Parul® University

SOFTWARE TESTING & QUALITY ASSURANCE (SUBJECT CODE: 203105395)

Computer Science & Engineering Engineering

Parul® University

TEXT BOOKS

Roger Pressman, Software engineering- A practitioner's Approach, McGraw-Hill International Editions

Reference Books:

- **Yogesh Singh, Software Testing, Cambridge University Press.**
- **William E. Perry, Effective Methods for Software Testing, Second Edition, John Wiley & Sons**

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-6

Risk Management and Change

Software Risks

- **Project risks** : if project risks become real, it is likely that project schedule will slip and that costs will increase. Identify potential budgetary, schedule, personnel (staffing and organization), resource, customer, and requirements problems
- **Technical risks:** threaten the quality and timeliness of the software to be produced. If a technical risk becomes a reality, implementation may become difficult or impossible. Technical risks identify potential design, implementation, interface, verification, and maintenance problems. In addition, specification ambiguity, technical uncertainty, technical obsolescence, and "leading-edge" technology are also risk factors.
- Technical risks occur because the problem is harder to solve than we thought it would be.

Image source : Google

Software Risks

- **Business risks**: threaten the viability of the software to be built.
- Candidates for the top five business risks are

(1) building a excellent product or system that no one really wants (market risk),

(2) building a product that no longer fits into the overall business strategy for the company (strategic risk)

(3) building a product that the sales force doesn't understand how to sell

(4) losing the support of senior management due to a change in focus or a change in people (management risk)

(5) losing budgetary or personnel commitment (budget risks). It is extremely important to note that simple categorization won't always work. Some risks are simply unpredictable in advance.

Image source : Google

Software Risks

- **Predictable risks:** These 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** are the joker in the deck. They can and do occur, but they are extremely difficult to identify in advance

Image source : Google

RISK IDENTIFICATION

- Risk identification 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.
- 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 project at hand
- Generic risks are a potential threat to every software project.
-

Image source : Google

RISK IDENTIFICATION

- **Business impact**—risks associated with constraints imposed by management or the marketplace.
- **Customer characteristics**—risks associated with the sophistication of the customer and the developer's ability to communicate with the customer in a timely manner.
- **Product size**—risks associated with the overall size of the software to be built or modified.
- **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.

Image source : Google

RISK IDENTIFICATION

- **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.
- **Staff size and experience**—risks associated with the overall technical and project experience of the software engineers who will do the work.

Image source : Google

Risk Projection

- Risk projection, also called risk estimation, attempts to rate each risk in two ways—
- the **probability** that the risk is real and the **consequences** of the problems associated with the risk, should it occur.

Risks	Category	Probability	Impact
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

Image source : book, software engineering , roger pressman

Assessing Risk Impact

- Determine the average probability of occurrence value for each risk component.
- determine the impact for each component based on the criteria
- Complete the risk table and analyze the results
- The overall **risk exposure**, RE, is determined using the following relationship

  RE = P x C

  where P is the probability of occurrence for a risk, and C is the cost to the project should the risk occur.

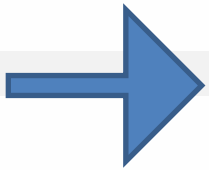
Image source : Google

Parul® University

RISK REFINEMENT & MITIGATION , MONITORING , MANAGEMENT

- As time passes and more is learned about the project and the risk, it may be possible to refine the risk into a set of more detailed risks, each somewhat easier to mitigate, monitor, and manage.
- All of the risk analysis activities presented to this point have a single goal—to assist the project team in developing a strategy for dealing with risk.
- An effective strategy must consider three issues:
- Risk avoidance
- Risk monitoring
- Risk management and contingency planning
- If a software team adopts a proactive approach to risk, avoidance is always the best strategy. This is achieved by developing a plan for risk mitigation

Image source : Google



Parul® University

RISK REFINEMENT & MITIGATION , MONITORING , MANAGEMENT plan

- The RMMM Plan is is a document in which all the risk analysis activities are described.
- each risk is documented individually using a risk information sheet
- Once RMMM has been documented and the project has begun, risk mitigation and monitoring steps commence.
- Risk mitigation is a problem avoidance activity. Risk monitoring is a project tracking activity with three primary objectives:

(1) to assess whether predicted risks occur.
(2) to ensure that risk aversion steps defined for the risk are being properly applied; and
(3) to collect information that can be used for future risk analysis.

Image source : Google

Parul® University

DIGITAL LEARNING CONTENT

SOFTWARE CONFIGURATION MANAGEMENT

- Software configuration management (SCM) is a set of activities designed to control change by identifying the work products that are likely to change, establishing relationships among them, defining mechanisms for managing different versions of these work products, controlling the changes imposed, and auditing and reporting on the changes made
- Origin of these changes

- New business or market conditions
- New customer needs demand modification
- Reorganization or business growth/downsizing
- Budgetary or scheduling constraints

- Software configuration management is a set of activities that have been developed to manage change throughout the life cycle of computer software.

Image source : Google

Baselines

- A baseline is a software configuration management concept that helps us to control change without seriously impeding justifiable change. 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.

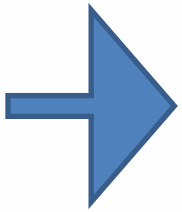


Image source : Google

Software Configuration Items

- software configuration item is information that is created as part of the software engineering process.
- A configuration object has a name, attributes, and is "connected" to other objects by relationships.
- The configuration objects, Design Specification, data model, component N, source code and Test Specification are each defined separately.
- data model and component N are part of the object
- Design Specification. A double-headed straight arrow indicates an interrelationship

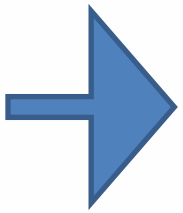


Image source : Google

Software Configuration Items

FIGURE 9.2
Configuration objects

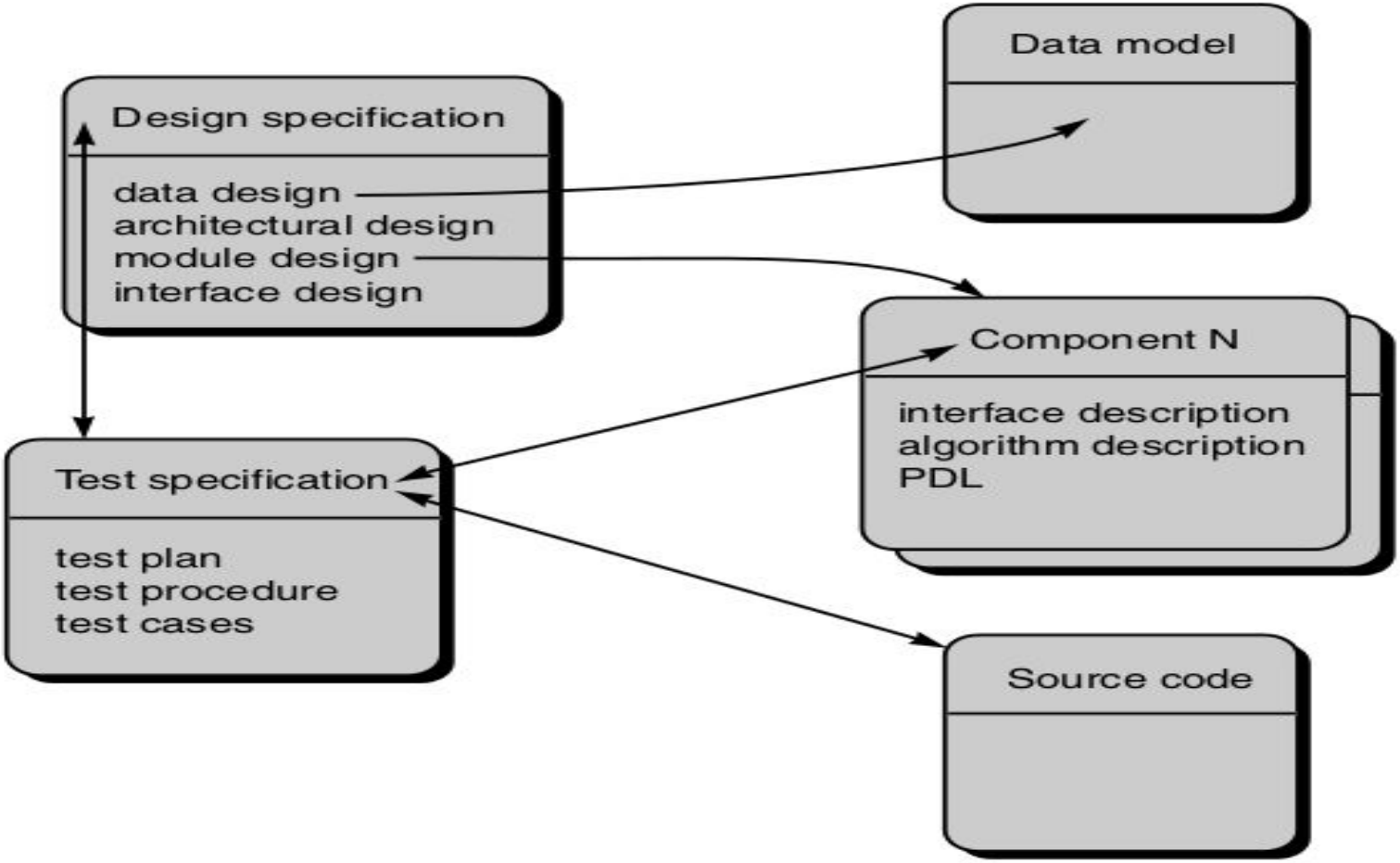


Image source : book, software engineering

CHANGE CONTROL

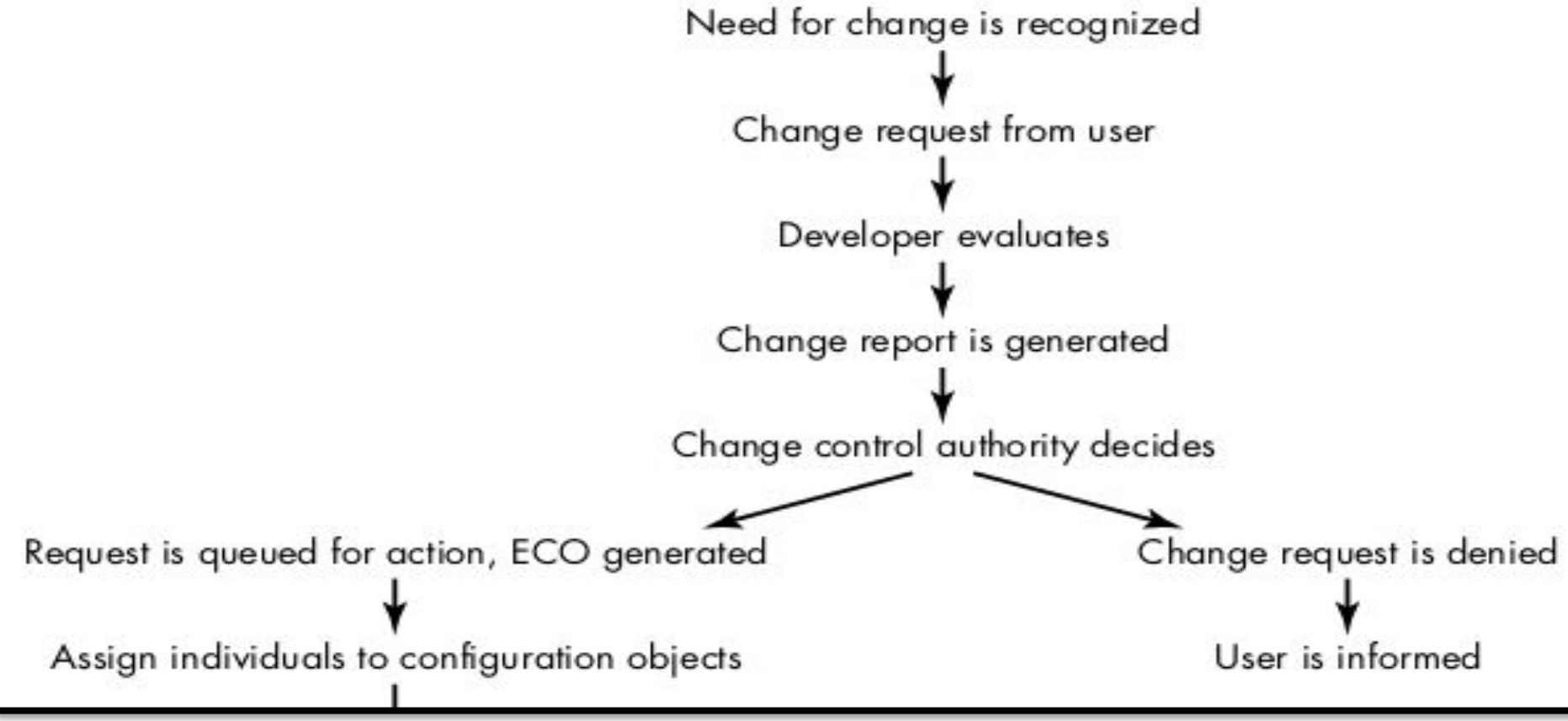


FIG.CHANGE CONTROL PROCESS

Image source : book,

CHANGE CONTROL

Assign individuals to configuration objects

"Check out" configuration objects (items)

Make the change

Review (audit) the change

"Check in" the configuration items that have been changed

Establish a baseline for testing

Perform quality assurance and testing activities

"Promote" changes for inclusion in next release (revision)

Rebuild appropriate version of software

Review (audit) the change to all configuration items

Include changes in new version

Distribute the new version

User is informed

Image source : book, software engineering

Parul® University

DIGITAL LEARNING CONTENT

CONFIGURATION AUDIT

- How can we ensure that the change has been properly implemented? The answer is twofold: (1) formal technical reviews and (2) the software configuration audit.
- A software configuration audit complements the formal technical review by assessing a configuration object for characteristics that are generally not considered during review. The audit asks and answers the following questions:
- Has the change specified in the ECO been made? Have any additional modifications been incorporated?
- Has a formal technical review been conducted to assess technical correctness?
- Has the software process been followed and have software engineering standards been properly applied?
- Has the change been "highlighted" in the SCI? Have the change date and change author been specified? Do the attributes of the configuration object reflect the change?

Image source: Google

Software Configuration Management for the Web

- **Variety of Types:** Web systems are built from a wide diversity of objects including documents in any of markup languages, document templates, style sheets, images, streaming media, animations, applets, and scripts.
- So, an SCM system for Web applications must be compatible with a wide variety of file types, including their templates and their corresponding editing tools.
- in Web applications it is common to intermix different object types in the same file. An example would be an HTML file that contains embedded CSS style commands and JavaScript.

Image source : Google

Software Configuration Management for the Web

- **Developers:** Web systems are developed by diverse teams ranging from graphic artists to specialized software engineers.
- Thus, the SCM system's interface needs to be accessible to less technical users or they will resist its use, leaving important portions of the system outside the control of the SCM tools
- **Rate of Change:** Large Web systems appear to change faster than traditional software systems.
- Some sites, such as news sites, must show thousands of daily content changes. While substantive structural changes visible to end users may be more rare on these sites, they still appear to happen at a faster rate than in traditional software.
- Web-oriented SCM must help developers understand changes at a very fine-grained level.

Image source : Google

Parul® University

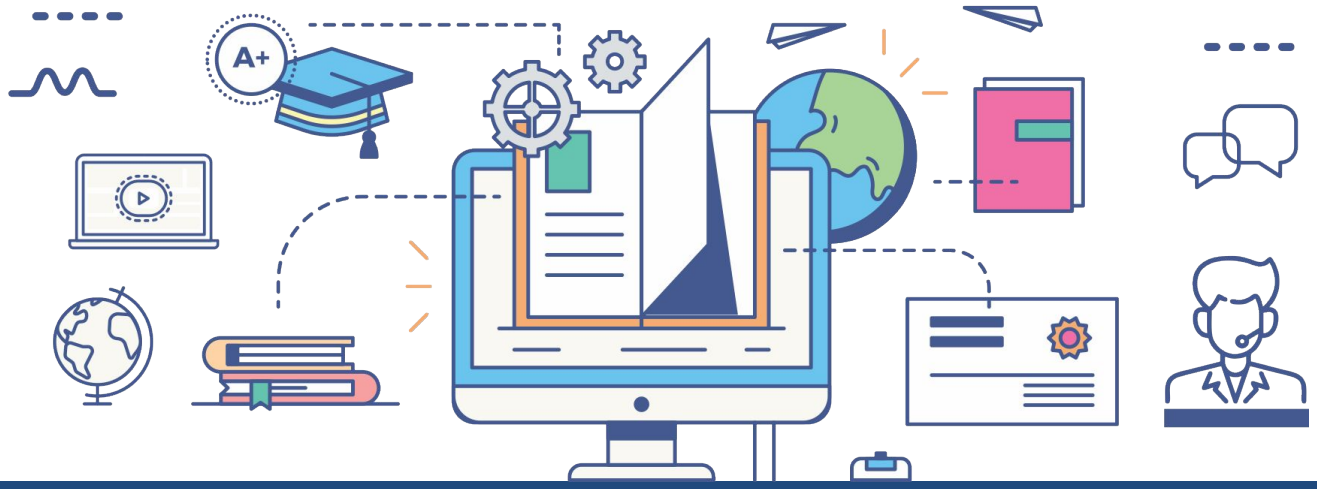
DIGITAL LEARNING CONTENT

Software Configuration Management for the Web

- For Software Configuration Management for the Web following tools may be used.
- GitLab. GitLab
- SourceForge. ...
- Cloud Source Repositories. ...
- GitKraken. ...
- Apache Allura.
- SolarWinds Server Configuration Monitor.
- CFEngine Configuration Tool.
- Puppet Configuration Tool.
- CHEF Configuration Tool.
- Ansible Configuration Tool.
- SALTSTACK Configuration Tool.
- JUJU Configuration Tool.

Image source : Google

DIGITAL LEARNING CONTENT



Parul® University

www.paruluniversity.ac.in