construction company assessing the risk of delays in a project.**

A construction company plans a large building project. Before beginning, they perform thorough assessments of potential risks, such as supply chain disruptions, bad weather, and labor shortages. They create a risk management plan that includes buffer time, diversified suppliers, and alternative labor resources in case something goes wrong.

- **By addressing potential issues ahead of time,**
- **the company minimizes the likelihood of delays and unexpected costs.**
- **In the event of a disruption, they can respond quickly**
- **have contingency plans in place.**
- **As a result, the project moves forward smoothly and on time, preventing significant losses.**

Risk Management Paradigm



Software Risks

The Risk always involves two characteristics:

- **Uncertainty :** the risk may or may not happen; that is, there are no 100% probable risks.

- **Loss:** if the risk becomes a reality, unwanted consequences or losses will occur.

Category of S/W Risks

- **Project risks -** threaten the project plan
- **Technical risks -** threaten product quality and the timeliness of the schedule .
- **Business risks -** threaten the viability of the software to be built.

1. Building a excellent product or system that no one really wants (Market Risk).
2. Building a product that the sales force doesn't understand how to sell.
3. Losing the support of senior management due to a change in focus or change in people.(Management Risk).
4. Losing budgetary or personnel commitment (Budget risk).

Category of S/W Risks

- **Known risks** - predictable from careful evaluation of current project plan.
- **Predictable risks** – are extrapolated from past project experience.
- **Unpredictable risks** – They can and do occur, but they are extremely difficult to identify in advance.

Risk Identification

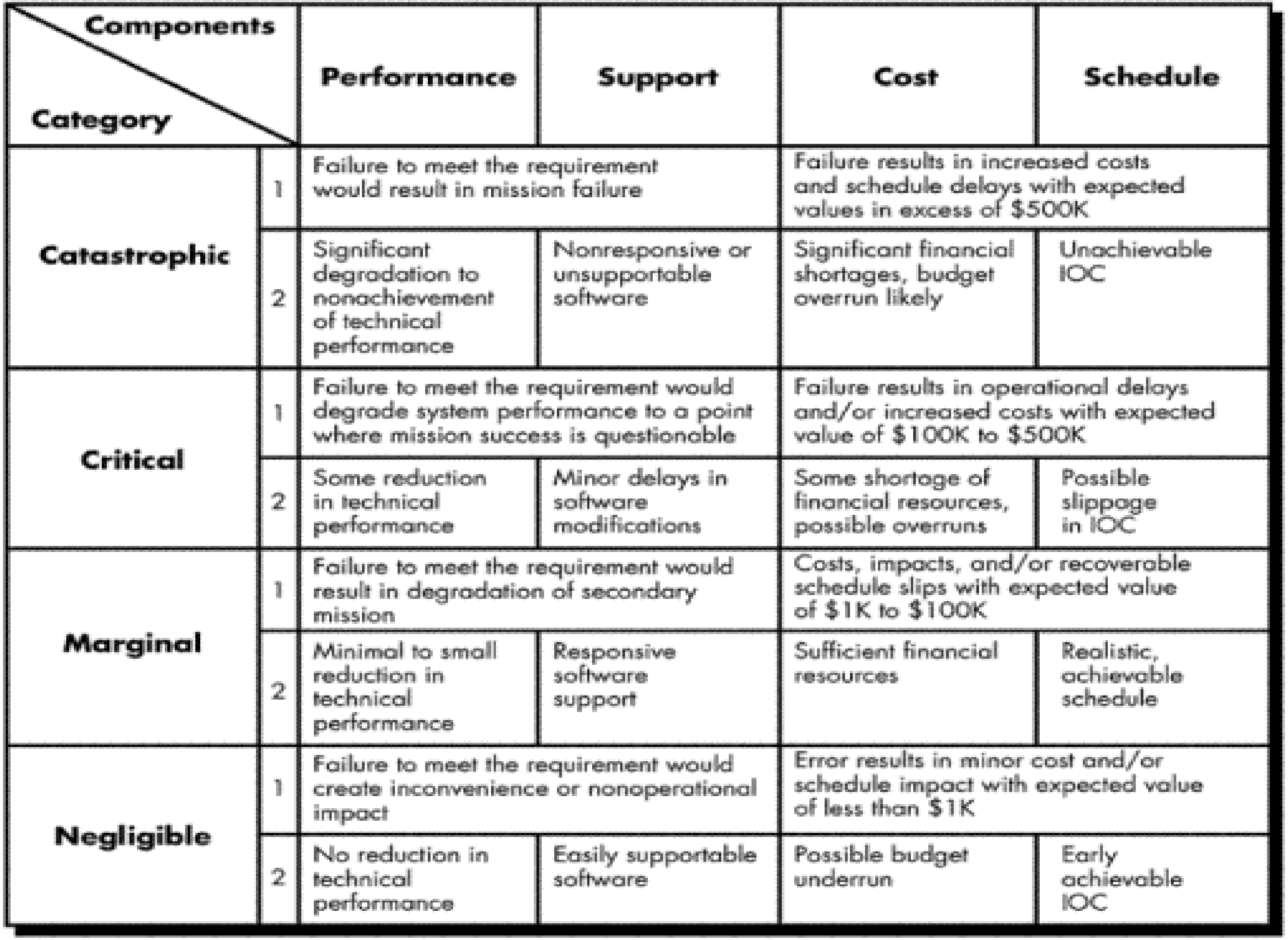
- **Product-specific risks - the project plan and software statement of scope are examined to identify any special characteristics of the product that may threaten the project plan.**
- **Generic risks - are potential threats to every software product.**
  - **product size**
  - **business impact**
  - **customer characteristics**
  - **process definition**
  - **development environment**
  - **technology to be built**
  - **staff size and experience**

Risk Impact Assessment

- Risk components - performance, cost, support, schedule
- Risk impact - negligible, marginal, critical, catastrophic
- The risk drivers affecting each risk component are classified according to their impact category and the potential consequences of each undetected software fault or unachieved project outcome are described

Impact Assessment

Components	Performance	Support	Cost	Schedule
Category				
Catastrophic				
Critical				
Marginal				
Negligible				



Components / Category		Performance	Support	Cost	Schedule
Catastrophic	1	Failure to meet the requirement would result in mission failure		Failure results in increased costs and schedule delays with expected values in excess of $500K	
	2	Significant degradation to nonachievement of technical performance	Nonresponsive or unsupportable software	Significant financial shortages, budget overrun likely	Unachievable IOC
Critical	1	Failure to meet the requirement would degrade system performance to a point where mission success is questionable		Failure results in operational delays and/or increased costs with expected value of $100K to $500K	
	2	Some reduction in technical performance	Minor delays in software modifications	Some shortage of financial resources, possible overruns	Possible slippage in IOC
Marginal	1	Failure to meet the requirement would result in degradation of secondary mission		Costs, impacts, and/or recoverable schedule slips with expected value of $1K to $100K	
	2	Minimal to small reduction in technical performance	Responsive software support	Sufficient financial resources	Realistic, achievable schedule
Negligible	1	Failure to meet the requirement would create inconvenience or nonoperational impact		Error results in minor cost and/or schedule impact with expected value of less than $1K	
	2	No reduction in technical performance	Easily supportable software	Possible budget underrun	Early achievable IOC

Note: (1) The potential consequence of undetected software errors or faults.
(2) The potential consequence if the desired outcome is not achieved.

Risk Projection (Estimation)

- Establish a scale that reflects the perceived likelihood of each risk
- Describe the consequences of the risk
- Estimate the impact of the risk on the project and product
- Note the overall accuracy of the risk projection to avoid misunderstandings

Risk Analysis

- Developing a Risk Table (implemented as a spreadsheet):

1. Identify risks
2. Estimate the probability of occurrence. Each member of the project team assigns a probability.
3. Estimate the impact on the project on a scale of 1 to 5:
4. Sort the table by probability and impact
5. Calculate risk exposure:

$$RE = \text{Probability} \times \text{Impact Cost}$$

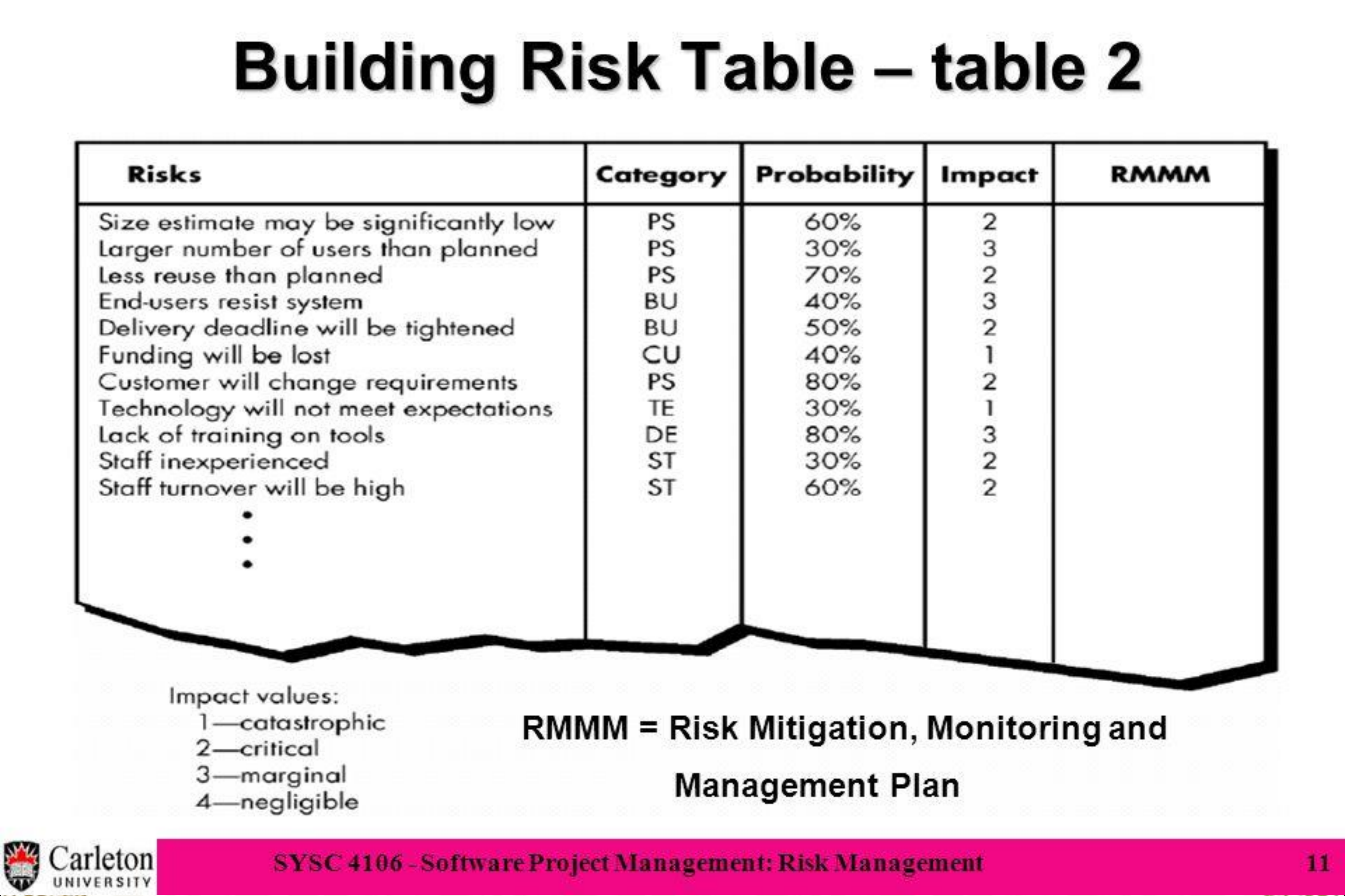
Building a Risk Table

Risk	Category	Probability	Impact	RMMM
				Risk Mitigation Monitoring & Management

Risk Table Construction

List all risks in the first column of the table

- Classify each risk and enter the category label in column two
- Determine a probability for each risk and enter it into column three
- Enter the severity of each risk (negligible, marginal, critical, catastrophic) in column four
- Sort the table by probability and impact value
- Determine the criteria for deciding where the sorted table will be divided into the first priority concerns and the second priority concerns
- First priority concerns must be managed (a fifth column can be added to contain a pointer into the RMMM)



Building Risk Table – table 2

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

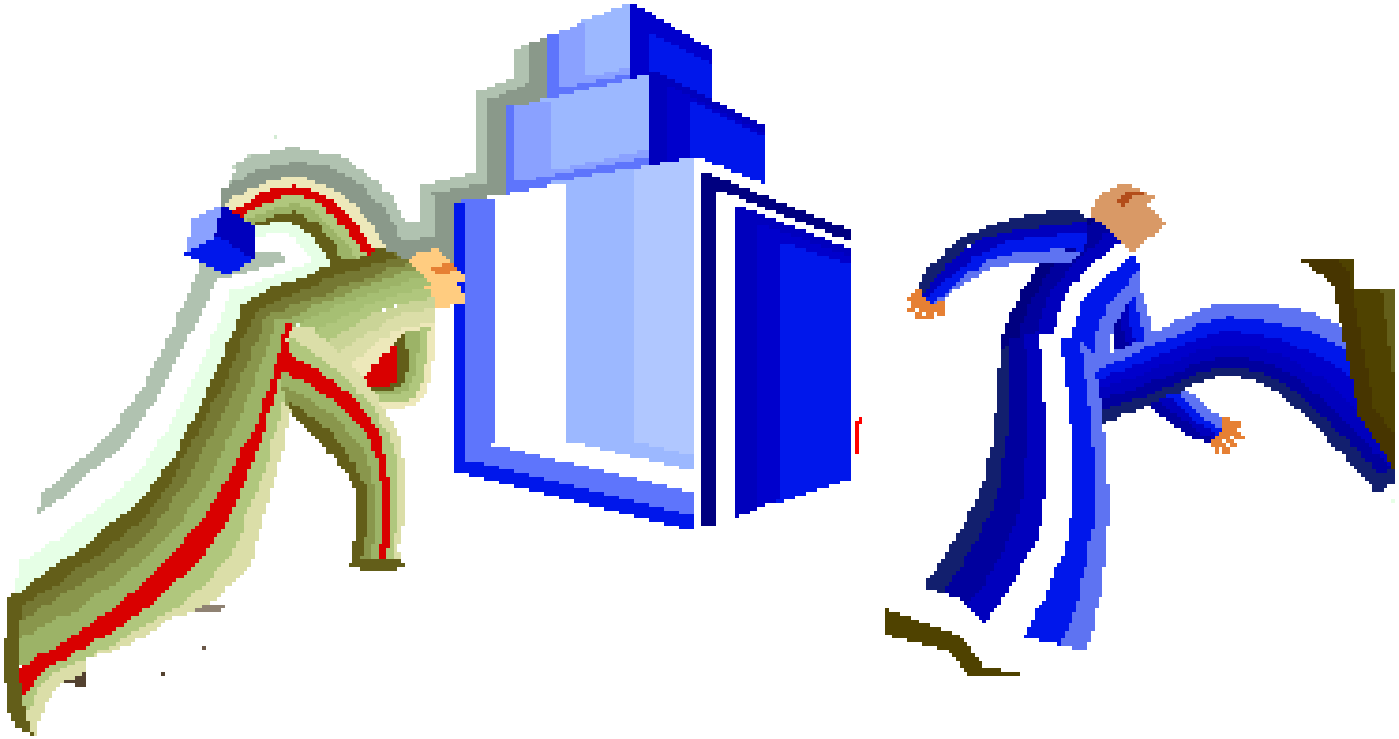
Risk Mitigation, Monitoring, and Management

- mitigation—how can we avoid the risk? (proactive planning for risk avoidance)
- monitoring—what factors can we track that will enable us to determine if the risk is becoming more or less likely? (assessing whether predicted risks occur or not, ensuring risk aversion steps are being properly applied, collect information for future risk analysis, attempt to determine which risks caused which problems)
- management—what contingency plans do we have if the risk becomes a reality? (actions to be taken in the event that mitigation steps have failed and the risk has become a live problem)

RMMM Example

Risk

High staff turnover.



Mitigation plan

- Meet with current staff to determine causes for turnover (e.g. poor working conditions, low pay, competitive job market).
- Once the project commences assume turnover will occur and each develop techniques to ensure continuity when people leave.
- Organize project teams so that information about each development activity is widely dispersed.
- Define documentation standards and establish mechanisms to be sure that documents developed in a timely manner.
- Conduct peer reviews of all work (so that more than one person is “up to date”)
- Assign a backup staff members for every critical technologist.

Monitoring Plan

Following factors should be monitored.

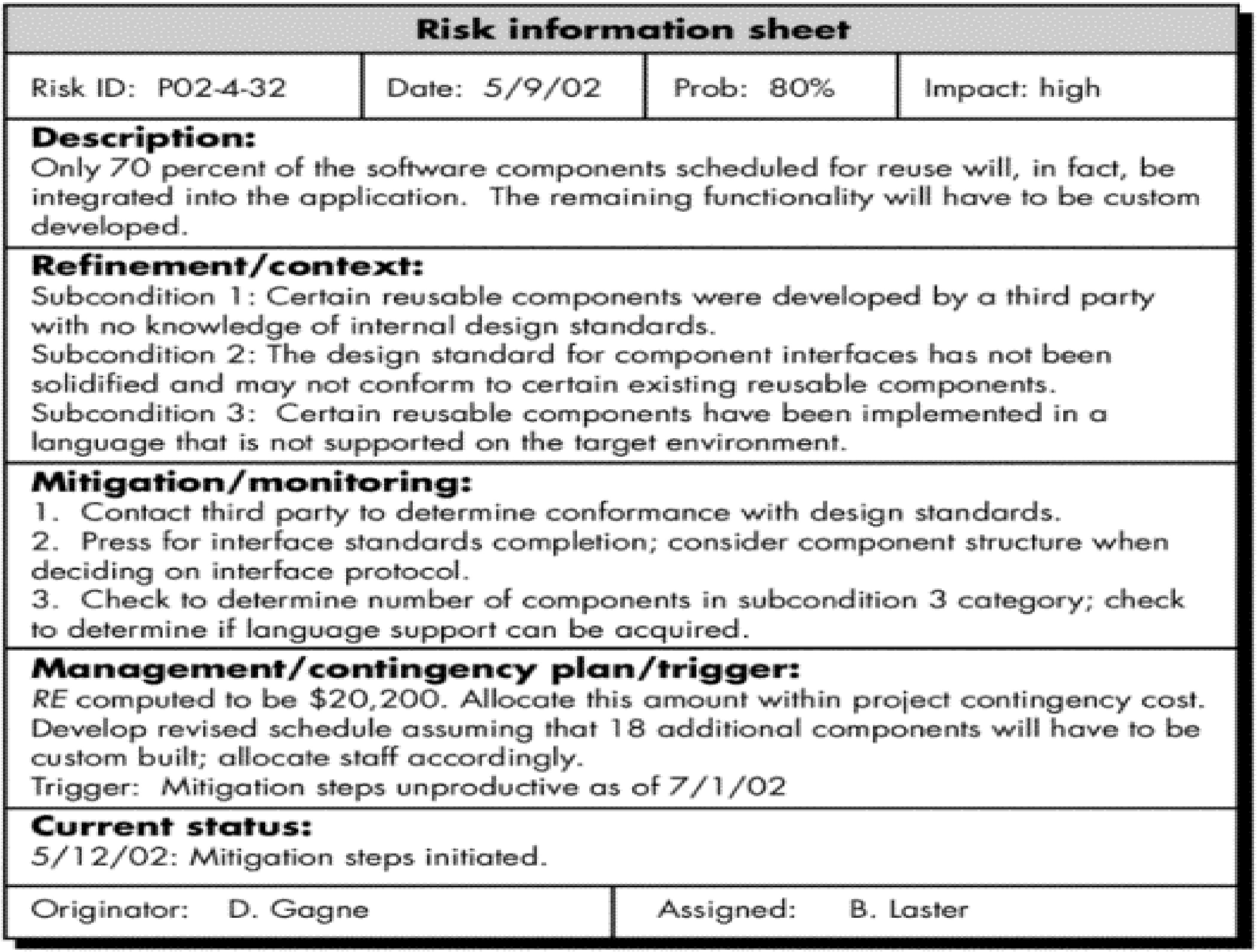
- **General attitude of team members based on project pressures.**
- **The degree to which the team has jelled.**
- **Interpersonal relationship among team members.**
- **Potential problems with compensation and benefits.**
- **The availability of jobs within the company and outside it.**

Risk Management & Contingency planning

- Temporarily refocus resources to those functions that are fully staffed, enabling newcomers who must be added to the team to “get up to speed”.
- Those individuals who are leaving are asked to stop at work and spend their last week in “Knowledge transfer mode”.

Risk Information Sheets

- Alternative to RMMM in which each risk is documented individually.
- Often risk information sheets (RIS) are maintained using a database system.
- RIS components - risk id, date, probability, impact, description, refinement, mitigation/monitoring, management/contingency/trigger, status, originator, assigned staff member.



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

- Explain risk identification, risk projection, RMMM plan in detail.
- Explain risk analysis & management in detail.

Thank You…!

Software Configuration Management

By

Purvi D. Sankhe

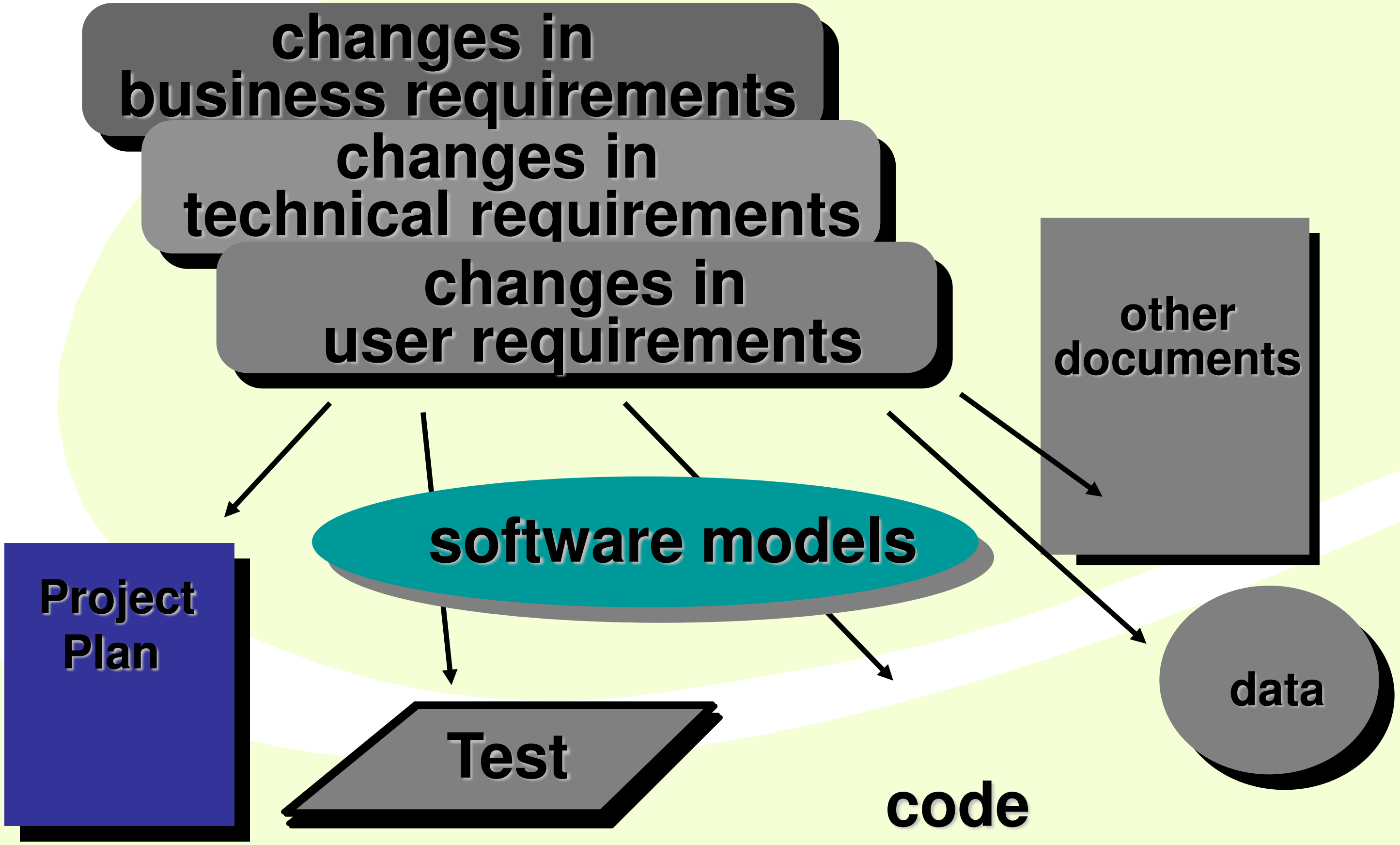
Software Configuration Management

- SCM is an umbrella activity i.e. applied throughout the software process, because change can occur at any time.
- SCM activities are developed to,
  1. Identify change.
  2. Control change
  3. Ensure that change is being properly implemented, and
  4. Report to others who may have an interest.

Fundamental Source of change

- New business or market conditions dictate changes to product requirements or business rules
- New customer needs demand modification of data, functionality, or services
- software engineering team structure
- Budgetary or scheduling constraints cause system to be redefined

What Are These Changes?



changes in business requirements
changes in technical requirements
changes in user requirements
other documents
software models
Project Plan
Test
code
data

Software Configuration Items

- It is information i.e. created as part of software engineering process.
- It may be,
  - Computer programs (both source and executable).
  - Documentation (both technical and user).
  - Data (contained within the program or external to it).

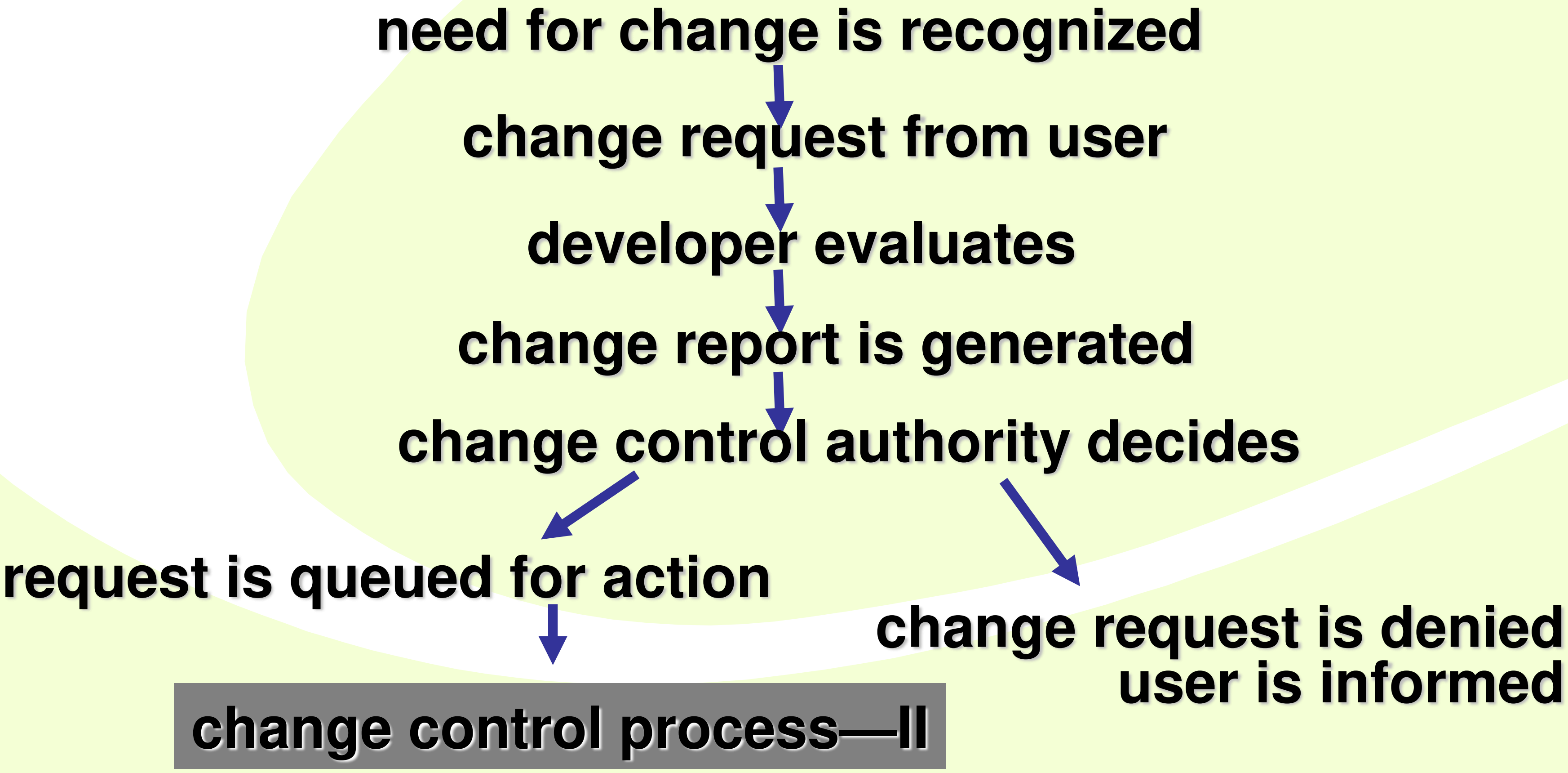
Software Configuration Management

- It is an important element of software quality assurance.
- Its primary responsibility is
  - Identification (tracking multiple versions to enable efficient changes)
  - Version control (control changes before and after release to customer)
  - Change control (authority to approve and prioritize changes)
  - Configuration auditing (ensure changes made properly)
  - Reporting (tell others about changes made)

Version Control

- Combines procedures and tools to manage the different versions of configuration objects created during the software process.
- Configuration management allows a user to specify alternative configuration of the software system through the selection of appropriate versions.
- The evolution graph can be used to describe different versions of a system.
- Each version of the software is a collection of SCIs.

Change Control Process—I



need for change is recognized

change request from user

developer evaluates

change report is generated

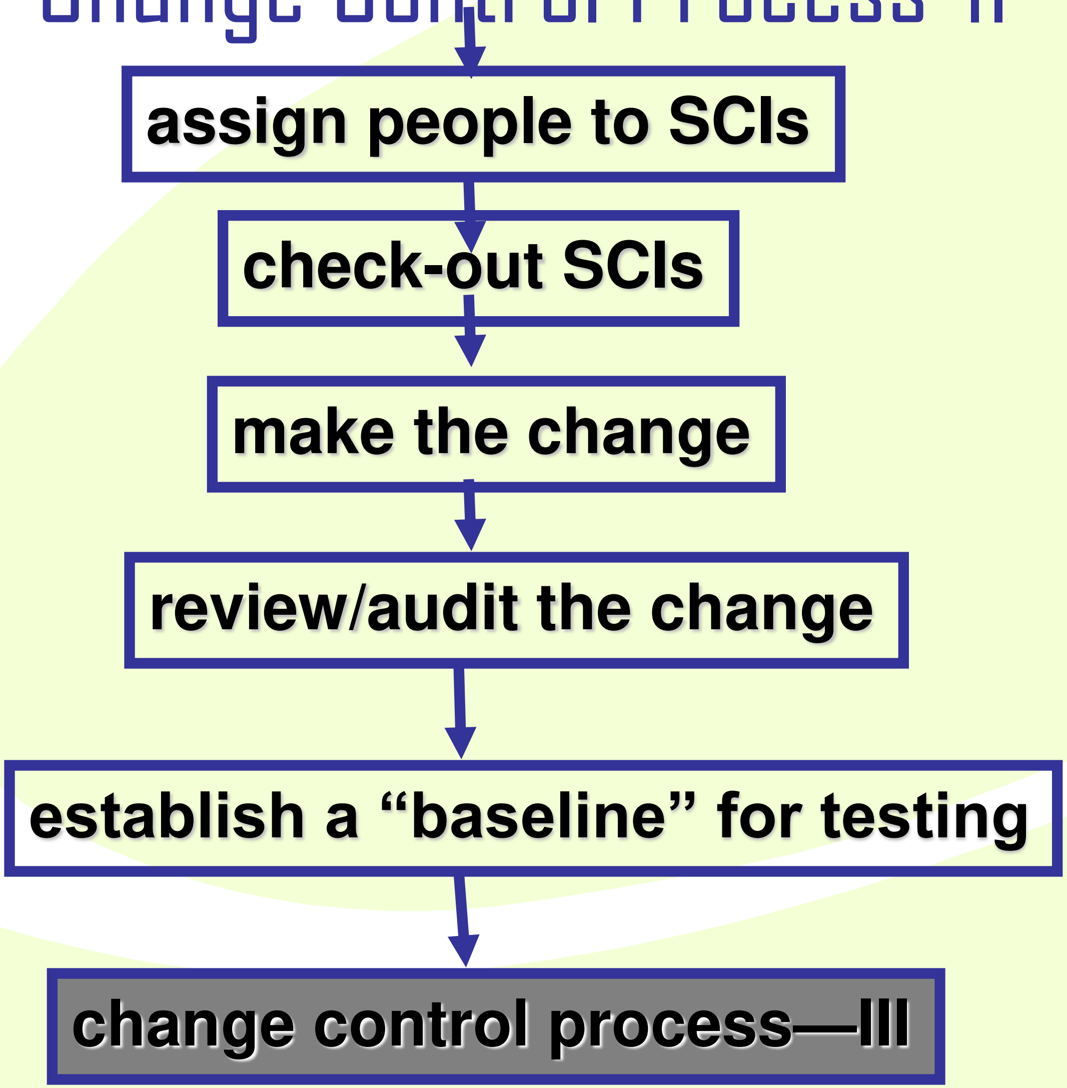
change control authority decides

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

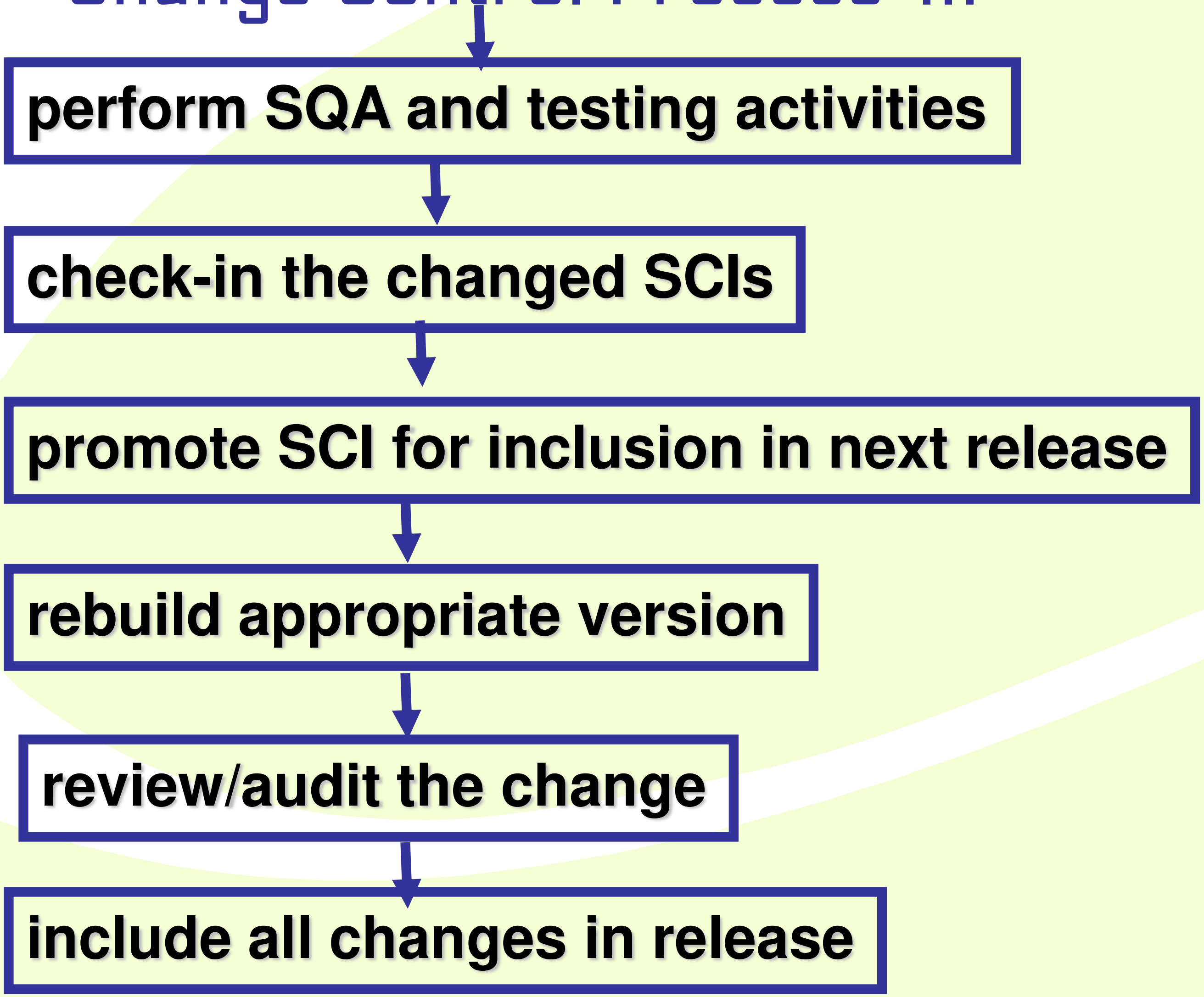
make the change

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

Configuration audit

- How to ensure change has been properly implemented.
  - Formal Technical review
  - Software Configuration audit.
- The FTR focuses on the technical correctness of the configuration object has been modified.
- In FTR, the reviewers assess the SCI to determine consistency with other SCIs, omissions, or potential side effects.

Configuration audit (Continue)

- The audit asks and answers the following questions:
  - Has the change specified by the ECO (Engineering change order) been made without modifications?
  - Has an FTR been conducted to assess technical correctness?
  - Was the software process followed and software engineering standards applied?
  - Have the SCM standards for recording and reporting the change been followed?
  - Were all related SCI's properly updated?

Configuration Status Reporting

Configuration status reporting (or status accounting) is an SCM task that answers following questions:

- What happened?
- Who did it?
- When did it happen?
- What else will be affected by the change?

Configuration Status Reporting contd..

- each time SCI is assigned a new or updated identification , A CSR entry is made.
- After CA. results are reported as a part of the CSR task.