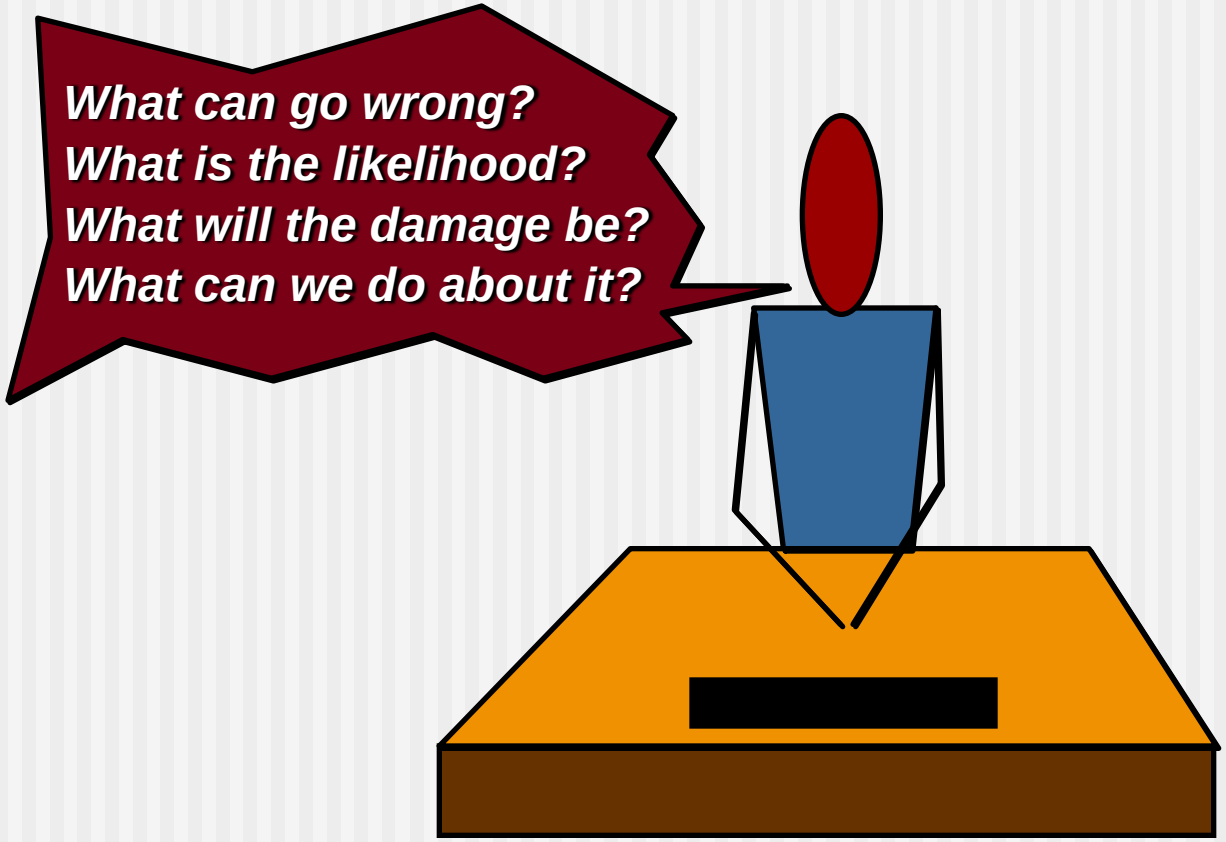
RISK MANAGEMENT

Project Risks



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
- crisis management—failure does not respond to applied resources and project is in jeopardy

Proactive Risk Management

- formal risk analysis is performed
- organization corrects the root causes of risk
  - TQM concepts and statistical SQA
  - examining risk sources that lie beyond the bounds of the software
  - developing the skill to manage change

SOFTWARE RISKS

- Uncertainty
- Loss

Types of risks that we likely encounter as software is built?

- Project Risk
- Technical Risk
- Business Risk
- Known Risk
- Predictable Risk
- Unpredictable Risk

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-I

- Have top software and customer managers formally committed to support the project?
- Are end-users enthusiastically committed to the project and the system/product to be built?
- Are requirements fully understood by the software engineering team and their customers?
- Have customers been involved fully in the definition of requirements?
- Do end-users have realistic expectations?

Assessing Project Risk-II

- Is project scope stable?
- Does the software engineering team have the right mix of skills?
- Are project requirements stable?
- Does the project team have experience with the technology to be implemented?
- Is the number of people on the project team adequate to do the job?
- Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?

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
  - the likelihood or probability that the risk is real
  - 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

Risk	Probability	Impact	RMMM
			Risk Mitigation Monitoring & Management

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]:

RE = *P* x *C*

where

P is the probability of occurrence for a risk, and

C is the cost to the project should the risk occur.

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

Attributes that affect risk:

- **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

Attributes that affect risk:

- **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?**

Software Quality Assurance

.

Comment on Quality

- Phil Crosby once said:
  - The problem of quality management is not what people don't know about it. The problem is what they think they do know . .
  - *Everybody is for it.* (Under certain conditions, of course.)
  - *Everyone feels they understand it.* (Even though they wouldn't want to explain it.)
  - *Everyone thinks execution is only a matter of following natural inclinations.* (After all, we do get along somehow.)
  - *And, of course, most people feel that problems in these areas are caused by other people.* (If only they would take the time to do things right.)

Elements of SQA

- **Standards**
- **Reviews and Audits**
- **Testing**
- **Error/defect collection and analysis**
- **Change management**
- **Education**
- **Vendor management**
- **Security management**
- **Safety**
- **Risk management**

Elements of SQA

- **Standards:**
- The IEEE, ISO, and other standards organizations have produced a broad array of software engineering standards and related documents. Standards may be adopted voluntarily by a software engineering organization or imposed by the customer or other stakeholders. The job of SQA is to ensure that standards that have been adopted are followed and that all work products conform to them.

Elements of SQA

- **Reviews and Audits:** Technical reviews are a quality control activity performed by software engineers for software engineers. Their intent is to uncover errors. Audits are a type of review performed by SQA personnel with the intent of ensuring that quality guidelines are being followed for software engineering work. For example, an audit of the review process might be conducted to ensure that reviews are being performed in a manner that will lead to the highest likelihood of uncovering errors.

Elements of SQA

- **Testing :**
- Software testing is a quality control function that has one primary goal—to find errors. The job of SQA is to ensure that testing is properly planned and efficiently conducted so that it has the highest likelihood of achieving its primary goal.

Elements of SQA

- **Error/defect collection and analysis :**
- The only way to improve is to measure how you’re doing. SQA collects and analyzes error and defect data to better understand how errors are introduced and what software engineering activities are best suited to eliminating them.

Elements of SQA

- **Change management:** Change is one of the most disruptive aspects of any software project. If it is not properly managed, change can lead to confusion, and confusion almost always leads to poor quality. SQA ensures that adequate change management practices have been instituted.

Elements of SQA

- **Education:**
- Every software organization wants to improve its software engineering practices. A key contributor to improvement is education of software engineers, their managers, and other stakeholders. The SQA organization takes the lead in software process improvement and is a key proponent and sponsor of educational programs.

Elements of SQA

- **Vendor management:**
- Three categories of software are acquired from external software vendors—
  - ***shrink-wrapped packages*** (e.g., Microsoft *Office*), a
  - ***tailored shell*** that provides a basic skeletal structure that is custom tailored to the needs of a purchaser, and
  - ***contracted software*** that is custom designed and constructed from specifications provided by the customer organization.
- The job of the SQA organization is to ensure that high-quality software results by suggesting specific quality practices that the vendor should follow (when possible),

Elements of SQA

- **Security management :** With the increase in cyber crime and new government regulations regarding privacy, every software organization should institute policies that protect data at all levels,
  - **establish firewall protection for WebApps, and**
  - **ensure that software has not been tampered with internally.**
- SQA ensures that appropriate process and technology are used to achieve software security.

Elements of SQA

- **Safety:** Because software is almost always a pivotal component of human rated systems (e.g., automotive or aircraft applications), the impact of hidden defects can be catastrophic. SQA may be responsible for assessing the impact of software failure and for initiating those steps required to reduce risk.

Elements of SQA

- **Risk management :** Although the analysis and mitigation of risk is the concern of software engineers, the SQA organization ensures that risk management activities are properly conducted and that risk-related contingency plans have been established.

Role of the SQA Group-I

- **Prepares an SQA plan for a project.**
  - The plan identifies
    - **evaluations** to be performed
    - **audits and reviews** to be performed
    - **standards** that are applicable to 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
- **Participates in the development of the project's software process description.**
  - The SQA group **reviews the process description** for compliance with **organizational policy, internal software standards, externally imposed standards** (e.g., ISO-9001), and other parts of the software project plan.

Role of the SQA Group-II

- **Reviews software engineering activities to verify compliance with the defined software process.**
  - **identifies, documents, and tracks deviations** from the process and **verifies** that corrections have been made.
- **Audits designated software work products to verify compliance with those defined as part of the software process.**
  - reviews selected work products; identifies, documents, and tracks deviations; verifies that corrections have been made
  - **periodically reports the results of its work to the project manager.**
- **Ensures that deviations in software work and work products are documented and handled according to a documented procedure.**
- **Deviations may be encountered** in the project plan, process description, applicable standards, or software engineering work products.
- **Records any noncompliance and reports to senior management.**
  - **Noncompliance items are tracked until they are resolved.**

SQA Goals (see Figure 16.1)

- **Requirements quality.** The correctness, completeness, and consistency of the requirements model will have a strong influence on the quality of all work products that follow.
- **Design quality.** Every element of the design model should be assessed by the software team to ensure that it exhibits high quality and that the design itself conforms to requirements.
- **Code quality.** Source code and related work products (e.g., other descriptive information) must conform to local coding standards and exhibit characteristics that will facilitate maintainability.
- **Quality control effectiveness.** A software team should apply limited resources in a way that has the highest likelihood of achieving a high quality result.

SQA Goals (see Figure 16.1)

FIGURE 16.1 Software quality goals, attributes, and metrics
Source: Adapted from [Hya96].

Goal	Attribute	Metric
Requirement quality	Ambiguity	Number of ambiguous modifiers (e.g., many, large, human-friendly)
	Completeness	Number of TBA, TBD
	Understandability	Number of sections/subsections
	Volatility	Number of changes per requirement
		Time (by activity) when change is requested
	Traceability	Number of requirements not traceable to design/code
	Model clarity	Number of UML models
		Number of descriptive pages per model
		Number of UML errors
Design quality	Architectural integrity	Existence of architectural model
	Component completeness	Number of components that trace to architectural model
		Complexity of procedural design
	Interface complexity	Average number of pick to get to a typical function or content
		Layout appropriateness
	Patterns	Number of patterns used
Code quality	Complexity	Cyclomatic complexity
	Maintainability	Design factors (Chapter 8)
	Understandability	Percent internal comments
		Variable naming conventions
	Reusability	Percent reused components
	Documentation	Readability index
QC effectiveness	Resource allocation	Staff hour percentage per activity
	Completion rate	Actual vs. budgeted completion time
	Review effectiveness	See review metrics (Chapter 14)
	Testing effectiveness	Number of errors found and criticality
		Effort required to correct an error
		Origin of error

ISO 9001:2000 Standard

- ISO 9001:2000 is the quality assurance standard that applies to software engineering.
- The standard contains 20 requirements that must be present for an effective quality assurance system.
- The requirements delineated by ISO 9001:2000 address topics such as
  - management responsibility, quality system, contract review, design control, document and data control, product identification and traceability, process control, inspection and testing, corrective and preventive action, control of quality records, internal quality audits, training, servicing, and statistical techniques.