Software Quality Management TDDE46

Azeem Ahmad

Linköping University, Sweden

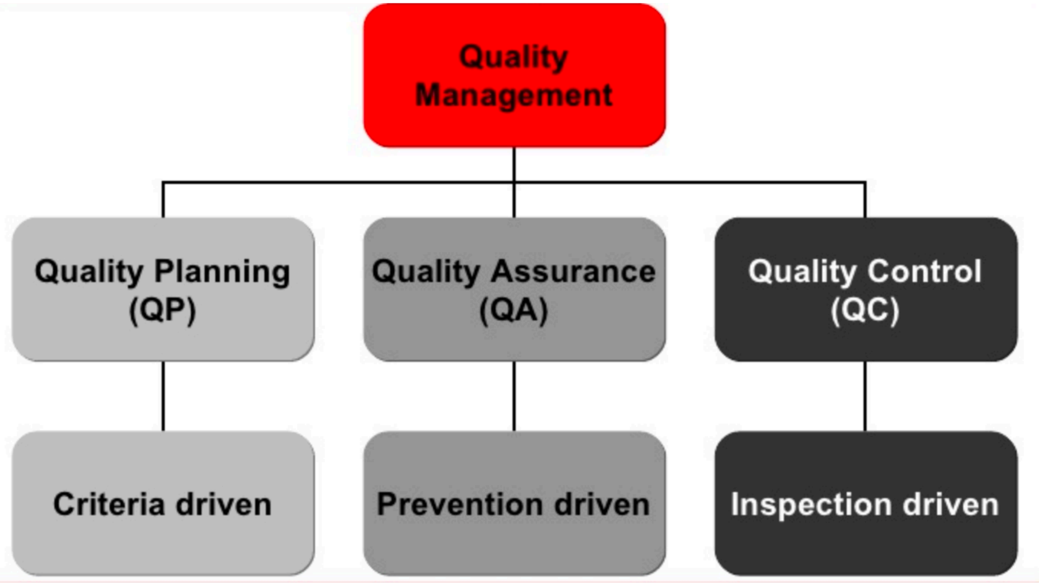
Objectives

- Introduction to Software Quality Management
- Introduction to Total Quality Management
- Understanding Cost of Quality
- Understanding Quality Function Deployment
- Introduction to Risk Management

Software Quality Management (SQM)

- Managing the quality of the software project, process and products
- Involves defining appropriate quality standards and procedures and ensuring that these are followed
- Should aim to develop a ‘quality culture’ where quality is seen as everyone’s responsibility

SQM Activities



- Quality assurance
  - Establish organisational procedures and standards for quality
- Quality planning
  - Select **applicable procedures** and standards for a particular project and modify these as required
- Quality control
  - Ensure that procedures and standards are followed by the software development team

SQM Activities - Quality Assurance

- A planned and systematic way of creating an environment to assure that the software product being developed meets the quality requirements
- QA refers to the implementation of well-defined standard practices and methods. It is a pro-active quality process
- This process is controlled and determined at managerial level
- Examples:
  - process checklists, process standards, process documentation and project audit.

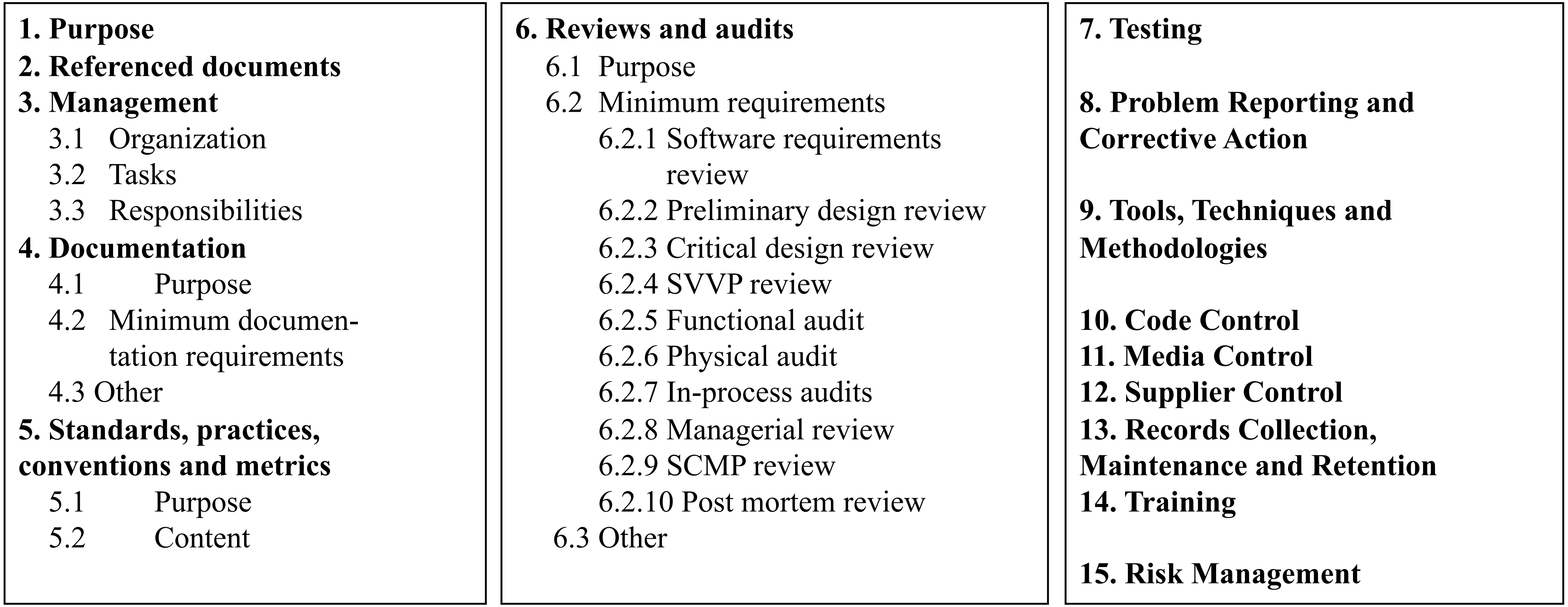
SQM Activities - Quality Planning

- A quality plan sets out the **desired product qualities** and how these are **assessed** and define the **most significant quality attributes**
- It should set out which organisational standards should be applied and, if necessary, define new standards

SQM Activities - Quality Planning Structure

- Product introduction
- Product plans
- Process descriptions
- Quality goals
- Risks and risk management
- Quality plans should be short, succinct documents

SQM Activities - Quality Planning – IEEE 730-1989 Software Quality Assurance Plans



1. Purpose
2. Referenced documents
3. Management
  3.1 Organization
  3.2 Tasks
  3.3 Responsibilities
4. Documentation
  4.1 Purpose
  4.2 Minimum documentation requirements
  4.3 Other
5. Standards, practices, conventions and metrics
  5.1 Purpose
  5.2 Content

6. Reviews and audits
  6.1 Purpose
  6.2 Minimum requirements
    6.2.1 Software requirements review
    6.2.2 Preliminary design review
    6.2.3 Critical design review
    6.2.4 SVVP review
    6.2.5 Functional audit
    6.2.6 Physical audit
    6.2.7 In-process audits
    6.2.8 Managerial review
    6.2.9 SCMP review
    6.2.10 Post mortem review
  6.3 Other

7. Testing

8. Problem Reporting and Corrective Action

9. Tools, Techniques and Methodologies

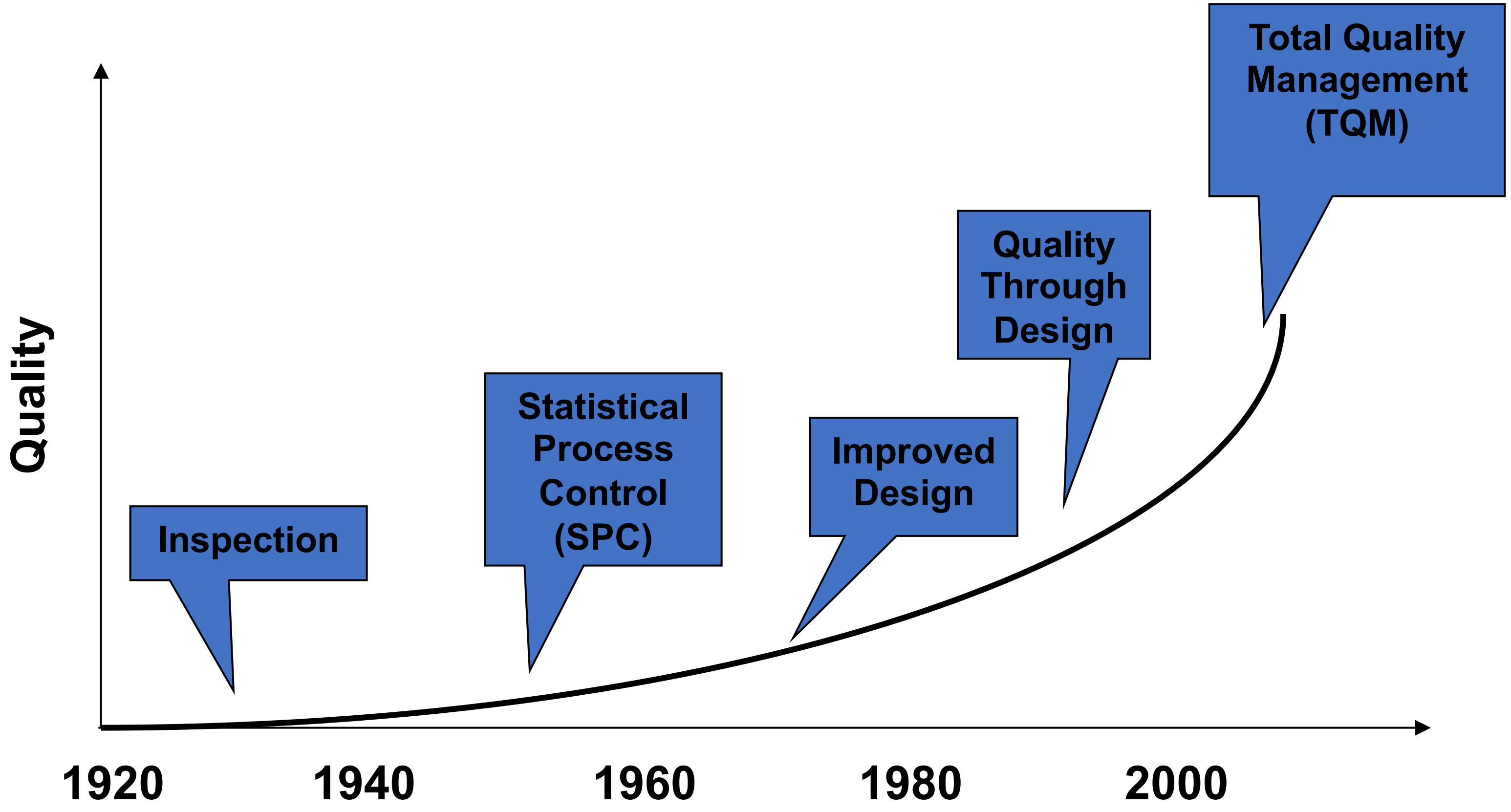
10. Code Control
11. Media Control
12. Supplier Control
13. Records Collection, Maintenance and Retention
14. Training

15. Risk Management

SQM Activities - Quality Control

- Checking the software development process to ensure that procedures and standards are being followed
- Examples:
    - inspection, deliverable peer reviews and the software testing process

How the Definition of Quality Changed Over Time



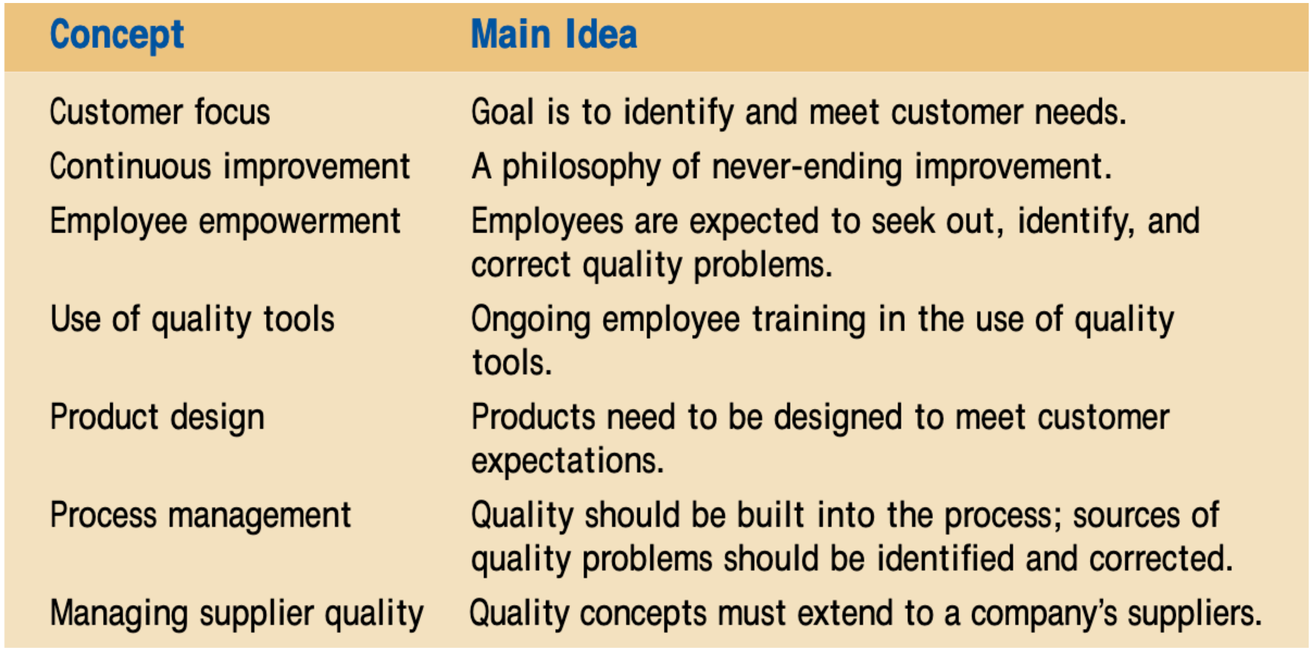
Total Quality Management

- Focus on identifying root causes of quality problems and correcting them at the source, as opposed to inspecting the product after it has been made
- Focuses on building quality into the processas opposed to simply inspecting for poor quality after production
- Specific concepts that make up the philosophy of TQM:
  - Customer Focus
  - Continuous Improvement
  - Employee Empowerment
  - Use of Quality Tools
  - Product Design
  - Process Management
  - Managing Supplier Quality

"Do the right things **right** the first time, every time."

Total Quality Management

- Specific concepts that make up the philosophy of TQM:
  - Customer Focus
  - Continuous Improvement
  - Employee Empowerment
  - Use of Quality Tools
  - Product Design
  - Process Management
  - Managing Supplier Quality



Concept	Main Idea
Customer focus	Goal is to identify and meet customer needs.
Continuous improvement	A philosophy of never-ending improvement.
Employee empowerment	Employees are expected to seek out, identify, and correct quality problems.
Use of quality tools	Ongoing employee training in the use of quality tools.
Product design	Products need to be designed to meet customer expectations.
Process management	Quality should be built into the process; sources of quality problems should be identified and corrected.
Managing supplier quality	Quality concepts must extend to a company's suppliers.

Total Quality Management – Customer Focus

- Company’s focus on its customers
- Quality is defined as meeting or exceeding customer expectations
- The goal is to first identify and then meet customer needs.
- It is a challenge because customer focus changes such as small cars to sports utilities to back to small cars and then electric cars.

Total Quality Management – Continuous Improvement

- Company continually strive to be better through learning and problem solving
- We can never achieve perfection, we must always evaluate our performance and take measures to improve it.
- Some of the approaches:
  - The Plan – Do – Study – Act Cycle
  - Others

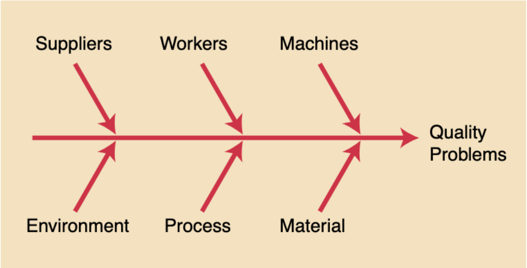
Total Quality Management – Employee Empowerment

- Empower all employees to seek out quality problems and correct them
- Avoid culture of making it “someone else’s problem”
- Incentives for employees to identify quality problems

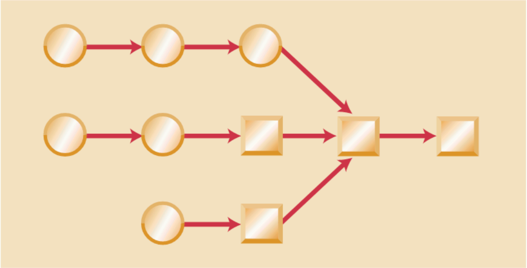
Total Quality Management – Use of Quality Tools

Employees need quality tool to asses quality.

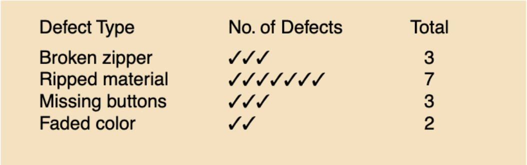
1. Cause-and-Effect Diagram



2. Flowchart

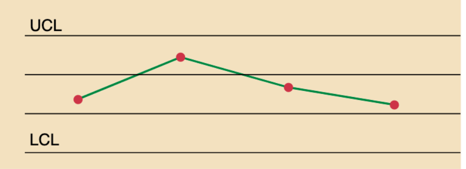


3. Checklist

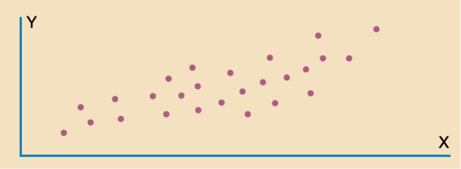


Defect Type	No. of Defects	Total
Broken zipper	✓✓✓	3
Ripped material	✓✓✓✓✓✓✓	7
Missing buttons	✓✓✓	3
Faded color	✓✓	2

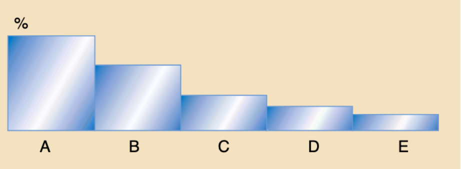
4. Control Chart



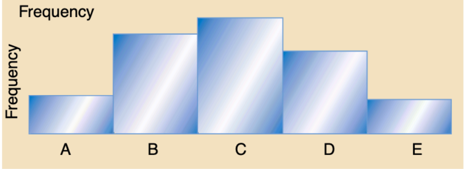
5. Scatter Diagram



6. Pareto Chart



7. Histogram



Total Quality Management – Product Design

- Building quality into a product and to ensure that the product design meets customer expectations
- To produce a product that customers want, we need to translate customers’ everyday language into specific technical requirements
- A useful tool for translating the voice of the customer into specific technical requirements is quality function deployment (QFD) (discussed later in slides)

Total Quality Management – Process Management

- A quality product comes from a quality process
  - *Burned Cookies: Throw away or the temperature setting may be too high; the pan may be curved, placing some cookies closer to the heating element; or the oven may not be distributing heat evenly*
- Old Quality Concept: if a defect is found, correct defect and release the product
- New Quality Concept: If a defect is found, look for root cause.

Total Quality Management – Managing Supplier Quality

- Extends the concept of quality to a company's suppliers
- If suppliers meet preset quality standards, materials do not have to be inspected upon arrival.
  - Many companies have a representative residing at their supplier's location, thereby involving the supplier in every stage from product design to final production

The Cost of Quality

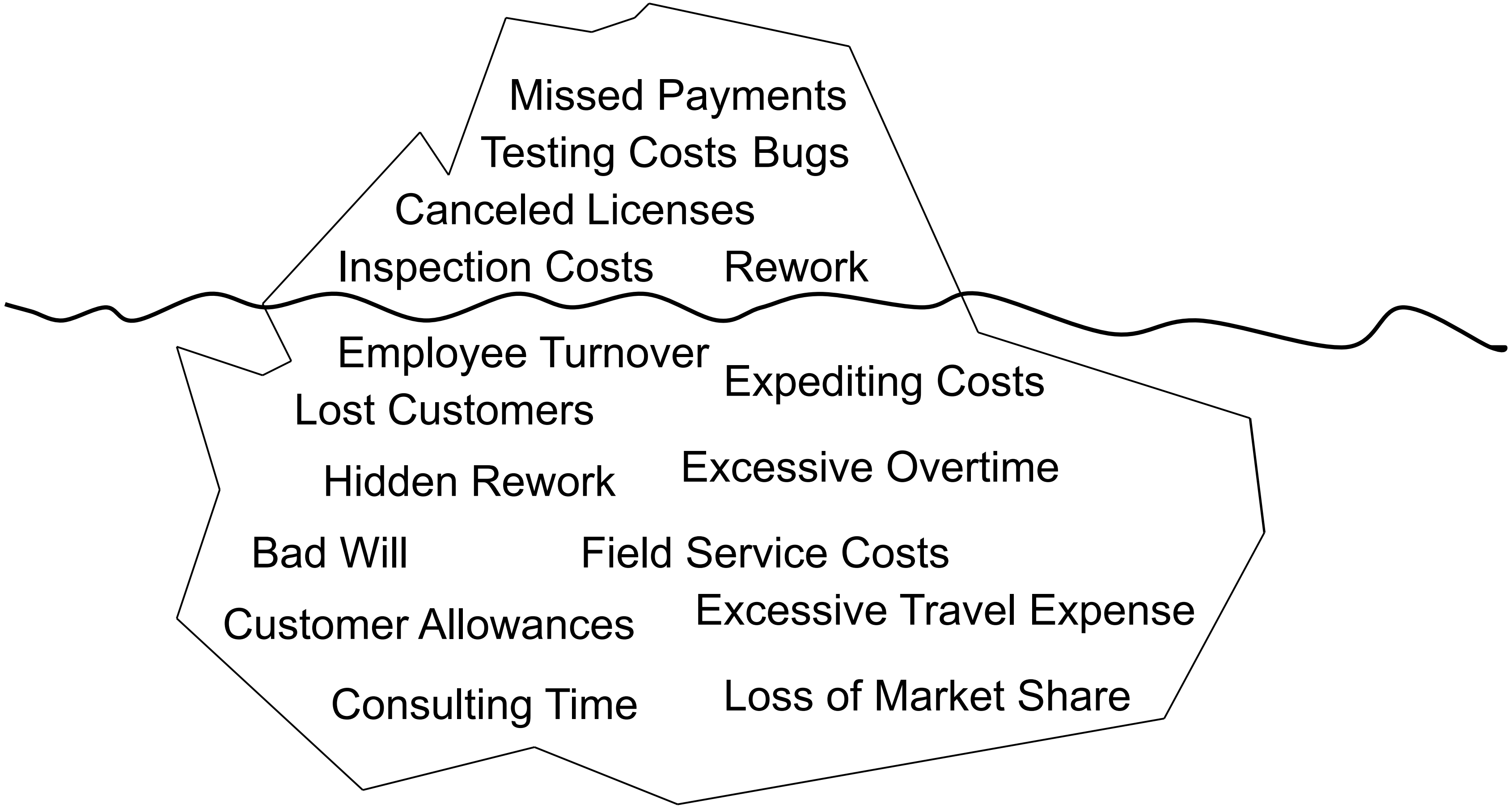
- Quality is important, but it costs us time and money. It costs too much time and money to get the level of software quality we really want.
- There is no doubt that quality has a cost, but lack of quality also has a cost — not only to end users who must live with faulty software, but also to the software organization that has built and must maintain it
- Cost of Quality Include:
  - All costs incurred in the pursuit of quality
  - All cost incurred in performing quality-related activities

The Cost of Quality

	Cost Type	Definition	Example
	Prevention	Costs that are associated with preventing defects.	Training, early reviews, quality planning, tools, process improvement initiatives.
	Appraisal/Asses	Costs that are associated with analyzing and testing the product to ensure it conforms to specifications.	Inspections, testing, audits, quality control.
Failures	Internal Failure	Costs that are associated with fixing defects found prior to release.	Fix the Bug, Repair, retesting, updating documentations.
	External Failure	Costs that are associated with fixing defects found after release.	Technical support, defect reporting and tracking, field updates, loss of future sales.

Joseph M. Juran and F. M. Gryna, *Quality Planning and Analysis* (New York: McGraw-Hill, 1970), p. 60.

The Hidden Cost of Quality



Missed Payments

Testing Costs Bugs

Canceled Licenses

Inspection Costs Rework

Employee Turnover

Expediting Costs

Lost Customers

Hidden Rework

Excessive Overtime

Bad Will

Field Service Costs

Customer Allowances

Excessive Travel Expense

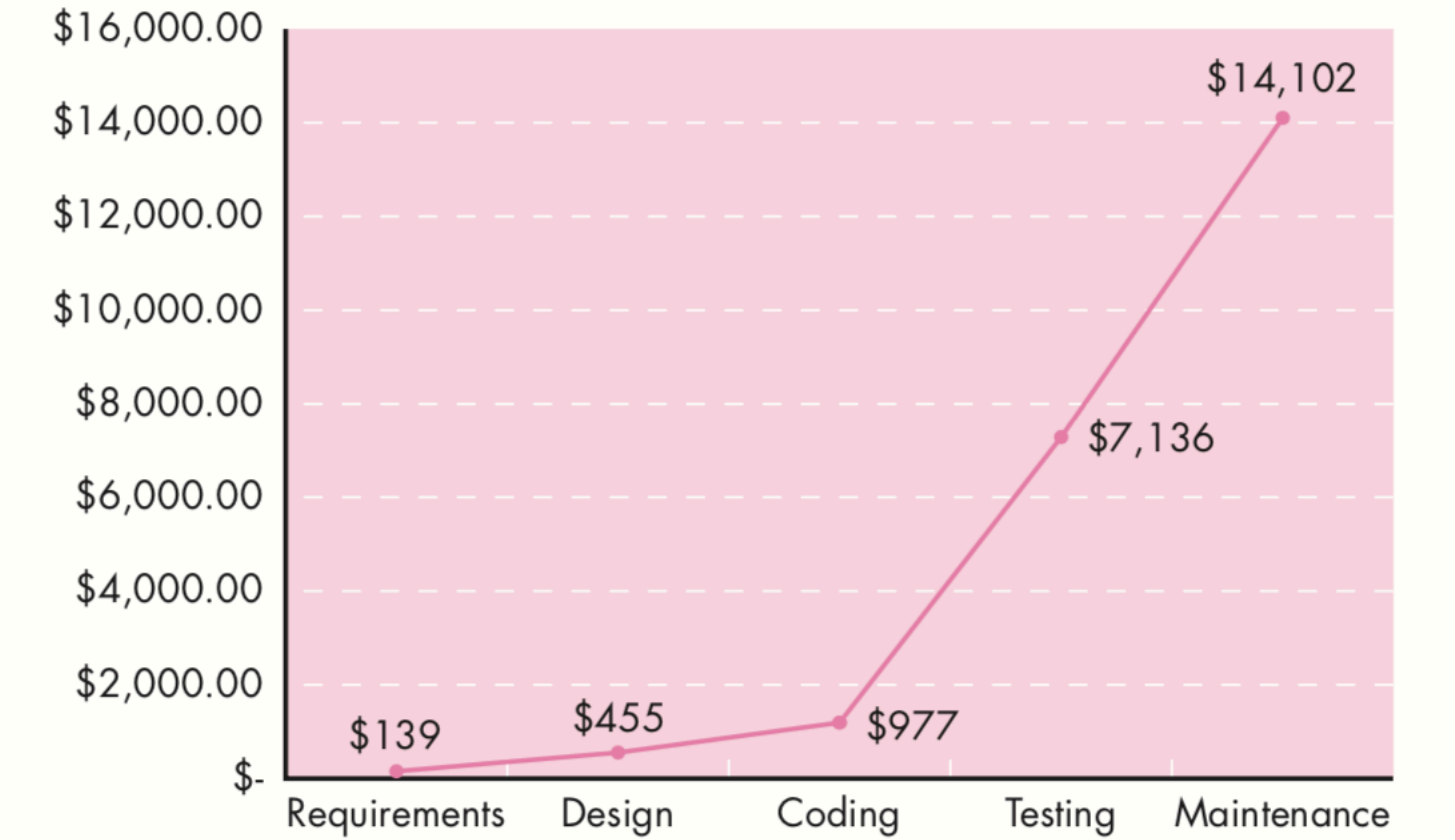
Consulting Time

Loss of Market Share

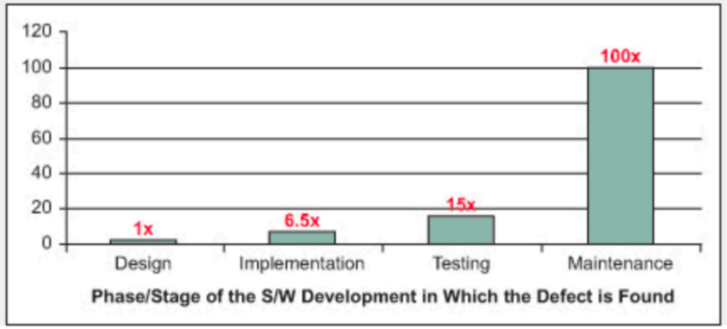
Its Menti Time

- Which phase of SDLC such as requirements, design, coding, testing or maintenance requires larger cost in your opinion?

Go to **www.menti.com** and use the code **25 68 02**



: Relative Costs to Fix Software Defects (Source: IBM Systems Sciences Institute)



According to industry average data, the cost of finding and correcting defects during the coding phase is $977 per defect. Thus, the total cost for correcting the 200 “critical” defects during this phase (200 X $977) is approximately $195,400.

Primary source: Boehm, B., and V. Basili, “Software Defect Reduction Top 10 List,” IEEE Computer, vol. 34, no. 1, January 2001, pp. 135–137 & Picture Copied from Software Engineering by Roger S. Pressmen

Risk of Poor Quality

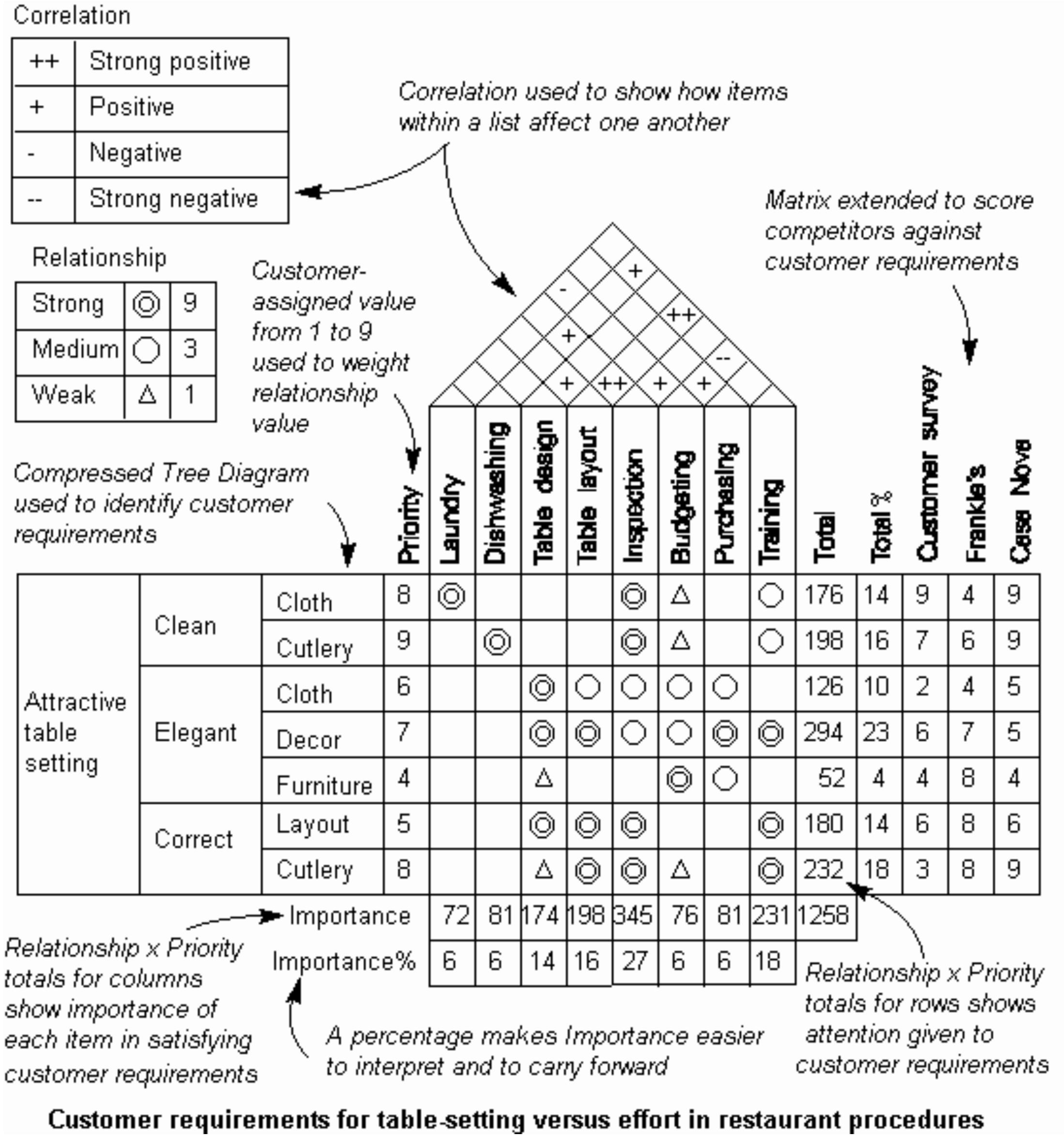
- ”Throughout the month of November 2000 at a hospital in Panama, 28 patients received massive overdoses of gamma rays during treatment for a variety of cancers. In the months that followed, 5 of these patients died from radiation poisoning and 15 others developed serious complications. What caused this tragedy? A software package, developed by a U.S. company, was modified by hospital technicians to compute doses of radiation for each patient”

- The three Panamanian medical physicists, who “tweeked” the software to provide additional capability, were charged with second-degree murder. The U.S. company is faced with serious litigation in two countries

Quality Function Deployment (QFD)

- A useful tool for translating the voice of the customer into specific technical requirements is quality function deployment (QFD)
- Enables us to view the relationships among the variables involved in the design of a product, such as technical versus customer requirements
  - Running tests to see how changes in certain technical requirements of the product affect customer requirements such as an automobile manufacturer evaluating how changes in materials affect customer safety requirements

Primary planning tool used in QFD is the house of quality



Correlation

++	Strong positive
+	Positive
-	Negative
--	Strong negative

Relationship

Strong	◎	9
Medium	○	3
Weak	△	1

Correlation used to show how items within a list affect one another

Customer-assigned value from 1 to 9 used to weight relationship value

Compressed Tree Diagram used to identify customer requirements

Matrix extended to score competitors against customer requirements

			Priority	Laundry	Dishwashing	Table design	Table layout	Inspection	Budgeting	Purchasing	Training	Total	Total %	Customer survey	Frankie's	Case Nova
Attractive table setting	Clean	Cloth	8	◎				◎	△		○	176	14	9	4	9
		Cutlery	9		◎			◎	△		○	198	16	7	6	9
	Elegant	Cloth	6			◎	○	○	○	○		126	10	2	4	5
		Decor	7			◎	◎	○	○	◎	◎	294	23	6	7	5
		Furniture	4			△			◎	○		52	4	4	8	4
	Correct	Layout	5			◎	◎	◎			◎	180	14	6	8	6
		Cutlery	8			△	◎	◎	△		◎	232	18	3	8	9
		Importance		72	81	174	198	345	76	81	231	1258				
		Importance%		6	6	14	16	27	6	6	18					

Relationship x Priority totals for columns show importance of each item in satisfying customer requirements

A percentage makes Importance easier to interpret and to carry forward

Relationship x Priority totals for rows shows attention given to customer requirements

Customer requirements for table-setting versus effort in restaurant procedures

QFD

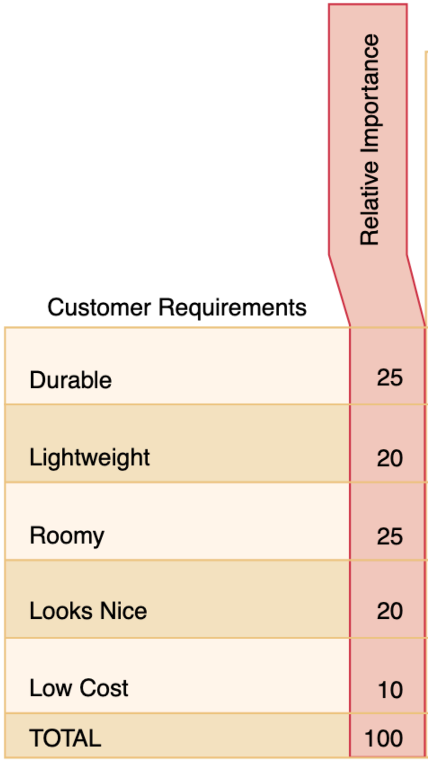
Step 1: Identifying important customer requirements, which typically come from the marketing department

Step 2: These requirements are numerically scored based on their importance, and scores are translated into specific product characteristics

Step 3: Evaluations are then made of how the product compares with its main competitors relative to the identified characteristics

Step 4: Specific goals are set to address the identified problems

The resulting matrix looks like a picture of a houseand is often called the house of quality



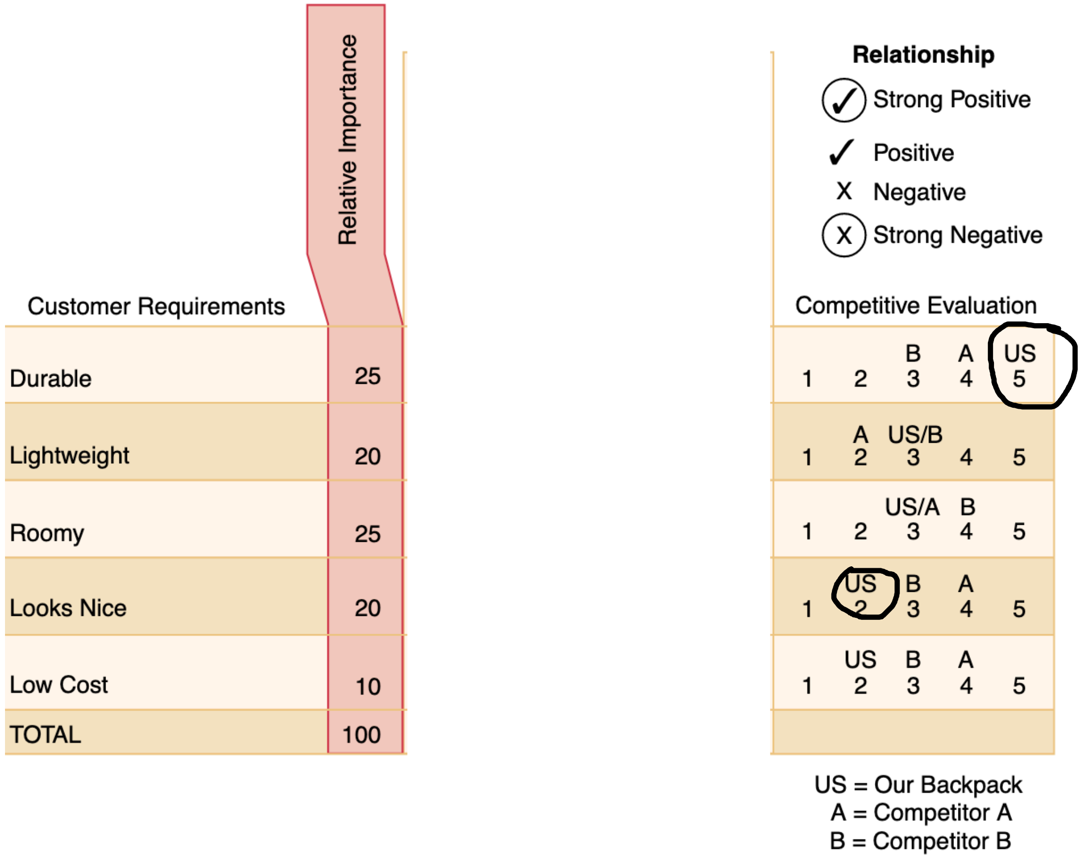
Customer Requirements	Relative Importance
Durable	25
Lightweight	20
Roomy	25
Looks Nice	20
Low Cost	10
TOTAL	100

QFD - Customer Requirements

- Conduct survey, phone calls, interview, focus study to find what customers wants.
- The importance customers attach to each of these requirements should also determined

QFD - Competitive Evaluation

- On the far right of our relationship matrix is an evaluation of how our product compares to those of competitors
- The important thing here is to identify which customer requirements we should pursue and how we fare relative to our competitors.
  - For example, you can see that our product excels in durability relative to competitors, yet it does not look as nice. This means that in designing ourproduct, we could gain a competitive advantage by focusing our design effortson a more appealing product



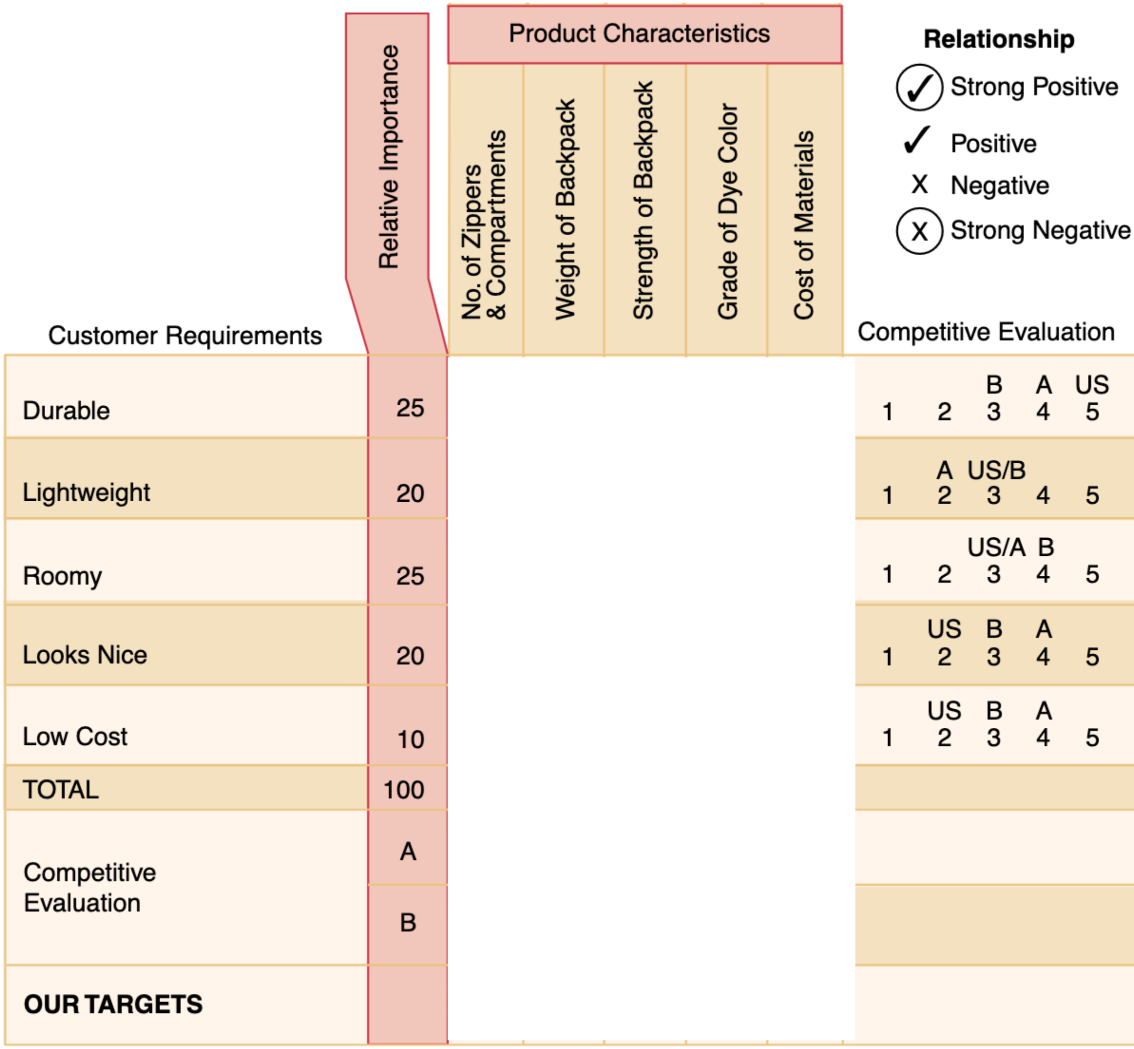
Relationship

Customer Requirements	Relative Importance	Competitive Evaluation 1	2	3	4	5
Durable	25			B	A	US
Lightweight	20		A	US/B		
Roomy	25			US/A	B	
Looks Nice	20		US	B	A	
Low Cost	10		US	B	A	
TOTAL	100					

US = Our Backpack
A = Competitor A
B = Competitor B

QFD - Product Characteristics

- Specific product characteristics are on top of the relationship matrix
- These are technical measures
    - In our example these include the number of zippers and compartments, the weight of the backpack, strength of the backpack, grade of the dye color, and the cost of materials



Customer Requirements	Relative Importance	No. of Zippers & Compartments	Weight of Backpack	Strength of Backpack	Grade of Dye Color	Cost of Materials	Competitive Evaluation				
		Product Characteristics									
Durable	25						1	2	B 3	A 4	US 5
Lightweight	20						1	A 2	US/B 3	4	5
Roomy	25						1	2	US/A 3	B 4	5
Looks Nice	20						1	US 2	B 3	A 4	5
Low Cost	10						1	US 2	B 3	A 4	5
TOTAL	100										
Competitive Evaluation	A										
	B										
OUR TARGETS											

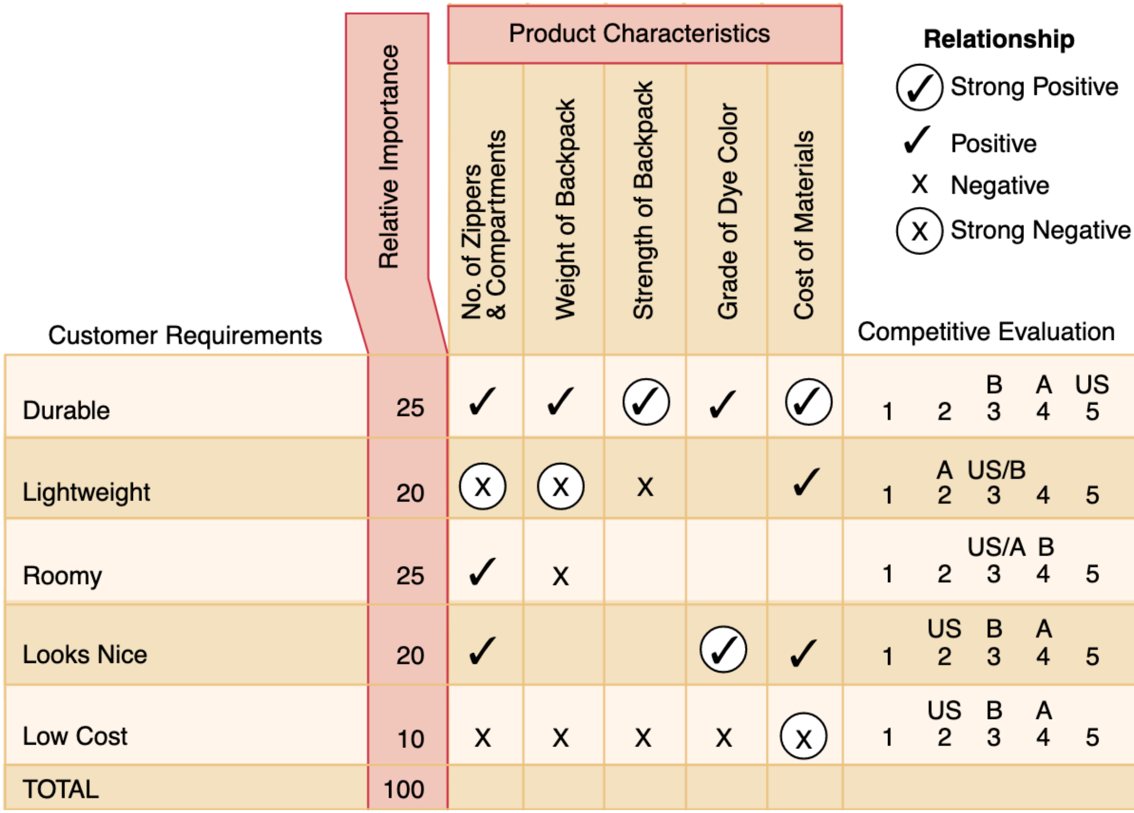
Relationship

Ⓥ	Strong Positive
✓	Positive
X	Negative
ⓧ	Strong Negative

US = Our Backpack
A = Competitor A
B = Competitor B

QFD - The Relationship Matrix

- The strength of the relationship between customer requirements and product characteristics is shown in the relationship matrix
- A negative relationship means that as we increase the desirability of one variable we decrease the desirability of theother.



Customer Requirements	Relative Importance	Product Characteristics					Competitive Evaluation				
		No. of Zippers & Compartments	Weight of Backpack	Strength of Backpack	Grade of Dye Color	Cost of Materials	1	2	3	4	5
Durable	25	✓	✓	(✓)	✓	(✓)			B	A	US
Lightweight	20	(X)	(X)	X		✓		A	US/B		
Roomy	25	✓	X						US/A	B	
Looks Nice	20	✓			(✓)	✓		US	B	A	
Low Cost	10	X	X	X	X	(X)		US	B	A	
TOTAL	100										

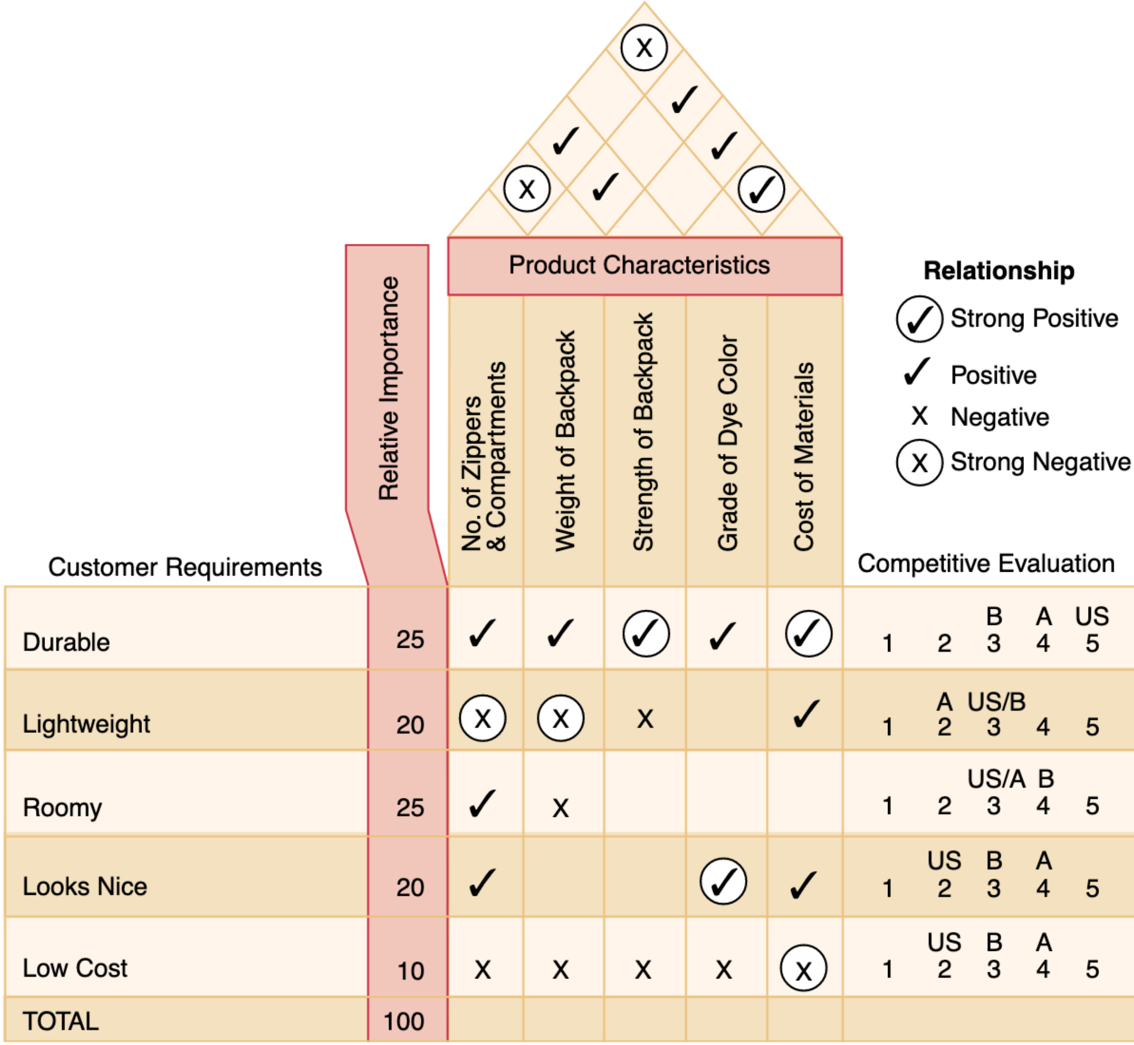
Relationship

(✓)	Strong Positive
✓	Positive
X	Negative
(X)	Strong Negative

US = Our Backpack
A = Competitor A
B = Competitor B

QFD - The Trade-off Matrix

- This is done through a trade-off matrix, which shows how each product characteristic is related to the others and thus allows us to see what tradeoffs we need to make
- For example, the number of zippers is negatively related to the weight of the back-pack



Product Characteristics

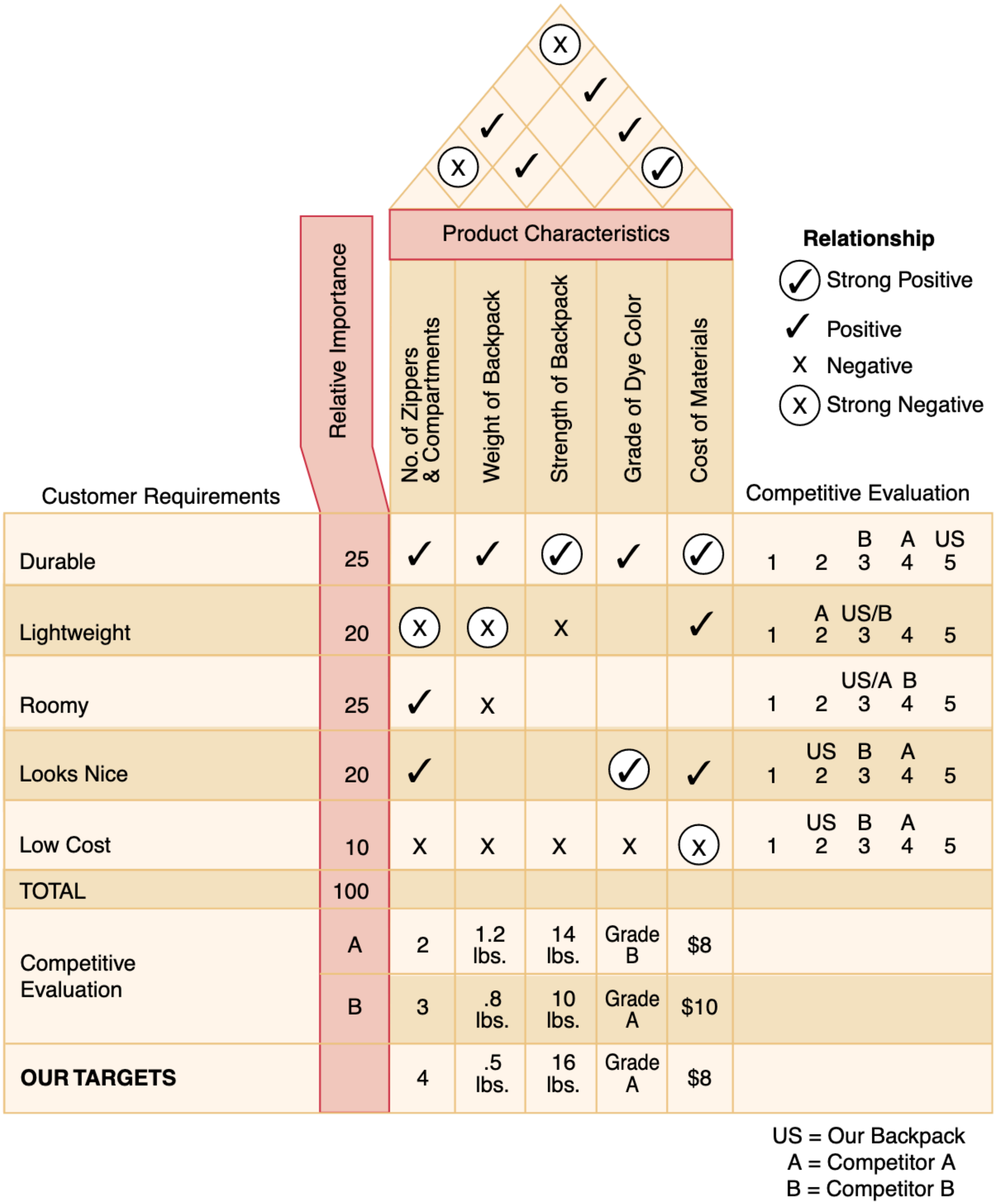
Relationship
✓⃝ Strong Positive
✓ Positive
X Negative
Ⓧ Strong Negative

Customer Requirements	Relative Importance	No. of Zippers & Compartments	Weight of Backpack	Strength of Backpack	Grade of Dye Color	Cost of Materials	Competitive Evaluation				
Durable	25	✓	✓	✓⃝	✓	✓⃝	1	2	B 3	A 4	US 5
Lightweight	20	Ⓧ	Ⓧ	X		✓	1	A 2	US/B 3	4	5
Roomy	25	✓	X				1	2	US/A 3	B 4	5
Looks Nice	20	✓			✓⃝	✓	1	US 2	B 3	A 4	5
Low Cost	10	X	X	X	X	Ⓧ	1	US 2	B 3	A 4	5
TOTAL	100										

US = Our Backpack
A = Competitor A
B = Competitor B

QFD - Setting Targets

- The last step in constructing the house of quality is to evaluate competitors' products relative to the specific product characteristics and to set targets for our own product
- The bottom row of the house is the output of quality function deployment.
- These are specific, measurable product characteristics that have been formulated from general customer requirements



Customer Requirements	Relative Importance	Product Characteristics					Competitive Evaluation				
		No. of Zippers & Compartments	Weight of Backpack	Strength of Backpack	Grade of Dye Color	Cost of Materials	1	2	3	4	5
Durable	25	✓	✓	✓⃝	✓	✓⃝			B	A	US
Lightweight	20	ⓧ	ⓧ	X		✓		A	US/B		
Roomy	25	✓	X						US/A	B	
Looks Nice	20	✓			✓⃝	✓		US	B	A	
Low Cost	10	X	X	X	X	ⓧ		US	B	A	
TOTAL	100										
Competitive Evaluation	A	2	1.2 lbs.	14 lbs.	Grade B	$8					
	B	3	.8 lbs.	10 lbs.	Grade A	$10					
OUR TARGETS		4	.5 lbs.	16 lbs.	Grade A	$8					

Relationship

✓⃝ Strong Positive

✓ Positive

X Negative

ⓧ Strong Negative

US = Our Backpack
A = Competitor A
B = Competitor B

QFD – Case Study - What Do Customers Want?

GCA was a Chicago-based advertising agency that developed campaigns and promotions for small- and medium-sized firms. Their expertise was in the retail area, **but they worked with a wide range of firms from the food service industry to the medical field**. GCA competed on price and speed of product development. Advertising in the retail area was competitive and price had always been important. Also, since retail fashions change rapidly, speed in advertising development was thought to be critical. George reminded himself that price and speed had always been what customers wanted. Now he felt confused that he really didn’t know his customers. This was just another crisis that would pass, he told himself. But he needed to deal with it immediately

1. Offer suggestions to George Stein on ways of identifying quality dimensions GCA’s customers consider important.

Go to **www.menti.com** and use the code **45 86 8**

Risk Management (some pointers)

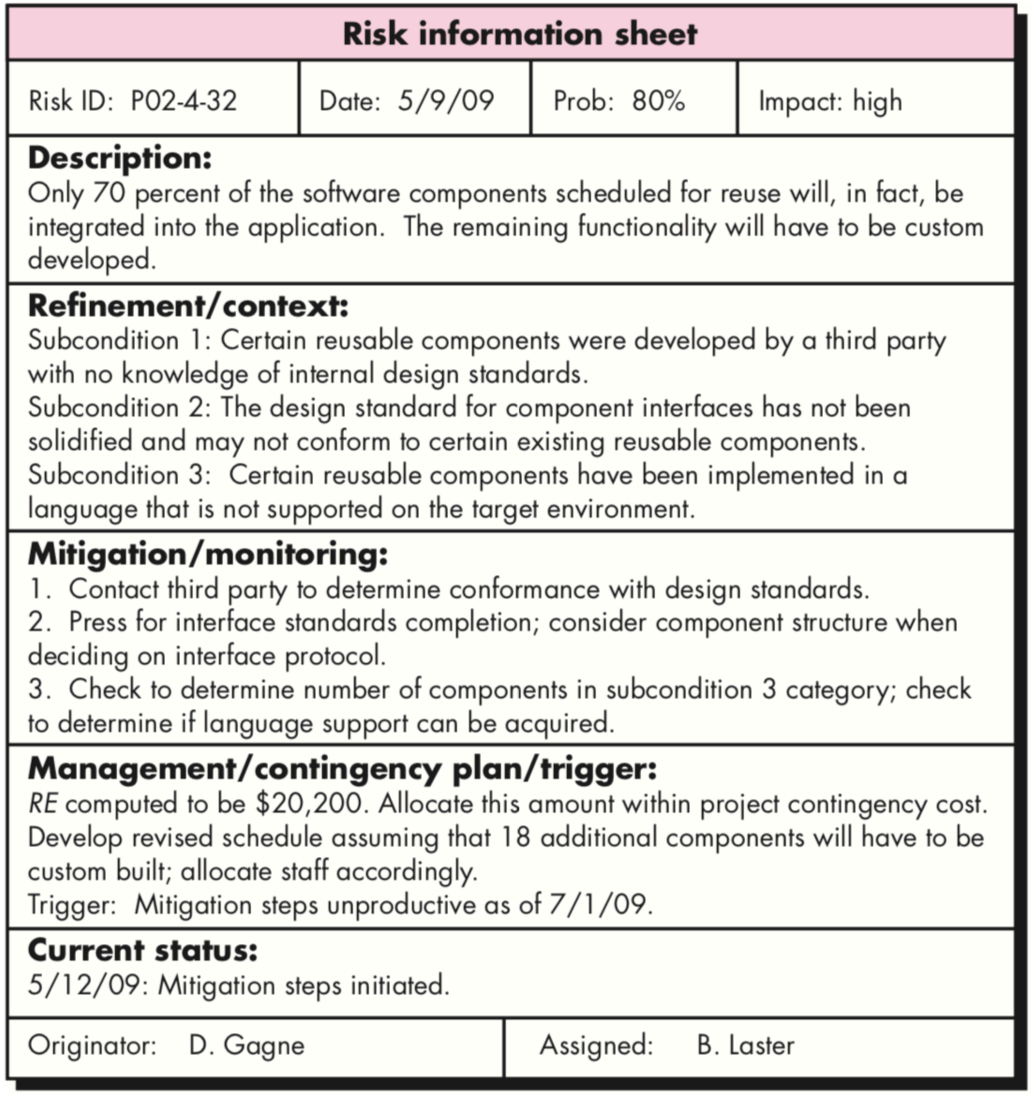
Reactive: At best, a reactive strategy monitors the project for likely risks. Resources are set aside to deal with them, should they become actual problems. More commonly, the software team does nothing about risks until something goes wrong

Proactive: 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

Risk Management (some pointers)

Chapter 28 in Rogger S. Pressmen

- Risk identification
- Risk estimation
- Risk refinement
- Risk mitigation, monitoring and management



Risk information sheet

Risk ID: P02-4-32	Date: 5/9/09	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 7/1/09.

Current status:
5/12/09: Mitigation steps initiated.

Originator: D. Gagne	Assigned: B. Laster

Muddy Card Time

- Thank You.