Software Configuration Management, Quality Assurance and Maintenance

- Risk Analysis and Management
  - Risk Mitigation
  - Monitoring
  - Management Plan (RMMM)
- Quality Concepts and Software Quality Assurance Metrics
  - Formal Technical Reviews
  - Software Reliability
- Software Configuration Management (SCM)
  - Version Control
  - Change Control

Risk

- Risk is a potential problem – it might happen, it might not happen.
  - Risk concerns future happenings.
  - It involves change.
  - It involves choice and the uncertainty that choice entails itself.

- Thus, risk denotes uncertainty that may occur in the choices due to past actions.
- It is something which causes heavy losses.
- We cannot eliminate the risk properly, but we can try to minimize it.

- Risk Management refers to the process of making decisions based on an evaluation of the factors that are threats to the business.

Software Risk

- Software Risks involves two characteristics:
  - i. Uncertainty
    - Risk may or may not happen
    - i.e. there are no 100% probable risks
  - ii. Loss
    - If the risk becomes reality, unwanted losses or consequences will occur

- When risks are analyzed, it is important to quantify the level of uncertainty and the degree of loss associated.

Software Risk …. Contd.

- Risks are identified, classified and managed before the actual execution of the program.
- These Risks are classified into different categories as:

1. Schedule Risk
   - Project schedules get slipped when project tasks and schedule release risks are not addressed properly.
   - Schedule risks mainly affect a project and finally on the company's economy and may lead to project failure.
   - Schedules often slip due to the following reasons:
     - Wrong/ Improper time estimation.
     - Resources are not tracked properly. All resources like staff, systems, skills of individuals, etc.
     - Failure to identify complex functionalities and time required to develop those functionalities.
     - Unexpected project scope expansions.

Categories of Risk …. Contd.

2. Budget Risk/ Financial Risk
   - These are the monetary risks which are associated with budget overruns.
   - Some of the reasons for such risks are:
     - Wrong/ improper budget estimation.
     - Cost overruns due to underutilization of resources.
       - ✔ This happens when resources are shared between projects because it becomes difficult to effectively manage such resources and a certain amount of productivity may go waste.
     - Expansion of project scope.
     - Improper tracking of finances.
     - Delay of projects may also have certain penalty costs associated with them (e.g. construction projects).

Categories of Risk …. Contd.

3. Operational Risk/ Procedural Risk
   - These are the risks which are associated with day-to-day operational activities of the project.
   - Some of the reasons for such risks are:
     - Risk of loss due to improper process implementation, failed system or some external event risks.
     - Failure to address priority conflicts.
     - Failure to resolve responsibilities.
     - Insufficient resources.
     - No proper subject training.
     - No resource planning
     - No communication within the team.

……..contd.

Categories of Risk …. Contd.

4. Technical/ Functional/ Performance Risk
   - It threatens the quality and timeliness of the software to be produced.
   - If a technical risk becomes a reality, implementation may become difficult and impossible.
   - Technical risks generally lead to failure of functionality and performance.
   - Technical risks occur when the problem is harder to solve than you thought it would be.
   - Causes of technical risks are:
     - Continuously changing requirements.
     - No advanced technology is available or the existing technology is in the initial stages.
     - The product is complex to implement.
     - Difficult project module integration.

Categories of Risk …. Contd.

5. Programmatic Risk
   - These are external risks beyond the operational limits. These are all uncertain risks that are outside the control of the program.
   - External events can be:
     - Running out of funds.
     - Market development
   - Changing customer product strategy and priorities.
     - Government rule changes.

……..contd.

Categories of Risk …. Contd.

6. Project Risk
   - Project risks concern differ forms of budgetary, schedule, personnel, resource, stakeholders, project complexity and size, degree of structural uncertainty and customer-related problems.
   - It threatens the project plan.
     - If the risk becomes real, it is likely that the project schedule will slip and the cost will increase.
   - A vital project risk is schedule slippage.
   - Since the software is intangible, it is very tough to monitor and control a software project.
   - It is very tough to control something which cannot be identified.

Categories of Risk …. Contd.

7. Business Risk
   - This type of risks contain risks of building an excellent product that no one needs, losing budgetary or personnel commitments, etc.
   - It threatens the viability of the software to be built and often jeopardize the project or the product.
   - The top five business risks are:
     i. Market Risk
        - Building an excellent product or system that no one really wants.

     ii. Strategic Risk
        - Building a product that no longer fits into the overall business strategy for the company.

Categories of Risk …. Contd.

iii. Sales Risk
- Building a product that the sales force does not understand how to sell.

iv. Management Risk
- Losing the support of senior management due to change in focus or change in people.

v. Budget Risk
- Losing budgetary or personnel commitment.

9. Other risk categories

a. Known Risk
- Those risks that can be uncovered after careful assessment/ evaluation of the project plan, the business and technical environment in which the project is being developed, and more reliable information/ data sources.

Categories of Risk …. Contd.

- Ex. unrealistic delivery date, lack of documented requirements, poor development environment.

c. Predictable Risk
   - Those risks that are hypothesized/ extrapolated from past/ previous project experience.
   - Ex. past turnover, poor communication with the customer, dilution of staff effort as on going maintenance requests are serviced.

d. Unpredictable Risk
   - Those risks that can and do occur, but are extremely difficult/ tough to identify in advance.
   - Ex. Joker in a deck of cards.

Strategies of Risk

❖ Following are the Risk strategies for managing risks:

A. *Reactive Risk Strategy*

- It is a risk management strategy in which, when a project gets into trouble then only corrective action is taken.
- It monitors the project for likely risks.
- Software team does nothing about risks until something goes wrong. Then the team attempts to correct the problem.
  - This is often called a fire fighting mode.
  - When such risks cannot be managed and new risks come up one after the other, the software team flies into action in an attempt to correct the problem rapidly.
- Resources are set aside to deal with risks when they become actual problem.
- When the team fails to solve the problem, "crisis management" takes over and the project is in real jeopardy.

Strategies of Risk …. Contd.

B. *Proactive Risk Strategy*

- It is better than reactive risk strategy.
- It begins long before technical work is initiated.
- Potential risks are identified, their probability and impacts are assessed and ranked by importance.
- The software team establishes a plan for managing the risks.
- The objective is to avoid/ prevent the risk.
- As all the risks cannot be avoided, the team works to develop a contingency plan that will enable it to respond in a controlled and effective manner.

Negative Impact of Risk

- If the risk is not identified and resolved in timely manner it can create a great negative impact:
  - Diminished quality of product
  - Increased cost
  - Delayed completion
  - Project failure

Risk Analysis and Management

- Risk Analysis in project management is a sequence of processes to identify the factors that may affect a project's success.
  - These processes include risk identification, analysis of risks, risk management and control, etc.
  - Proper risk analysis helps to control possible future events that may harm the overall project.
  - It is more of a pro-active than a reactive process.

- Risk is an undesired event or circumstance that occur while a project is underway.
  - It is necessary for the project manager to anticipate and identify different risks that a project may be susceptible to.
- Risk Management aims at reducing the impact of all kinds of risk that may affect a project by identifying, analyzing and managing them.

Risk Management

- Definition of a "risk" is a problem that could cause some loss or threaten the progress of the project, but which has not happened yet.
- “Risk is an uncertain future event with a probability of occurrence and potential for loss”
  - These potential issues might harm cost, schedule or technical success of the project and the quality of our software device, or project team morale.
- Risk Management is the system of identifying, addressing and eliminating these problems before they can damage the project.
  - Risk identification and management are the main concerns in every software project.
  - Effective analysis of software risks will help in effective planning and assignment of work.

Risk Management …. Contd.

- Risk management is the area which tries to ensure that the impact of risks on cost, quality and schedule is minimized.
- The main purpose of risk management is to identify and manage the risks associated with a software project and solve the problem.
- Estimating the risks that can affect the project schedule or the quality of the software being developed and taking action to avoid the risk is the important task of a project manager.
- Identifying and preparing plans to reduce their impact on the project is called risk management.
- The basic motivation of risk management is to avoid disaster or heavy losses.

Risk Management …. Contd.

- The process of risk management involves several stages are as follows-
  - a) Risk Identification :
    - In this stage, the possible project, product and business risks are identified.
  - b) Risk Analysis :
    - In this stage or process, the likelihood and consequences of these risks assessed.
  - c) Risk Planning :
    - In this stage, risk avoidance in either planned to affect the plan or mitigate its effects on the project.
  - d) Risk Monitoring :
    - In this stage, risk assessment is done continuously and the risk reduction plan is revised as more information about risk is available.

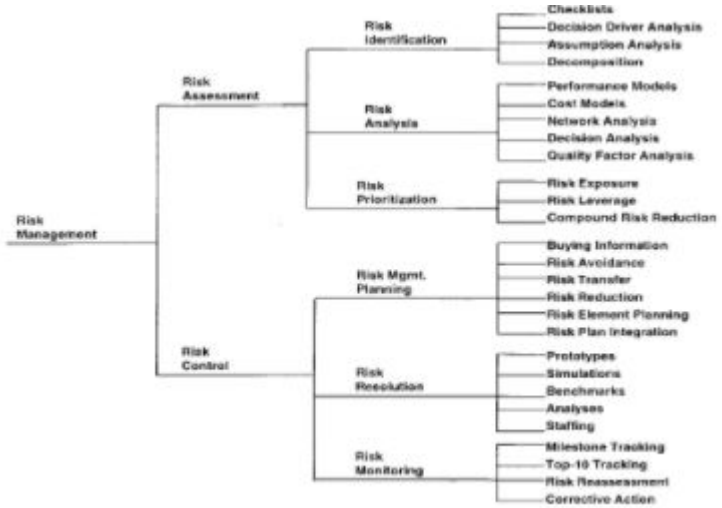
Risk Management …. Contd.

Risk Management Activities

- Risk Management basically includes the following activities:
  - Risk Assessment
  - Risk Identification
  - Risk Containment

Risk Management Activities

- Risk Management basically includes the following activities:
  - Risk Assessment
  - Risk Identification
  - Risk Containment



Risk Management Activities

- Risk Management basically includes the following activities:
  - Risk Assessment
    - The objective of risk assessment is to rank the risks in terms of their damage causing potential.
    - For risk assessment, each risk should first be rated in two ways:
      - The likelihood of a risk coming true (r)
      - The severity of damage caused due to the risk (s)
    - Based on these factors, the priority of each risk can be computed as p= r * s
  - Risk Identification
  - Risk Containment

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 towards avoiding them when possible and controlling them when necessary.
- He/ she needs to anticipate the risks in the project as early as possible so that the impact of the risk can be minimized by making effective risk management plans.
- In order to be able to systematically identify the important risks, it is necessary to categorize risks into different classes.
  - Product Specific Risks
    - Identification of scope and special characteristics of your software.

Risk Identification …. Contd.

- Can be specified by those who have clear understanding of current technology, people, environment, market situation, etc. that is specific to software that is to be built.

❑ Generic Risks

- They are potential threats to all software projects.

❖ Types of common risks

1. Technology risks
   - Risks that derive from the software or hardware technologies that are used to develop the system.
2. People risks
   - Risks that are associated with the people in the development team.

Risk Identification …. Contd.

3. Organizational risks
   - Risks that derive from the organizational environment where the software is being developed.
4. Tools risks
   - Risks that derive from the software tools and other support software used to develop the system.
5. Requirements risks
   - Risks that derive from the changes to the customer requirements and the process of managing the requirements change.
6. Estimate risks
   - Risks that derive from management estimates of the resources required to build the system.

Risk Identification …. Contd.

- One method is to create risk checklist.
- The project manager follows a checklist for risk identification:
  a. Product size:
     - The risk items based on overall size of the software product is identified.
  b. Business impact:
     - Risk items related to marketplace or management can be predicted.
  c. Customer characteristics:
     - Risks associated with customer-developer communication can be identified.
     - It includes customer interests, knowledge and developers ability to communicate.

Risk Identification …. Contd.

d. Process definition:
   - Risks that get raised with the definition of software process.
   - This category exposes important risk items because whichever is the process definition made, it is then followed by the whole team.

e. Development environment:
   - The risks associated with the technology and tools being used for developing the product.

f. Technology to be built:
   - The overall complexity of the system should be understood and related risk items need to be identified.

g. Staff size and experience:
   - Once the technology and tools related risk items are identified, it is essential to identify the risk associated with sufficiently highly experienced and skilled staff who will do the development.

Risk Identification …. Contd.

- After preparing a risk item checklist, a questionnaire is prepared.
  - These set of questions should be answered and based on these answers, the impact or seriousness of a particular risk item can be judged.
- Later, the set of risk components and drivers list is prepared along with their probability of occurrence. Then their impact on the project can be analyzed.
  - ❖ Risk components can be as follows:
    - Performance risk:
      - It is the degree of uncertainty that the product will satisfy the requirements.
    - Cost risk:
      - It is the degree of uncertainty that the project will maintain the budget.

Risk Identification …. Contd.

- Support risk:
  - It is the degree of uncertainty that the software project being developed will be easy to correct, modify or adopt.
- Schedule risk:
  - It is the degree of uncertainty that the software project will maintain the schedule and the project will be delivered in time.

- Associated with these components are the risk drivers that are used to analyze the impact of risk. They are:
  - Negligible ……. Impact value 1
  - Marginal ……. Impact value 2
  - Critical ……. Impact value 3
  - Catastrophic ……. Impact value 4

Risk Analysis

- Risk analysis can be divided as:

I. Qualitative Risk Analysis

- It is a procedure in which assignment of priorities to risk can be done according to their possibility of occurrence and their effort.
- It supports the managers to decrease the uncertainty level and focuses on risks which have high priority.
- Create Plan for risk management as early as possible in the project. It can impact on different factors like cost, time, scope, quality and procurement.
- The inputs of qualitative analysis involves –
  a. Risk management plan
  b. Scope baseline
  c. Risk register

Risk Analysis …. Contd.

- Enterprise environmental factors are –
  - » Organizational process assets
- The output of this stage would be –
  - » Project document updated

III. Quantitative Risk Analysis

- It is the process of mathematically analyzing the impact of recognized risks on complete project purpose.
- To minimize uncertainty, this type of analysis is very useful for decision making.
- The input to this stage is –
  - a. Risk Management plan
  - b. Cost Management plan

……..contd.

Risk Analysis …. Contd.

c. Schedule Management plan
d. Risk register

- Enterprise environmental factors –
  - » Organizational process assets
- The output will be –
  - » Project documents updates

Risk Assessment

- The best approach is to prepare a set of questions that can be answered by the project managers in order to assess the overall project risks.
- The questions are ordered by their relative importance to the success of a project.
- These questions can be –
    1) Have the top software and customer managers formally committed to support the project?
    2) Are end-users enthusiastically committed to the project and the system/ product to be built?
    3) Are requirements fully understood by the software engineering team and its customers?
    4) Have customers been involved fully in the definition of the requirements?
    5) Do end-users have realistic expectations?
    6) Is the project scope stable?

Risk Assessment ….Contd.

7) Does the software engineering team have the right mix of skills?
8) Are the project requirements stable?
9) Does the project team have experience with the technology to be implemented?
10) Is the number of people on the project team adequate to do the job?
11) Do all customers/ users constituencies agree on the importance of the project and on the requirements of the system/ product to be built?

- If any one of these questions is answered negatively, mitigation, monitoring and management steps should be instituted without fail.
- The degree to which the project is at risk is directly proportional to the number of negative responses to these questions.

Risk Assessment ….Contd.

- The objective of risk assessment is to rank the risks in terms of their damage causing potential.
- For risk assessment, each risk should first be rated in two ways:
  - The likelihood of a risk coming true (r)
  - The severity of damage caused due to the risk (s)
- Based on these factors, the priority of each risk can be computed as
  p= r * s

  where p is the priority with which the risk must be controlled,

  r is the probability of the risk becoming true, and

  s is the severity of loss caused due to the risk becoming true.
- If all identified risks are set up, then the most likely and damaging risks can be controlled first, and more comprehensive risk abatement methods can be designed for these risks.

Risk Projection/ Estimation

- There are two ways by which risk can be rated –
  i. Likelihood or probability that the risk is real
  ii. Consequences of the problems associated with the risks, should they occur

- The project planner along with the other managers and technical staff, performs four risk projection steps:-
  a) Estimates a scale that reflects the perceived likelihood / probability of a risk.
  b) Enlist the consequences of the risk.
  c) Estimate the impact of the risk on the project and product.
  d) Assess/ maintain the overall accuracy of the risk projection so that there will be no misunderstanding.

Risk Projection/ Estimation ….

- The intent of the above steps is to prioritize the risks.
  - No software team has the resources to address every possible risk with the same degree of rigor.
  - By prioritizing risks, the team can allocate resources where they will have the most impact.
- Building risk table is the common method adopted by project managers in order to project/ estimate the risks.–
  - ✔ A risk table provides simple techniques for risk projection.
  - ✔ Begin by listing all the risks in the table with the help of risk item checklist.
  - ✔ Each risk is categorized in the form of table.
  - ✔ Probability of occurrence of each risk is entered in the adjacent/ next column in the table.

Risk Projection/ Estimation ….

Following is the sample risk table:

Risk	Category	Probability	Impact	RMMM
Is skilled staff available?	Staff	50%	Catastrophic (4)	
Is the team size sufficient?	Staff	62%	Critical (3)	
Has the staff received sufficient training?	Staff	25%	Marginal (2)	
Will technology meet the expectations?	Technology	30%	Critical (3)	
Is software management tool available?	Environment	40%	Negligible (1)	
How much amount of reused software is required?	Project size	60%	Marginal (2)	
Will the customer change requirements?	Customer	20%	Critical (3)	

Risk Projection/ Estimation ….

❖ While building the risk table,

i. The project team first enlists all the probable risks with the help of risk item checklist.

ii. Each risk is then categorized as
- Project size
- Technology
- Customer
- Staff
- Business
- Developing Environment

iii. Probability of occurrence of each risk is then estimated by each team member individually.

iv. Then, impact of each risk is assessed.
- It is calculated using cost drivers.

Risk Projection/ Estimation ….

v. Risk is sorted by probability and impact.
   - High risk impact and high probability is at the top of first-order prioritization table.

vi. The project manager goes through first-order prioritized risk table and draws a cut off line.
   - Risks above cut off line are considered for further risk analysis.

vii. Risks below cut off line are sorted again and second-order prioritization is applied on this table.

viii. Using Risk Mitigation, Monitoring and Management plan, the last column of the risk table is filled up.

Risk Projection/ Estimation ….

❖Assessing Risk Impact:-

- While assessing risk impact, three factors are considered –

a) Nature of risk

- It indicates the problems that are likely to come across, if it occurs.
- i.e.; the type or kind of risk.
- Ex.
  - If software requirement is poorly understood, the software processes get poorly designed and ultimately it will create a problem in unit testing.

b) Scope of the risk

- It indicates the severity of the risk.
  - how serious it is?

……. contd.

Risk Projection/ Estimation ….

- how much of the project will be affected?
- how many customers will be harmed?

d) Timing at which risk occurs
   - It indicates when and for how long the impact will be felt.
     - i.e.; determining at which phase of software development life cycle the risk will occur and how long it will persist.

- Risk Exposure (RE):
  - RE = Probability of occurrence of risk * Cost
  - Cost = Number of components * LOC * Cost of each LOC

– Thus, the total risk exposure of all risks helps in determining the final cost of the project.

Risk Containment

- Risk Containment is the process of keeping track of noticed risks, finding new risks, monitoring residual risks and analyzing the risk.
- The inputs for this phase are –
    a) Project Management plan
    b) Risk register
    c) Work performance data
    d) Work performance reports
- The outputs for this phase are –
    a) Work performance information
    b) Change requests
    c) Project plan management updates
    d) Project documents updates
    e) Organizational process assets updates

RMMM

- RMMM stands for Risk Mitigation, Monitoring and Management

- Following strategies are developed for dealing with risk:-
  1. Risk Mitigation/ Avoidance
     - If a software team adopts a proactive approach to risk, avoidance is always the best strategy.
     - Risk Mitigation means preventing the risks from occurring (avoidance).
     - Following steps are taken for mitigating the risk –
       - Meet the current staff to determine the causes of probable risks.
         - Ex. Cause for turnover (poor working conditions, low pay, competitive job market, etc.)

RMMM - Mitigation

- Find out and eliminate all those causes that can create risk before the project starts/ begins.
- Develop a policy in an organization which will help to continue the project even though some staff leaves the organization.
- Everybody in the project team should be acquainted with the current development activity.
- Maintain the corresponding documents in timely manner.
  - This documentation should be strictly as per the standards set by the organization.
- Conduct timely reviews in order to speed up the work.
- For conducting very critical activity during software development, provide additional staff, if required.

RMMM - Monitoring

II. Risk Monitoring

In risk monitoring process, following things must be monitored by the project manager –

- The approach or behavior of the team members as pressure of the project varies.
- The degree in which the team performs with the spirit of "team-work".
- The type of cooperation among the team members.
- The type of problems that are occurring.
- Availability of jobs within and outside the organization.
- The objective of risk management is:
  a) to check whether the predicted risks really occur or not.

…….contd.

RMMM - Management

b) to ensure the steps defined to avoid the risk are applied properly or not.

c) to gather the information which can be useful for analyzing the risks.

IV. Risk Management

- Risk management and contingency planning assumes that mitigation efforts have failed and that risk has become a reality.
- If project manager is successful in applying the project mitigation effectively, then it becomes easy to manage the risks.
- Ex.
  - Consider a scenario that many people are leaving the organization.

……. contd.

RMMM – Management ……

- In this case, if
  - ✔ sufficient additional staff is available,
  - ✔ current development activity is known to everyone in the team,
  - ✔ latest and systematic documentation is available,
- Then any ‘new comer’ can easily understand the current development activity.
- This will ultimately help in continuing the work without any interruption or delay.

RMMM Plan

- The RMMM plan may be a part of the software development plan.
  - It documents all work performed as part of risk analysis.
  - It is used by the project manager as part of the overall project plan.
  - RMMM plan may take any of the following forms:
    - ✔ Separate document
    - ✔ Risk Information Sheet (RIS)

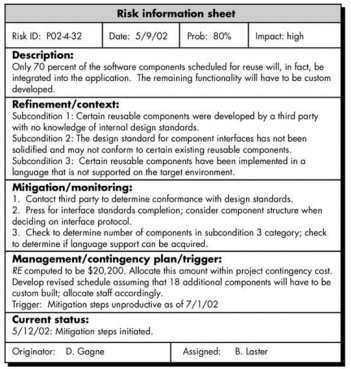
- Some software teams do not develop a formal RMMM document.
  - Rather, each risk is documented individually using a risk information sheet (RIS).
  - In most cases, RIS is maintained using a database system, so that creation and information entry, priority ordering, searches and other analysis may be accomplished easily.

RMMM Plan ….Contd.

- Once RMMM has been documented and the project has begun, risk mitigation and monitoring steps commence.
  - Risk mitigation is a problem avoidance activity.
    - It is achieved by developing a plan.
  - Risk monitoring is a project tracking activity.
    - The primary objectives of monitoring are:
      i. To assess whether predicted risks do, in fact, occur.
      ii. To ensure that risk aversion steps defined for the risk are being properly applied.
      iii. To collect information that can be used for future risk analysis.
- The findings from RMMM plan may tell the project manager to ascertain what risks caused which problems throughout the project.

Risk Information Sheet (RIS)

- Risk Information Sheet (RIS) contains the following information:
  - ✔ Risk ID, Date, Probability and Impact
  - ✔ Description
  - ✔ Refinement/ Context
  - ✔ Mitigation/ Monitoring
  - ✔ Management/ Contingency Plan/ Trigger
  - ✔ Current Status
  - ✔ Originator and Assigned (to whom) information



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 Information Sheet (RIS)

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

RMMM

- **Drawbacks of RMMM:**
  - It incurs additional project costs.
  - It takes additional time.
  - For larger projects, implementing an RMMM may itself turn out to be another tedious project.
  - RMMM does not guarantee a risk-free project, in fact, risks may also come up after the project is delivered.

Quality Concepts

- Quality is a complex concept - it means different things to different people, and it is highly context dependent .
- Following are the five different views of quality:

i. Transcendental View :

- It envisages quality as something that can be recognized but is difficult to define.
- Here quality is something that can be recognized through experience but is not defined in some tractable form.
- Quality is viewed to be something ideal, which is too complex to lend itself to be precisely defined.

ii. User View:

- It perceives quality as fitness for purpose.
- According to this view, while evaluating the quality of the product, one must ask the key question, “Does the product satisfy user needs and expectations?”

Quality Concepts ….Contd.

- In this view, a user is concerned with whether or not a product is fit for use.
- Quality is not just viewed in terms of what a product can deliver, but it is also influenced by the service provisions in the sales contract.

iii. Manufacturing View :

- Here quality is understood as conformance to the specifications.
- The quality level of a product is determined by the extent to which the product meets its specifications.
- Any deviation from the stated requirements is seen as reducing the quality of the product.
- The manufacturing view has its genesis in the manufacturing sectors, such as the automobile and electronics sectors.

Quality Concepts ….Contd.

iv. Product View :

- In this case, quality is viewed as tied to the inherent characteristics of the product.
- A product's internal qualities determine its external qualities.
- The product view is attractive because it gives rise to an opportunity to explore casual relationships between internal properties and external qualities of a product.
- An example of the product view of software quality is that high degree of modularity, which is an internal property, makes a software testable and maintainable.

Quality Concepts ….Contd.

v. Value - Based View :

- Quality, in this perspective, depends on the amount the customer is willing to pay for it.
- The value based view represents a merger of two independent concepts: excellence and worth where Quality is the measure of excellence and value is the measure of worth.

- ❑ Quality is meaningless if a product does not make economic sense.
- ❑ The value based view represents a trade- off between cost and quality.

Quality Concepts ….Contd.

Software Quality Management (SQM) is a method that guarantees that required level of software is achieved, so that the users are satisfied by its performance.

- The process involves quality assurance, quality planning and quality control.
- This concept provides a complete understanding of SQM and illustrates the various steps involved in the process.
- SQM will make sure that the necessary level of quality is achieved by correcting the improvements to the product development process.
- The main motive behind the SQM is to develop an ethnicity within the team and it is seen as everyone's liability.
- SQM should be independent of project management to ensure the self-determination of cost and schedule adherence.
- Its effect directly impacts the process quality and indirectly affects the product quality.

Quality Concepts ….Contd.

Activities of Software Quality Management:

- ***Quality Assurance (QA)***
  - ✔ It is a planned and systematic pattern of activities necessary to provide a high degree of confidence in the quality of product.
  - ✔ It provides quality assessment of the quality control activities and determines the validity of the data or procedures for determining quality.
  - ✔ QA consists of a set of reporting and auditing functions.
    - These functions are useful for assessing and controlling the effectiveness and completeness of quality control activities.
  - ✔ It aims at building measure and standards for maintaining quality at organizational level.

…….contd.

Quality Concepts ….Contd.

- ***Quality Planning***
  - ✔ To choose applicable procedures/ actions and standards for a certain project, changes can be done at required to develop a quality plan.

- ***Quality Control***
  - ✔ It is a process in which activities are conducted in order to maintain the quality of the product.
  - ✔ It involves a series of inspections, reviews and tests used throughout the software process.
    - This ensures that each work product meets the requirements placed on it.
  - ✔ Quality control includes a feedback loop to the process that created the work product. ……contd.

Quality Concepts ….Contd.

- With the help of feedback, we can tune the process, if it does not satisfy the requirements.
  - Feedback loop helps in minimizing the defects in the software product.
- Quality control activities can be fully automated or it can be completely manual or it can be a combination of automated tools and manual procedures.

McCall's Quality Model

McCall software quality model was introduced in 1977.

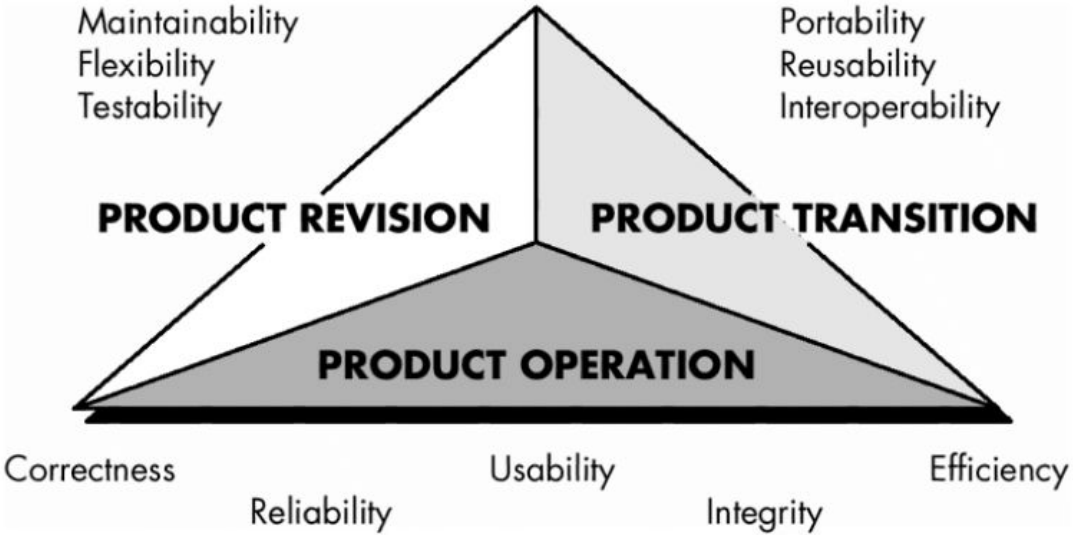
- This model is incorporated with many attributes, termed as software factors, which influence a software.
- The model distinguishes between two levels of quality attributes :

A. Quality Factors –
   - The higher level quality attributes which can be assessed directly are called quality factors.
   - These attributes are external attributes.
   - The attributes in this level are given more importance by the users and managers.

B. Quality Criteria –
   - The lower or second level quality attributes which can be assessed either subjectively or objectively are called Quality Criteria.
   - These attributes are internal attributes.
   - Each quality factor has many second level of quality attributes or quality criteria.

McCall's Quality Model …..

- This model classifies all software requirements into 11 software quality factors.
- The 11 factors are organized into three product quality factors – product operation, product revision, and product transition factors.



McCall's Quality Model …..

Following are the product quality factors –

1)Product Operation :

- It includes five software quality factors, which are related with the requirements that directly affect the operation of the software such as operational performance, convenience, ease of usage and its correctness.
- These factors help in providing a better user experience.
  - Correctness
    - The extent to which a software meets its requirements specification.
  - Efficiency
    - The amount of hardware resources and code the software, needs to perform a function.

McCall's Quality Model …..

- Integrity
  - The extent to which the software can control an unauthorized person from the accessing the data or software.
- Reliability
  - The extent to which a software performs its intended functions without failure.
- Usability
  - The extent of effort required to learn, operate and understand the functions of the software.

2) Product Revision :

•It includes three software quality factors, which are required for testing and maintenance of the software.

McCall's Quality Model …..

- They provide ease of maintenance, flexibility and testing effort to support the software to be functional according to the needs and requirements of the user in the future.
  - Maintainability
    - The effort required to detect and correct an error during maintenance phase.
  - Flexibility
    - The effort needed to improve an operational software program.
  - Testability
    - The effort required to verify a software to ensure that it meets the specified requirements.

…….contd.

McCall's Quality Model …..

3) Product Transition :

- It includes three software quality factors, that allows the software to adapt to the change of environments in the new platform or technology from the previous.
  - Portability
    - The effort required to transfer a program from one platform to another.
  - Re-usability
    - The extent to which the program's code can be reused in other applications.
  - Interoperability
    - The effort required to integrate two systems with one another.

Cost of Quality

The cost of quality can be defined as the total cost required to obtain the quality in the product and to conduct the quality related activities.

- Quality cost may be into costs associated with prevention, appraisal and failure.

1)Prevention Cost:

- It is the cost of quality required for conducting quality planning, formal technical reviews, test equipment and training.

2)Appraisal Cost:

- It is the cost of quality required for gaining the insight into the product.
- It includes the cost required for in-process and inter-process inspection, maintenance and testing.

Cost of Quality

3) Failure Cost:
   - It is the cost of quality required to remove the defects in the product before delivering it to the customer.
   - It is further divided into two types –
     a) Internal Failure Cost
        - It is nothing but the cost of defects occurred in the product before delivering it to the customer.
        - It includes rework, repair and failure mode analysis.
        - Ex. Repair in some kind of networking, repairing of communication network.
     b) External Failure Cost
        - It is the cost of defects occurred in the product after delivering it to the customer.
        - Ex. Complaint resolution, product return and replacement, help line support and warranty work.

Software Quality Failure

❖ Following are the reasons for Software Quality Failure:

i. Software requirements must be well understood before software development process begins.

ii. It is essential to understand the implicit requirements of the software in a similar manner as the explicit requirements.
   - If the software confirms the explicit requirements but does not satisfy the implicit requirements, then the quality of the software being developed will surely be poor.

iii. A set of development criteria has to be decided in order to specify the standards of the product.
   - This is useful during the development for the software engineer.
   - If such a criteria is not fixed, then the software product will lack quality.

Software Quality Assurance

Software Quality Assurance is the process in which conformance to the requirements of the product is made.

Software Quality Assurance (SQA) Activities:

- SQA is composed of a variety of tasks associated with two different constituencies –
  - A. Software Engineer(s)
    - They do/ perform the technical work.
    - They are responsible for developing the product.
    - They address quality (i.e. perform quality assurance and quality control activities) by applying solid technical methods and measures, conducting formal technical reviews and performing well-planned software testing.

…….. contd.

Software Quality Assurance …..

B. SQA Group

- Role of SQA group is to assist the software team in achieving a high-quality end product.
- SQA activities address the following –
  - ✔ Quality assurance planning
  - ✔ Oversight
  - ✔ Record keeping
  - ✔ Analysis and reporting
- Following SQA activities are performed (or facilitated) by an independent SQA group:-
  1. Prepares an SQA plan for a project
     - The plan is developed during project planning and is reviewed by all the stakeholders.

……..contd.

Software Quality Assurance …..

- Quality assurance activities performed by the software engineering team and the SQA group are governed by the plan.
- This plan
  - » Identifies evaluations to be performed
  - » Audits and reviews to be conducted, standards that should be adopted for the project
  - » Procedures for error reporting and tracking
  - » Documents to be produced by the SQA group
  - » Amount of feedback provided to the software project team

3. Participates in the development of the project's software process description
   - The software team selects a process for the work to be performed. …….contd.

Software Quality Assurance …..

- The SQA group reviews the process description for
  - Ensuring that it follows the organizational policy
  - Internal software standards
  - Explicitly imposed standards (adopted by the organization)

4. Reviews software engineering activities to verify compliance with the defined software process
   - SQA group identifies and documents the processes.
   - They track deviations from the process and verifies that correctness have been met.
5. Audits designated software work products to verify compliance with those defined as part of the software process
   - SQA group reviews selected work products.
   - Identifies the processes and documents them.
   - Verifies the correctness made in the processes.

…contd.

Software Quality Assurance …..

- Tracks deviations.
- Periodically reports the results of its work to the project manager.

6. Ensures that deviations in the software work and work products are documented and handled according to a documented procedure
   - The deviations in the software work are identified from the project plan.
   - These processes are identified and handled according to a documented procedure
7. Records any non-compliance and reports to senior management
   - Non-compliance items are identified and pursued until they get resolved.
   - The periodic reporting about it is done to the project manager.

Formal Technical Review (FTR)

- Formal Technical Review (FTR) is a software quality control activity performed by software engineers.

- Objectives of formal technical review (FTR):
  - Useful to uncover error in logic, function and implementation for any representation of the software.
  - The purpose of FTR is to verify that the software meets specified requirements.
  - To ensure that software is represented according to predefined standards.
  - It helps to review the uniformity in software that is development in a uniform manner.
  - To makes the project more manageable.

Formal Technical Review (FTR) …..

- FTR (Formal Technical Review) is also a learning ground for junior developers to know more about different approaches to software analysis, design and implementation.
- It also serves as a backup and continuity for the people who are not exposed to the software development so far.

- Actually, FTR is a class of reviews that include walkthroughs, inspections, round robin reviews and other small group technical assessments of software.
- Each FTR is conducted as meeting and is considered successfully only if it is properly planned, controlled and attended.

…….contd.

Formal Technical Review (FTR) …..

❖ **Steps in FTR:-**

a) *The review meeting*:

- Each review meeting should be held considering the following constraints-
  - Every review meeting should be conducted by considering the following constraints- Involvement of people Between 3 and 5 people should be involved in the review.
  - Advance preparation should occur but it should be very short that is at the most 2 hours of work for each person can be spent in this preparation.
  - The short duration of the review meeting should be less than two hour.
- Gives these constraints, it should be clear that an FTR focuses on specific (and small) part of the overall software.

Steps in FTR

b) *Decision Making*:

- At the end of the review, all attendees of FTR must decide what to do.
  i. Accept the product without any modification.
  ii. Reject the project due to serious error (Once corrected, another app need to be reviewed), or
  iii. Accept the product provisional (minor errors are encountered and are should be corrected, but no additional review will be required).
- The decision is made, with all FTR attendees completing a sign-of indicating their participation in the review and their agreement with the findings of the review team.

……contd.

Steps in FTR …..contd.

c) *Review reporting and record keeping*:

- During the FTR, the reviewer actively records all issues that have been raised.
- At the end of the meeting all these issues raised are consolidated and a review list is prepared.
- Finally, a formal technical review summary report is prepared.
- It answers three questions :-
  - What was reviewed ?
  - Who reviewed it ?
  - What were the findings and conclusions ?

d) *Review guidelines :-*

- Guidelines for the conducting of formal technical reviews should be established in advance. ……..contd.

Steps in FTR …..contd.

- These guidelines must be distributed to all reviewers, agreed upon, and then followed.
- A review that is unregistered can often be worse than a review that does not have minimum set of guidelines for FTR.
  i. Review the product, not the manufacture (producer).
  ii. Take written notes (record purpose)
  iii. Limit the number of participants and insists upon advance preparation.
  iv. Develop a checklist for each product that is likely to be reviewed.
  v. Allocate resources and time schedule for FTRs in order to maintain time schedule.
  vi. Conduct meaningful training for all reviewers in order to make reviews effective.

……contd.

Steps in FTR …..contd.

vii. Reviews earlier reviews which serve as the base for the current review being conducted.
viii. Set an agenda and maintain it.
ix. Separate the problem areas, but do not attempt to solve every problem notes.
x. Limit debate and rebuttal.

- **Walkthrough**
  - Walkthrough is a method of conducting informal group/individual review.
  - In a walkthrough, author describes and explain work product in a informal meeting to his peers or supervisor to get feedback.
  - Here, validity of the proposed solution for work product is checked.

……contd.

Walkthrough/ Inspection

- It is cheaper to make changes when design is on the paper rather than at time of conversion.
- Walkthrough is a static method of quality assurance.
- Walkthrough are informal meetings but with purpose.

Inspection:

- An inspection is defined as formal, rigorous, in depth group review designed to identify problems as close to their point of origin as possible.
- Inspections improve reliability, availability, and maintainability of software product.
- Anything readable that is produced during the software development can be inspected.

…….contd.

Inspection

- Inspections can be combined with structured, systematic testing to provide a powerful tool for creating defect-free programs.
- Inspection activity follows a specified process and participants play well-defined roles.
- An inspection team consists of three to eight members who plays roles of moderator, author, reader, recorder and inspector.
- Ex.
  - A designer can act as inspector during code inspections while a quality assurance representative can act as standard enforcer.

- Stages in the inspections process :
  - Planning : Inspection is planned by moderator.
  - Overview meeting : Author describes background of work product.

…..contd.

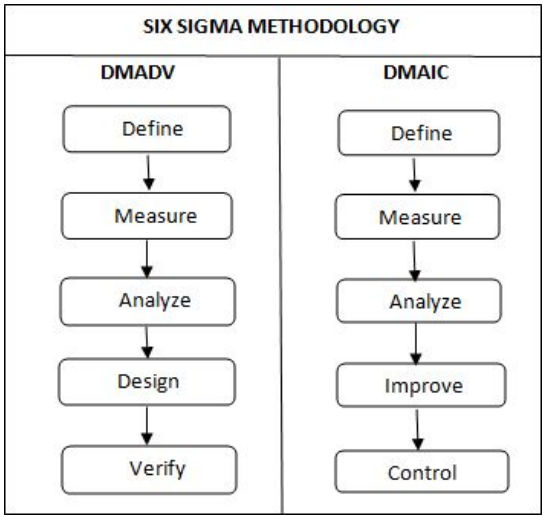
Inspection …..Contd.

- Preparation : Each inspector examines work product to identify possible defects.
- Inspection meeting : During this meeting, reader reads through work product, part by part and inspectors points out the defects for every part.
- Rework : Author makes changes to work product according to action plans from the inspection meeting.
- Follow-up : Changes made by author are checked to make sure that everything is correct.

Six Sigma

- Six Sigma is the process of producing high and improved quality output.
- This can be done in two phases – identification and elimination.
- The cause of defects is identified and appropriate elimination is done which reduces variation in whole processes.
- A six sigma method is one in which 99.99966% of all the products to be produced have the same features and are of free from defects.

- Six Sigma Methodologies:



Six Sigma Methodologies

- Two methodologies used in the Six Sigma projects are DMAIC and DMADV.
  - DMAIC is used to enhance an existing business process.
  - The DMAIC project methodology has five phases:
    - Define
      - Customer requirements, project goals and deliverables are defined by communicating with the customers.
    - Measure
      - The existing process and its output is measured in order to determine the current quality performance.
    - Analyze
      - In this phase, defect metrics are analyzed in order to determine the vital few causes.

……contd.

Six Sigma Methodologies ….

- If an improvement is needed to an existing software, then there are additional two methods -
- Improve
  - By eliminating the root causes of defects, the process can be improved.
- Control
  - The process can be controlled in such a way that the causes of defects cannot be re-introduced.

❑ DMADV is used to create new product designs or process designs. The DMADV project methodology also has five phases:

- Define
  - It defines the problem or project goal that needs to be addressed.

….contd.

Six Sigma Methodologies ….

- Measure
  - It measures and determines the customer's needs and specifications.
- Analyze
  - It analyzes the process to meet customer needs.

❑For newly developing software, some organizations suggest the following two alternating steps –

- Design
  - It can design a process that will meet customer needs.
- Verify
  - It can verify the design performance and ability to meet customer needs.

Software Reliability

- Software Reliability is defined in statistical terms as “the probability of failure free operation of a computer program in a specified environment for a specified time”.
  - Failure indicates non-conformance to software requirements.

- Measures of Reliability and Availability:
  - A simple measure of reliability is mean-time-between-failure (MTBF),

where

MTBF = MTTF + MTTR

MTTF -> mean-time-to-failure

MTTR-> mean-time-to-repair

Software Reliability …..

- Software Availability is the probability that a program is operating according to requirements at a given point in time and is defined as:

Availability = [MTTF/ (MTTF + MTTR)]*100%

- The MTBF reliability measure is equally sensitive to MTTF and MTTR.
- The availability measure is somewhat more sensitive to MTTR, an indirect measure of the maintainability of software.

Quality Metrics

The goal of software engineering is to produce high quality software.

- To achieve this goal, software engineers use effective methods along with modern tools while developing the software.
- Quality of the software depends on –
  - ✔ Requirements that describe the problem
  - ✔ Design method used to produce the software
  - ✔ Code that leads to executable program
  - ✔ Tests that are carried out in order to uncover the errors from the software
- The project manager evaluates the quality of the software project using the following factors –
  - – Errors and defects in the software
  - – Quality metrics collected by each software engineer who is involved in the software development process
- Such an evaluation of software quality helps in improving quality assurance and control activities.

Quality Metrics ….Contd.

❖ Metrics used for assessing the software quality are:

a. Work product errors per function
b. Errors found in the per review hour
c. Errors found in testing

This error data is useful in computing the Defect Removal Efficiency (DRE).

- While developing the software project, many work products such as SRS, design document, source code, etc. are being created.
- Along with these work products, many errors may get generated.
- Project manager has to identify all these errors to bring quality software
  - Error tracking is a process of assessing the status of the software project.
  - Software team performs FTRs to test the software developed.
    - In this review, various errors are identified and corrected.
    - Any errors that remain uncovered and are found later tasks are called defects.

Quality Metrics ….Contd.

- The defect removal efficiency (DRE) can be defined as:

  DRE = E/ (E + D)

  where, DRE is the defect removal efficiency

  E is the error

  D is defect

  - DRE represents the effectiveness of quality assurance activities.
  - It helps the project manager to assess the progress of software project as it gets developed through its scheduled work task.

- Error tracking metrics can also be used for better target review and testing resources.

Measuring Software Quality

❖ Following are the measures of the software quality:

1)Code Quality

- Code quality metrics measure the quality of code used for the software project development.
- Maintaining the software code quality by writing Bug-free and semantically correct code is very important for good software project development.
- In code quality both Quantitative metrics like the number of lines, complexity, functions, rate of bugs generation, etc. and Qualitative metrics like readability, code clarity, efficiency, maintainability, etc. are measured.

……contd.

Measuring Software Quality ….

2) Reliability
- Reliability metrics express the reliability of software in different conditions.
- The software is able to provide exact service at the right time or not is checked.
- Reliability can be checked using Mean Time Between Failure (MTBF) and Mean Time To Repair (MTTR).

3) Performance
- Performance metrics are used to measure the performance of the software.
- Each software has been developed for some specific purposes.
- Performance metrics measure the performance of the software by determining whether the software is fulfilling the user requirements or not, by analyzing how much time and resource it is utilizing for providing the service.

……..contd.

Measuring Software Quality ….

4) Correctness

- Correctness is one of the important software quality metrics as this checks whether the system or software is working correctly without any error by satisfying the user.
- Correctness gives the degree of service each function provides as per developed.

5) Integrity

- Software integrity is important in terms of how much it is easy to integrate with other required software's which increases software functionality and what is the control on integration from unauthorized software which increases the chances of cyberattacks.

……contd.

Measuring Software Quality ….

6) Usability
   - Usability metrics check whether the program is user-friendly or not.
   - Each software is used by the end-user.
   - So it is important to measure that the end-user is happy or not by using this software.

7) Maintainability
   - Each software product requires maintenance and up-gradation.
   - Maintenance is an expensive and time-consuming process.
   - So, if the software product provides easy maintainability then we can say software quality is up to mark.

……..contd.

Measuring Software Quality ….

- Maintainability metrics include time requires to adapt to new features/functionality, Mean Time to Change (MTTC), performance in changing environments, etc.

9) Security

- Security metrics measure how much secure the software is?
- In the age of cyber terrorism, security is the most essential part of every software.
- Security assures that there are no unauthorized changes, no fear of cyber attacks, etc. when the software product is in use by the end-user.

Software Safety

- Software safety is a quality assurance activity in which potential hazards are identified and assessed.
  - These hazards may bring total failure of the system.
  - If such hazards are identified and specified in early stage of software development, then such hazards can be eliminated or controlled in order to make the software safe.
  - Modelling and analysis process is conducted as a part of software safety.
  - Ex.
    - In a computer based automobile system, software hazards are –
      - Uncontrolled acceleration that cannot be stopped
      - Does not respond to depression of brake pedal (by turning off)
      - Does not engage when switch is activated
      - Slowly loses or gains speed

Software Safety ….Contd.

❖ Steps to handle system level hazards are:

1. Identify the hazards
2. Use analysis techniques to assign severity of the hazards
   - The probability of occurrence of such hazards is also analyzed with the help of analysis techniques.
   - Commonly used techniques are fault-tree analysis, real-time logic and Petri-net models.
   - These techniques basically predict the chain of events that can cause hazards.
3. Once hazards are identified, safety related requirements can be specified for the software.
   - This specification basically includes list of undesirable events and desired system responses.
4. Finally, the role of software in managing undesirable event is then indicated.

Software Quality Assurance (SQA) Plan

- SQA Plan provides a road map for instituting software quality assurance.
  - It is developed by SQA group (or the software team, if SQA group does not exist)
  - SQA plan serves as a template for SQA activities that are instituted for each software project.
  - A standard for SQA plan has been published by IEEE
    - The standard recommends a structure that identifies –
      i. The purpose and scope of the plan
      ii. A description of all software engineering work products (Ex. Models, documents, source code, etc.) that fall within the purview of SQA.
      iii. All applicable standards and practices that are applied during the software process

…….contd.

Software Quality Assurance (SQA) Plan

iv. SQA actions and tasks (including reviews and audits) and their placement throughout the software process

v. The tools and the methods that support SQA actions and tasks

vi. Software configuration management procedures for managing change

vii. Methods for assembling, safeguarding and maintaining all SQA - related records

viii. Organizational roles and responsibilities relative to product quality

Software Configuration Management (SCM)

- The output of the software process is information that may be divided into three broad categories:
  - Computer programs (both source level and executable forms)
  - Work products that describe the computer programs (targeted at both technical practitioners and users)
  - Data (contained with the program or external to it)

- When we develop software, the product (software) undergoes many changes in its maintenance phase.
  - We need to handle these changes effectively.
- Several individuals (programs) works together to achieve these common goals.
  - This individual produces several work product (SC Items) e.g., Intermediate version of modules or test data used during debugging, parts of the final product.

SCM …..Contd.

- The elements that comprise all information produced as a part of the software process are collectively called a software configuration.
- As software development progresses, the number of Software Configuration elements (SCI's) grow rapidly.
- 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.

- In Software Engineering, Software Configuration Management(SCM) is a process to systematically manage, organize, and control the changes in the documents, codes, and other entities during the Software Development Life Cycle.
  - The primary goal is to increase productivity with minimal mistakes.
  - SCM is part of cross-disciplinary field of configuration management and it can accurately determine who made which revision.

SCM …..Contd.

- The fundamental sources of change are:-
  1. New business or market conditions dictate changes in product requirements or business rules.
  2. New customer needs demand modifications of data produced by information systems, functionality delivered by products, or services delivered by a computer-based system.
  3. Re-organization or business growth/ downsizing causes change in project priorities or software engineering team structure.
  4. Budgetary or scheduling constraints cause a re-definition of the system or product.
  5. Providing different functionalities or change in user requirements.

SCM …..Contd.

- Software Configuration Management (SCM) is a set of activities carried out for identifying, organizing and controlling changes throughout the life cycle of computer system.
- During the development of software, a change must be managed and controlled in order to improve the quality and reduce error.
- Hence, SCM is a quality assurance activity that is applied throughout the software process.
- It is a set of tracking and controlling activities that begin when a software development project begins and terminates when the software is taken out of operation.

Need for SCM

- SCM is concerned with managing the changes in the evolving software.
  - If the changes are not controlled at all, then this stream of uncontrolled change can cause a well-running software project into chaos.
- Hence, it is essential to perform the following activities –
  i. Identify the changes
  ii. Control the changes
  iii. Ensure that the changes are properly implemented and
  iv. Report these changes to others

- SCM may be seen as part of quality management process.

Need for SCM ……

- The primary reasons for implementing SCM System are:
  - There are multiple people working on software which is continually updating.
  - It may be a case where multiple version, branches, authors are involved in a software config project, and the team is geographically distributed and works concurrently.
  - Changes in user requirement, policy, budget, schedule need to be accommodated.
  - Software should be able to run on various machines and Operating Systems.
  - Helps to develop coordination among stakeholders.
  - SCM process is also beneficial to control the costs involved in making changes to a system.

- Any change in the software configuration Items will affect the final product. Therefore, changes to configuration items need to be controlled and managed.

Tasks in SCM Process

❖ Following are the tasks in SCM process:

I. Configuration Identification
II. Baselines
III. Change Control
IV. Configuration Status Accounting
V. Configuration Audits and Reviews

I. Configuration Identification:

- Configuration identification is a method of determining the scope of the software system.
- With the help of this step, you can manage or control something even if you do not know what it is.
- It is a description that contains the CSCI type (Computer Software Configuration Item), a project identifier and version information.

…….contd.

SCM Process - Configuration Identification

- Activities during this process:
  - Identification of configuration Items like source code modules, test case, and requirements specification.
  - Identification of each CSCI in the SCM repository, by using an object-oriented approach.
  - The process starts with basic objects which are grouped into aggregate objects.
    - Details of what, why, when and by whom changes in the test are made
  - Every object has its own features that identify its name that is explicit to all other objects
  - List of resources required such as the document, the file, tools, etc.

…….contd.

SCM Process - Configuration Identification

- Example:
  - ✔ Instead of naming a file login.php, it should be named login_v1.2.php, where v1.2 stands for the version number of the file.
  - ✔ Instead of naming folder “Code”, it should be named “Code_D” where D represents code should be backed up daily.

II. Baseline

- A baseline is a formally accepted version of a software configuration item.
- It is designated and fixed at a specific time while conducting the SCM process.
- It can only be changed through formal change control procedures.

…….contd.

SCM Process - Baseline

- Activities during this process:
  - Facilitate construction of various versions of an application
  - Defining and determining mechanisms for managing various versions of these work products
  - The functional baseline corresponds to the reviewed system requirements
  - Widely used baselines include functional, developmental, and product baselines
- In simple words, baseline means ready for release.

III. Change Control

- Change control is a procedural method which ensures quality and consistency when changes are made in the configuration object.

SCM Process – Change Control

- In this step, the change request is submitted to software configuration manager.

- Activities during this process:
  - Control ad-hoc change to build stable software development environment.
    - Changes are committed to the repository
  - The request will be checked based on the technical merit, possible side effects and overall impact on other configuration objects.
  - It manages changes and making configuration items available during the software lifecycle

SCM Process – Configuration Status Accounting

IV. Configuration Status Accounting:

A. Configuration status accounting tracks each release during the SCM process.

B. This stage involves tracking what each version has and the changes that lead to this version.

C. Activities during this process**:**

1. Keeps a record of all the changes made to the previous baseline to reach a new baseline
2. Identify all items to define the software configuration
3. Monitor status of change requests
4. Complete listing of all changes since the last baseline
5. Allows tracking of progress to next baseline
6. Allows to check previous releases/versions to be extracted for testing

SCM Process – Configuration Audits and Reviews:

V. Configuration Audits and Reviews:

- Software Configuration audits verify that all the software product satisfies the baseline needs.
- It ensures that what is built is what is delivered.
- Activities during this process:
  - Configuration auditing is conducted by auditors by checking that defined processes are being followed and ensuring that the SCM goals are satisfied.
  - To verify compliance with configuration control standards. auditing and reporting the changes made
  - SCM audits also ensure that traceability is maintained during the process.
  - Ensures that changes made to a baseline comply with the configuration status reports
  - Validation of completeness and consistency

Software Configuration Items (SCI)

- A software configuration item is information that is created as part of the software engineering process.
- Examples of Software Configuration Items are:
  - Computer Programs
    - Source programs
    - Executable programs
  - Documents describing the programs
    - Technical manual
    - Users manual
  - Data
    - Program components or functions
    - External data
    - File structure

…….contd.

Software Configuration Items (SCI)

- For each type of item, there may be a large number of different individual items produced.
- Ex.
  - ✔ There may be many documents for a software specification such as project plan, quality plan, design documents, programs, test, reports, review reports, etc.
- These SCI or items will be produced during the project, then stored, retrieved, changed, stored again, and so on.
  - Each configuration item must have a unique name, and a description or specification which distinguishes it from other items of the same types.

SCM Processes

- The primary objective of SCM process are:
  - Configuration Identification
    - Identify the items that define the software configuration
  - Change Control
    - Manage changes to one or more items
  - Version Control
    - Facilitate to create different versions of the application
  - Configuration Authentication
    - To ensure that the quality of the software is maintained as the configuration evolves over the time.

……contd.

SCM Processes ..... Contd.

- SCM process must be developed in such a way that the software team must answer the following questions:
    1. How does the software team identify the software configuration items (SCI)?
    2. How does the software team control the changes in the software before and after delivering it to the customer?
    3. How does the software team manage the versions of the programs in the software package?
    4. How does the team get ensured that the changes are made properly?
    5. Who is responsible for approving the changes in the software?

……contd.

SCM Processes …..Contd.

- The answers lead to the definition of five tasks of SCM –
  1. Identification
  2. Change control
  3. Version control
  4. Configuration audit
  5. Status reporting

Identification of Objects in Software Configuration

- SCIs must be separately named and identified as objects.
  - These objects must be arranged using object oriented approach.
  - Objects are of two categories:

I. Basic Objects
   - Basic object is the unit of information created during requirement analysis, design, coding and testing.
   - Ex.
     - ✔ Basic object can be a part of the source code

II. Aggregate Objects
   - Aggregate object is a collection of basic/essential objects and other aggregate objects.
   - Each object has a set of distinct characteristics that identify it uniquely: a name, a description, a list of resources, and a "realization".

……contd.

Identification of Objects in Software Configuration

- Ex.
  - ✔ Design Specification is an aggregate object.
    - ✔ SRS or data model
  - ❑ Name
    - Name of the object is a collection of characters, string or some text.
    - It is unique
  - ❑ Description
    - Object description contains document, program or some other description such as project identifier or version information
  - ❑ List of resources
    - Resources are the entities that are used for accessing, referencing and processing of objects.

……. contd.

Identification of Objects in Software Configuration

- Data types and functionalities can serve as a resource.

❑ Realization or identification

- It is a pointer to the object.

❖ Configuration object identification can also consider relationships that exist between the named objects.

❖ If a change is made to one configuration item, it is possible to determine which other configurations objects in the repository can be affected by the change.

❖ As object evolves throughout the software process, its evolution along with the process must be identified.

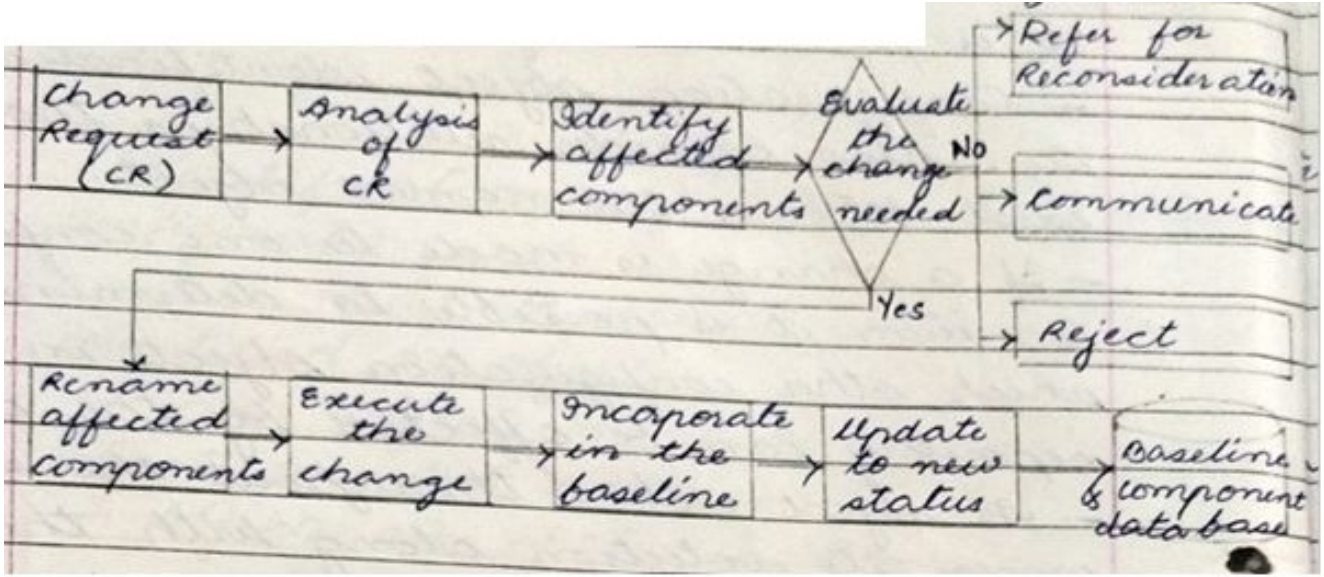
❖ Major modifications in the object must be noted.

Change Control Management

Changes in any software project are vital.

- Introducing small changes in the system may lead to big problems in the product.
- Changes may enhance the capabilities of the system.
- Uncontrolled change creates a lot of chaos.
- For maintaining such changes, human procedures or automated tools can be used.
- Request for change may arise due to the following reasons:-
    - Bugs detected while in use.
    - Improving the process.
    - Adding functions/ features for effectiveness and efficiency.
    - Out of a business need, for moving to a new platform.
    - Expanding the scope of the system to another domain.

Change Control Process



Change Control Process …..Contd.

- Change control is manual step in software lifecycle. It combines human procedures and automated tools.
- Change control process is illustrated in the figure in the next slide.
- The process is as follows –
  - Change request submitted and evaluated to assess technical merit, potential side effects, overall impact on other configuration object and system function, and project cost of change.
  - The result of the evaluation are presented as a change report, which is used by the change control authority(CCA) – A person or group who make final decision on the status and priority of the change.
  - An engineering change order (ECO) is generated for each approved change.

…….contd.

Change Control Process …..Contd.

- The ECO describes the change order to be made, the constraints that must be respected, and the criteria for view and audit.

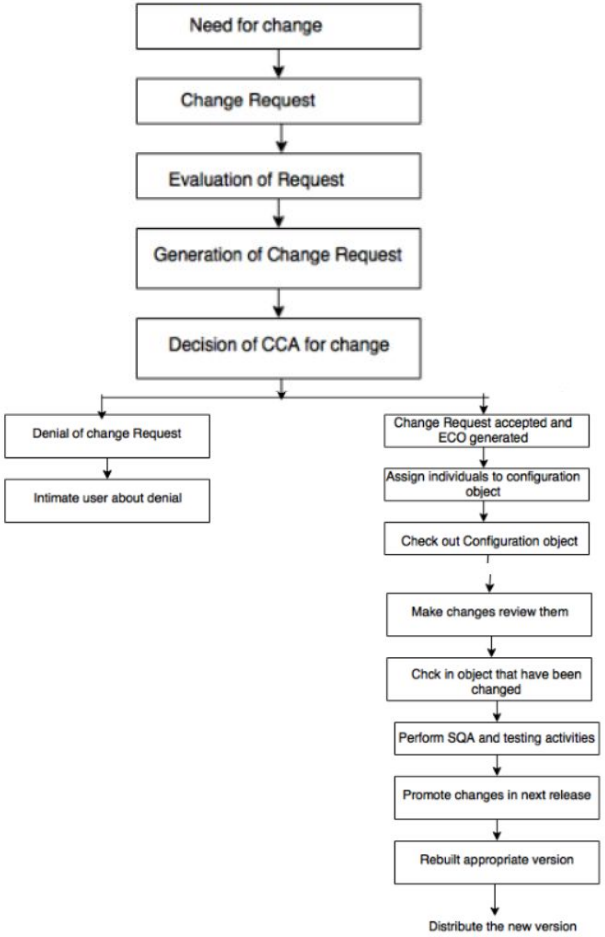
- The object to be changed can be placed in a directory that is controlled by software engineer making the change.
  - As an alternative, the object to be changed can be "checked out" of the project database, change is made, and appropriate SQA activities are applied.
- The object are then "checked in" to the database and appropriate version control mechanism are used to create the next version of the software.

……..contd.

Change Control Process …..Contd.

- Checked in and Checked out mechanism require two important elements:
  1) Access Control
     - The Access control mechanism gives the authority to the software engineer to access and modify the specific configuration object.
  2) Synchronization Control
     - The Synchronization control mechanism allows to make parallel changes or the change made by two different people without overwriting each other's work.

Change Control Process …..Contd.



Version Control

Version is an instance of a system which is functionally distinct in some way from the other system instances.

- It helps to manage different versions of configuration items during the development process.
- Configuration management allows a user to specify the alternative configurations of the software system by selecting appropriate version.
- Certain attributes are associated with each software version. These attributes are useful in identifying the version.
  - Ex. The attribute can be date, creator, customer, status, etc.
- Version needs an associated name for easy reference.
- Each version of software system is a collection of software configuration items.

Version Control ……

- Version Control combines procedures and tools to manage different version of configuration objects that are created during the software process.
- A version control system implements or is directly integrated with four major capabilities:
    a) A project database that stores all relevant configuration objects,
    b) A version management capability that stores all version of configuration object,
    c) A make facility that enables the software engineer to collect all relevant configuration objects, and
    d) Construct a specific version of the software.

…….contd.

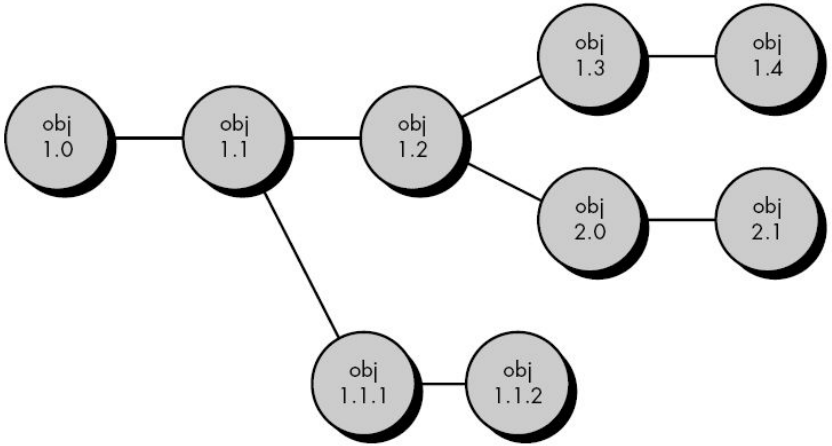
Version Control ……

- A number of version control systems establish a set – a collection of all changes (to some baseline configuration) that are required to create a specific version of the software.
- “Changes set” captures all changes to all files in the configuration along with reason for changes and details of who made the changes and when.
- A number of named change set can be identified for an application or system.
  - This enables a software engineer to construct a version of the software by specifying the changes set (by name) that must be applied to the baseline configuration.
- To accomplish this, a system modelling approach is applied.

…….contd.

Version Control ……

- The system model contains -
  - i. A template that include a component hierarchy and build order for the component that describe how the system must be constructed,
  - ii. Construction rules, and
  - iii. Verification rules.



Configuration Audit

- To ensure that the change has been properly implemented or not, two activities are carried out –
  - A. Formal Technical Reviews (FTR)
    - In this, the correctness of the configuration object is identified and corrected.
    - It is conducted by technical reviewer.
  - B. Software Configuration Audit (SCA)
    - SCA assess the configuration object for the characteristics that are not reviewed in formal technical review (FTR).
    - It is conducted by software quality assurance group.

……contd.

Configuration Audit …..

- Following are some of the primary questions asked during configuration audit –
    1. Whether FTR is conducted to assess the technical correctness?
    2. Whether or not the change specified by ECO has been made?
    3. If additional changes need to be made or not?
    4. Whether the software engineering standards are properly followed?
    5. Do the attributes of configuration object reflect the change?
    6. Whether all the SCI are updated properly?
    7. Whether SCM process are properly followed?

Status Reporting

- Status reporting is also called as Status accounting.
  - It focuses on communication of changes to all the people in the organization that involve with changes.
  - Following questions are asked during status reporting –
    - What happened?
      - What are the changes that are required?
    - Who did it?
      - Who will be handling these changes?
    - When did it happen?
      - The time at which these changes have arised?
    - What else will be affected?
      - The objects or part of the software that might be reflected due to these changes.