ICT-5405

PROJECT MANAGEMENT AND QUALITY ASSURANCE

08 Risk Management

Six-Sigma for Software Engineering

- The term “six sigma” is derived from six standard deviations—3.4 instances (defects) per million occurrences—implying an extremely high quality standard.
- The Six Sigma methodology defines three core steps:
  1. *Define* customer requirements and deliverables and project goals via well-defined methods of customer communication
  2. *Measure* the existing process and its output to determine current quality performance (collect defect metrics)
  3. *Analyze* defect metrics and determine the vital few causes.
  4. *Improve* the process by eliminating the root causes of defects.
  5. *Control* the process to ensure that future work does not reintroduce the causes of defects.

Risk analysis and management

- Risk analysis and management are actions that help a software team to understand and manage uncertainty. Many problems can plague a software project.
- A risk is a potential problem—it might happen, it might not. But, regardless of the outcome, it’s a really good idea to identify it, assess its probability of occurrence, estimate its impact, and establish a contingency plan should the problem actually occur.

Reactive Risk Management

- Project team reacts to risks when they occur
- mitigation—plan for additional resources in anticipation of fire fighting
- fix on failure—resource are found and applied when the risk strikes
- When this fails, “crisis management” takes over and the project is in real jeopardy.

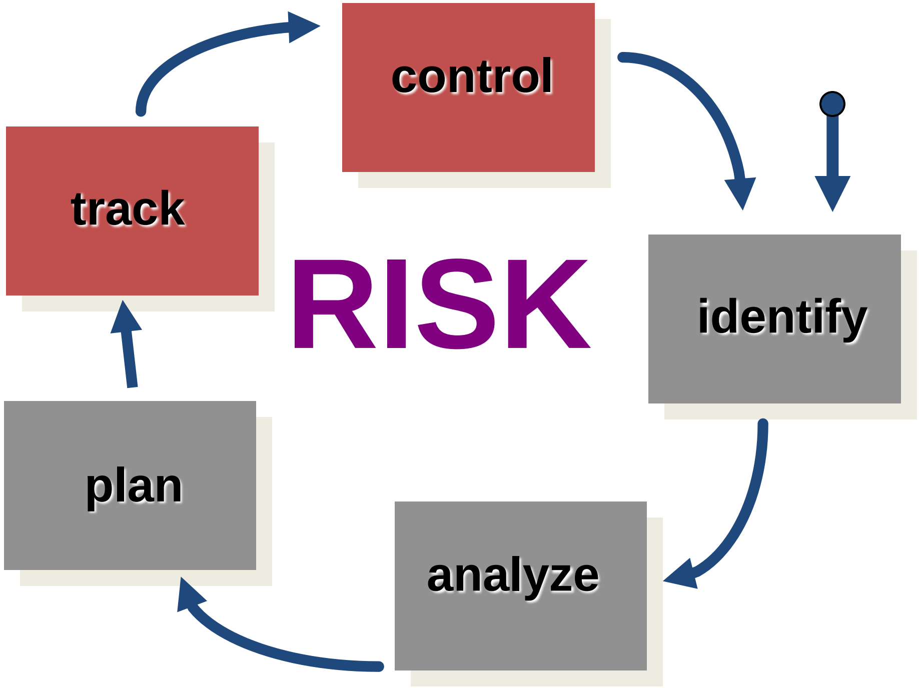
Proactive Risk Management

- A proactive strategy begins long before technical work is initiated.
- Potential risks are identified, their probability and impact are assessed, and they are ranked by importance. Then, the software team establishes a plan for managing risk .
- The primary objective is to **avoid risk**, but because not all risks can be avoided, the team works to develop a contingency plan that will enable it to respond in a controlled and effective manner.

Seven Principles of Risk Management

- **Maintain a global perspective**—view software risks within the context of system and the business problem
- **Take a forward-looking view**—think about the risks that may arise in the future; establish contingency plans
- **Encourage open communication**—if someone states a potential risk, don’t discount it.
- **Integrate**—a consideration of risk must be integrated into the software process
- **Emphasize a continuous process**—the team must be vigilant throughout the software process, modifying identified risks as more information is known and adding new ones as better insight is achieved.
- **Develop a shared product vision**—if all stakeholders share the same vision of the software, it likely that better risk identification and assessment will occur.
- **Encourage teamwork**—the talents, skills and knowledge of all stakeholder should be pooled

Risk Management Paradigm



Risk Identification

- Product size—risks associated with the overall size of the software to be built or modified.
- Business impact—risks associated with constraints imposed by management or the marketplace.
- Customer characteristics—risks associated with the sophistication of the customer and the developer's ability to communicate with the customer in a timely manner.
- Process definition—risks associated with the degree to which the software process has been defined and is followed by the development organization.
- Development environment—risks associated with the availability and quality of the tools to be used to build the product.
- Technology to be built—risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system.
- Staff size and experience—risks associated with the overall technical and project experience of the software engineers who will do the work.

Assessing Project Risk

The following questions have been derived from risk data obtained by surveying experienced software project managers in different parts of the world . The questions are ordered by their relative importance to the success of a project.

1. Have 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 their customers?
4. Have customers been involved fully in the definition of requirements?
5. Do end-users have realistic expectations?

Assessing Project Risk

6. Is project scope stable?
7. Does the software engineering team have the right mix of skills?
8. Are 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 customer/user constituencies agree on the importance of the project and on the requirements for 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 Components

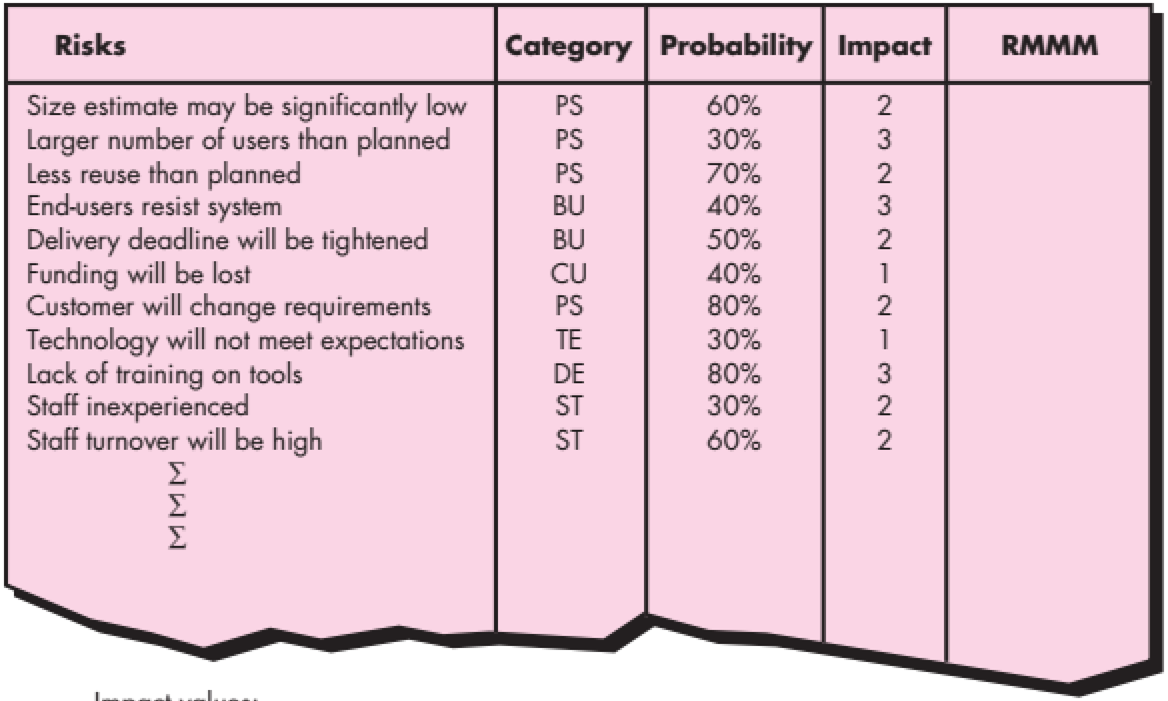
- Performance risk—the degree of uncertainty that the product will meet its requirements and be fit for its intended use.
- cost risk—the degree of uncertainty that the project budget will be maintained.
- support risk—the degree of uncertainty that the resultant software will be easy to correct, adapt, and enhance.
- schedule risk—the degree of uncertainty that the project schedule will be maintained and that the product will be delivered on time.

Risk Projection

- Risk projection, also called risk estimation, attempts to rate each risk in two ways
  1. the likelihood or probability that the risk is real
  2. the consequences of the problems associated with the risk, should it occur.
- The are four risk projection steps:
  - establish a scale that reflects the perceived likelihood of a risk
  - delineate the consequences of the risk
  - estimate the impact of the risk on the project and the product,
  - note the overall accuracy of the risk projection so that there will be no misunderstandings.

Building a Risk Table

- A risk table provides you with a simple technique for risk projection.



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

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

Building the Risk Table

- Estimate the probability of occurrence
- Estimate the impact on the project on a scale of 1 to 5, where

  1 = low impact on project success

  5 = catastrophic impact on project success
- sort the table by probability and impact

Risk Exposure (Impact)

- The overall *risk exposure,* RE, is determined using the following relationship [Hal98]:

$$\mathbf{RE} = \boldsymbol{P} \times \boldsymbol{C}$$

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

Risk Exposure Example

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

Risk Mitigation, Monitoring, and Management

- mitigation—how can we avoid the risk?
- monitoring—what factors can we track that will enable us to determine if the risk is becoming more or less likely?
- management—what contingency plans do we have if the risk becomes a reality?

Risk Due to Product Size

- estimated size of the product in LOC or FP?
- estimated size of product in number of programs, files, transactions?
- percentage deviation in size of product from average for previous products?
- size of database created or used by the product?
- number of users of the product?
- number of projected changes to the requirements
- for the product? before delivery? after delivery?
- amount of reused software?

Risk Due to Business Impact

- affect of this product on company revenue?
- visibility of this product by senior management?
- reasonableness of delivery deadline?
- number of customers who will use this product
- interoperability constraints
- sophistication of end users?
- amount and quality of product documentation that must be produced and delivered to the customer?
- governmental constraints
- costs associated with late delivery?
- costs associated with a defective product?

Risks Due to the Customer

Questions that must be answered:

- Have you worked with the customer in the past?
- Does the customer have a solid idea of requirements?
- Has the customer agreed to spend time with you?
- Is the customer willing to participate in reviews?
- Is the customer technically sophisticated?
- Is the customer willing to let your people do their job—that is, will the customer resist looking over your shoulder during technically detailed work?
- Does the customer understand the software engineering process?

Risks Due to Process Maturity

Questions that must be answered:

- Have you established a common process framework?
- Is it followed by project teams?
- Do you have management support for software engineering
- Do you have a proactive approach to SQA?
- Do you conduct formal technical reviews?
- Are CASE tools used for analysis, design and testing?
- Are the tools integrated with one another?
- Have document formats been established?

Technology Risks

Questions that must be answered:

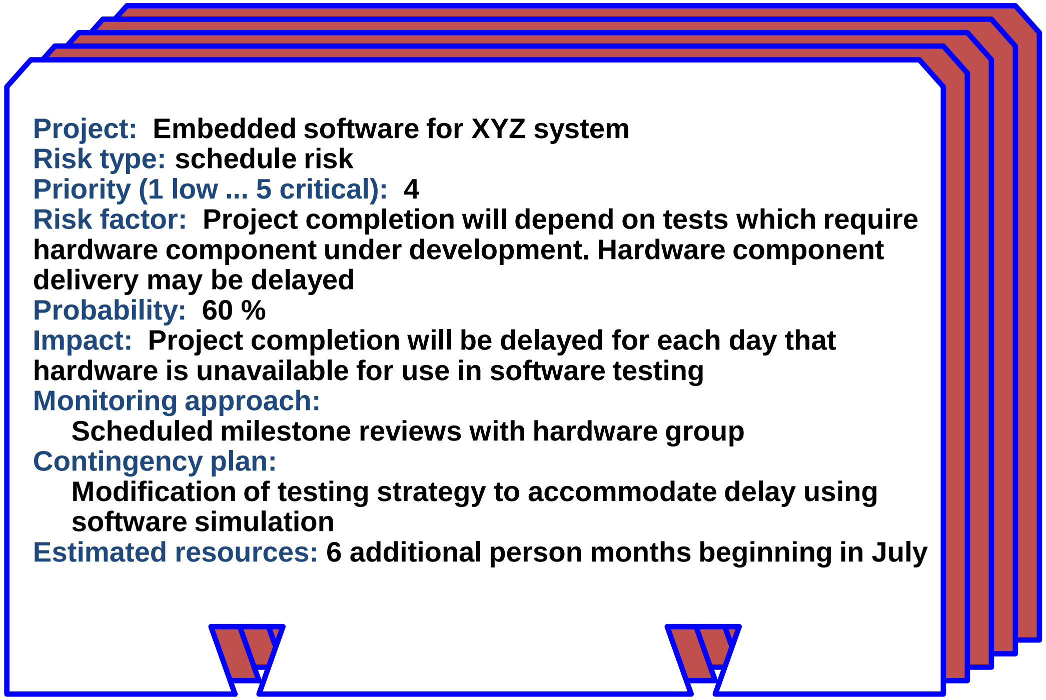
- Is the technology new to your organization?
- Are new algorithms, I/O technology required?
- Is new or unproven hardware involved?
- Does the application interface with new software?
- Is a specialized user interface required?
- Is the application radically different?
- Are you using new software engineering methods?
- Are you using unconventional software development methods, such as formal methods, AI-based approaches, artificial neural networks?
- Are there significant performance constraints?
- Is there doubt the functionality requested is "do-able?"

Staff/People Risks

Questions that must be answered:

- Are the best people available?
- Does staff have the right skills?
- Are enough people available?
- Are staff committed for entire duration?
- Will some people work part time?
- Do staff have the right expectations?
- Have staff received necessary training?
- Will turnover among staff be low?

Recording Risk Information



Project: Embedded software for XYZ system
Risk type: schedule risk
Priority (1 low ... 5 critical): 4
Risk factor: Project completion will depend on tests which require hardware component under development. Hardware component delivery may be delayed
Probability: 60 %
Impact: Project completion will be delayed for each day that hardware is unavailable for use in software testing
Monitoring approach:
Scheduled milestone reviews with hardware group
Contingency plan:
Modification of testing strategy to accommodate delay using software simulation
Estimated resources: 6 additional person months beginning in July