Course Structure

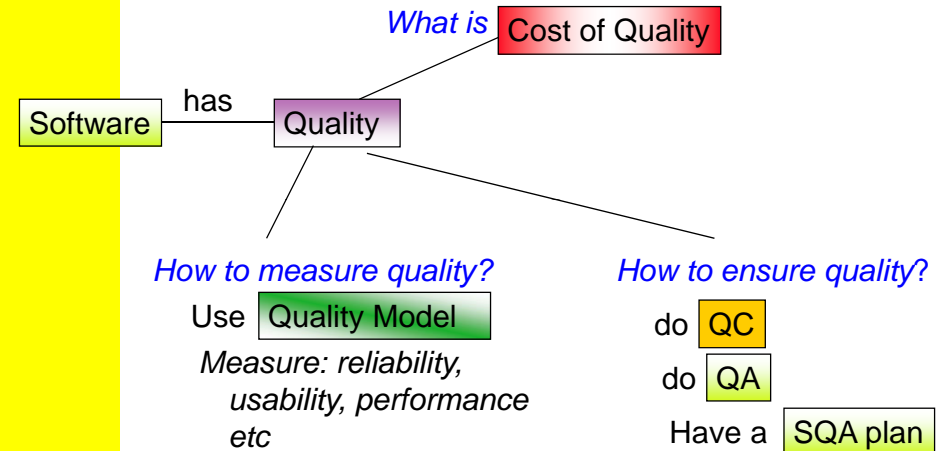1. **Software Quality Assurance**
2. Testing Fundamentals
3. Code-based Techniques
4. Specification-based Techniques
5. Inspection Technique
6. Test Tools
7. Measuring Software Quality
8. TDD

Software Quality Assurance

What is Cost of Quality

Software — has — Quality

How to measure quality?
Use Quality Model
Measure: reliability, usability, performance etc

How to ensure quality?
do QC
do QA
Have a SQA plan

Learning Objectives

- Define **quality** and learn a software **quality model**
- Understand **cost of quality**
- Know the difference between **QC** and **QA**
- Learn the **activities, roles, tools** of **SQA** and **content of quality plan**

Software and systems engineering vocabulary:
http://www.computer.org/sevocab

"What can you do today without software?"

- Introduction
- Definition
- Cost of Quality
- QC
- SQA

Many System Functionality Requires Software

Weapon System	Year	% of Functions Performed in Software
F-4	1960	8
A-7	1964	10
F-111	1970	20
F-15	1975	35
F-16	1982	45
B-2	1990	65
F-22	2000	80
Source: PM Magazine		

Many Large and Complex Systems

GM cars have 100 MLOC (2010).

Do you know?
S-Class Mercedes has as many microprocessors as the new Airbus A380!

- IBM AS/400 system software has 45 MLOC. Each new release has 2-3 MLOC of new and changed code.
- Boeing 787 flight control system has 6.5 MLOC, 3 times as much as the Boeing 777.
- Boeing 777 has 1280 onboard processors
- Blackhawk helicopters have almost 2000 pounds of wire connecting all the computers and sensors.

State of the Software

From US DOD, Software projects

60%	Schedule overruns
50%	Cost overruns
45%	**Unusable**
29%	Never delivered
19%	Need rework
2%	Usable exactly as delivered

From IBM on 24 leading commercial companies:
- Every 6 large-scale software systems put into operation, 2 others are canceled.
- Average software project overshoots 50% schedule
- 68% projects overran their schedules.
- 3/4 of all large systems either do not function as intended or are not used.

Microsoft Software Not Much Better

According to an internal Microsoft memo, the Windows 2000 product had
- over 27,000 bugs
- 21,000 'postponed' bugs
- over 65,000 issues.

Ref: Mary J. Foley, 'BugFest! Win2000 has 63,000 defects', ZDNet News, Feb, 2000

Open Source Software seems better

PHP:	0.474 def/KLOC
LAMP:	0.29 def/KLOC
Average of 32 OSS programs:	0.434 def/KLOC
	Data from 2007

The High Cost of Failures

Software defects have
- Wrecked a European satellite launch
- Destroyed a NASA Mars mission
- Induced a U.S. Navy ship to destroy a civilian airliner,
- Shut down ambulance systems in London, leading to as many as 30 deaths
+ many more failures

Denver Airport
⌘ $1.1M per day lost due to defects in baggage handling system.
American Airlines
⌘ System error incorrectly showed flights full - $50M loss!
⌘ Outages in SABRE cost $20000/minute!

Cost of Downtime (per Hour)

2006 data:

• Brokerage operations	US$6450,000
• Credit card authorization	$2,600,000
• eBay	$225,000
• Amazon.com	$180,000
• Home shopping channel	$113,000
• Airline reservation center	$90,000
• Online network fees	$25,000
• ATM service fees	$14,000

Aug 2012 – Knight Capital

A problem in their trading software, which sent many erroneous orders in 140 stocks listed in NYSE!
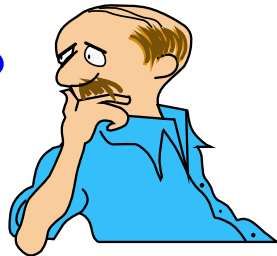
Results:

- Lost of US$440 million!
- Their stock value dropped by 70%!

Quality ?

"Quality is generally transparent when present, but easily recognized in its absence."   *Gillies*

"Quality, like beauty, is in the eye of the beholder.   *Unknown author*

Which pen has higher Quality ?

MONT BLANC Starwalker
$2500

STABILO
$9

What is Quality?

Quality is not a single idea – it has many aspects !!

Product quality can be expressed in terms of
- *Conformance to requirements* – including timeliness, cost
- *Fitness for use* – does it actually do the job?
- *Freedom from errors and failures* – is it reliable and robust?
- *Customer satisfaction* – are users happy with it?

SQA

Quality Definitions

1. **Quality is Factual**
   - ⌘ Does what the requirement says (satisfy the requirement), but software that only meets requirements is still often inadequate!
   - ⌘ **Developer-based** view

2. Quality is **Perceived**
   - ⌘ User/customer is perceiver
   - ⌘ Does what the user expects (meet expectation)
   - ⌘ **User-based** view

SQA

ISO Quality Definition

"Quality is conformance to explicit stated functional and performance requirements, explicitly <u>documented development standards</u>, and implicit characteristics that are expected of the developed software."

- This definition combines the developer-based and user-based views on quality.
- Quality software satisfies the <u>stated requirements</u> and also meets the <u>user expectation</u> (which is implicit and hard to identify.)

Therefore, Software requirements determine the quality of the system!

SQA

Common Questions on Quality
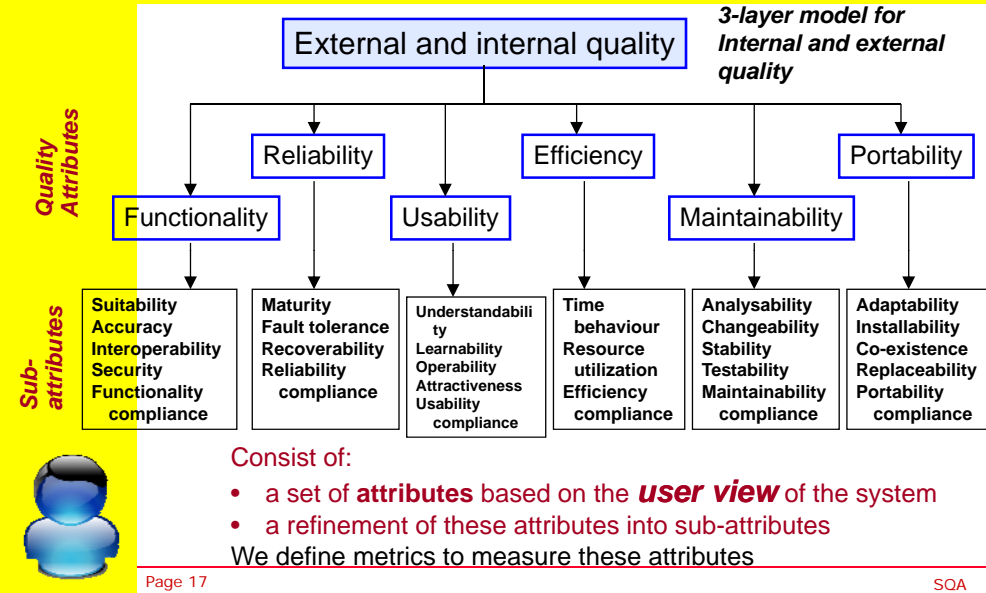
1. *Does Quality = Absence of Errors?*

2. *Does Quality = Conformance to Requirements?*

3. *Is there a standard definition of software quality that is stated in quantifiable terms?*

4. *Is software quality absolute?*

5. *Is software quality specific to a particular system?*

SQA

ISO 9126 Quality Model

3-layer model for Internal and external quality

External and internal quality

Quality Attributes

	Reliability		Efficiency			Portability	

Functionality | Usability | | Maintainability

Sub-attributes

Suitability Accuracy Interoperability Security Functionality compliance	Maturity Fault tolerance Recoverability Reliability compliance	Understandability Learnability Operability Attractiveness Usability compliance	Time behaviour Resource utilization Efficiency compliance	Analysability Changeability Stability Testability Maintainability compliance	Adaptability Installlability Co-existence Replaceability Portability compliance

Consist of:
- a set of **attributes** based on the **user view** of the system
- a refinement of these attributes into sub-attributes

We define metrics to measure these attributes

Quality Attributes (1)

Attribute	Definitions
Correctness	Extent to which a program satisfies its specifications and fulfills the user's objectives (measured in terms of defect rate – defects/KLOC)
Reliability	Extent to which a program can be expected to perform its intended function with required precision (measured in terms of failure rate – failure/hour)
Efficiency	Amount of computing resources and code required by a program to perform a function
Integrity	Extent to which access to software or data by unauthorized persons can be controlled
Usability	Effort required to learn, operate, prepare input, and interpret output of a program (measured in terms of user satisfaction - % of users happy with the interface)

Quality Attributes (2)

Attribute	Definitions
Maintainability	Effort required to locate and fix an error in an operational program (measured in terms of change logs – time and effort required to add a new feature)
Testability	Effort required to test a program to ensure that it performs its intended function
Flexibility	Effort required to modify an operational program
Portability	Effort required to transfer a program from one hardware configuration and software system environment to another
Reusability	Extent to which a program can be used in other applications
Interoperability	Effort required to couple one system with another

---

Mapping User's Needs to Quality Attributes

User's Needs	User's Concerns	Quality Attribute
Functional	How secure is it?	Integrity
	How often will it fail?	Reliability
	Can it survive during failure?	Survivability
	How easy is it to use?	Usability
Performance	How much resource is needed?	Efficiency
	Does it comply with requirement?	Correctness
	Does it prevent hazards?	Safety
	Does it interface easily?	Interoperability
Change	How easy is it to repair?	Maintainability
	How easy is it to expand?	Expandability
	How easy is it to change?	Flexibility
	How easy is it to transport?	Portability
	Is it reusable in other system?	Reusability
Management	Is the software easily managed?	Manageability
	Is performance verification easy?	Verifiability

Quality Attributes

- ⌘ Different parts of software system can have different combinations of desired quality characteristics.
- ⌘ **Efficiency** may be critical for certain components, while **usability** is important for others.
- ⌘ Identify quality characteristics that apply to the entire product from those that are specific to certain components, certain user classes, or particular usage situations.
- ⌘ Document quality characteristics in the requirements specification.
- ⌘ Try to define the priorities and preferences of these quality attributes.

Quality Attributes Depend on System Types

TYPE OF SOFTWARE	PORTABILITY	RELIABILITY	EFFICIENCY	PERFORMANCE	ROBUSTNESS	CORRECTNESS
System	N	Y	Y	Y	Y	Y
Real-Time	N	Y	Y	Y	Y	Y
Business	Y	Y	N	N	N	Y
Engineering and Scientific	Y	Y	N	N	N	Y
Embedded	N	Y	Y	Y	Y	Y
Personal Computer	Y	N	N	N	N	Y
Artificial Intelligence	Y	N	N	N	N	Y

Y: important, N: not relevant


Quality Attributes in Practice

The most common quality attributes according to the SEI-led ATAM evaluations.

Quality attribute	Distribution (%)
Modifiability	14.1
Performance	13.6
Usability	11.4
Maintainability	8.5
Interoperability	7.8
Security	7.3
Configurability	6.9
Availability	6.8
Reliability	5.7
Scalability	3.2
Testability	2.6
Affordability	2.0
Reusability	1.9
Integrability	1.9
Safety	1.1
User data management	1.0
Portability	0.8
Assurance	0.8
Product line	0.8
Net-centric operation	0.5

From SEI study of 24 projects (1999-2007)

When We Buy a New Car
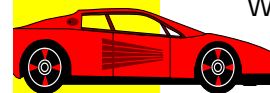
Will we accept delivery on an expensive new automobile if it was in **perfect condition** except for a big scratch on the door?

The car would deliver the same fuel economy, same performance, same cargo capacity, same expected service lifetime.

NO. We paid for a **perfect product**, and that is what we expected to get!

No one expects
- a building to fall,
- a bridge to collapse,
- a train to derail or
- a plane to crash.
Not so with software!!

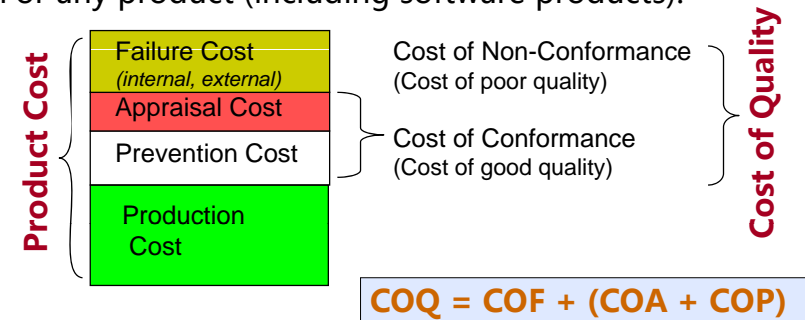# Why do we accept low quality software?

Quality Costs

For any product (including software products):

Product Cost
- Failure Cost *(internal, external)* — Cost of Non-Conformance (Cost of poor quality)
- Appraisal Cost — Cost of Conformance (Cost of good quality)
- Prevention Cost — Cost of Conformance (Cost of good quality)
- Production Cost

Cost of Quality

$$COQ = COF + (COA + COP)$$

Prevention: costs incurred attempting to prevent or avoid errors

Appraisal: costs incurred attempting to detect errors

Failure: costs incurred because the other attempts were not successful

---

Internal vs External Failure Cost

Software released

Internal failure costs: failure costs that arise before releasing the product to the customer.

Examples: costs of finding and fixing defects; if a defect prevents someone from doing her job, the costs of the wasted time, the missed milestones, overtime, etc.

External failure cost: failure costs that arise after releasing the product to the customer

Examples: customer service costs, the cost of patch.

Quality cost your customer too!

Customer also suffers quality-related costs.

He faces significant expenses in dealing with the bad product.

Sometime, poor quality may cost the customer much, much more than it costs the developer.

Customer Failure Costs:
- Wasted time
- Lost data
- Lost business
- Embarrassment
- Frustrated employees
- Cost of replacing product
- Cost of reconfiguring the system
- Cost of recovery software
- Cost of tech support
- Injury/death

Example Cost Items

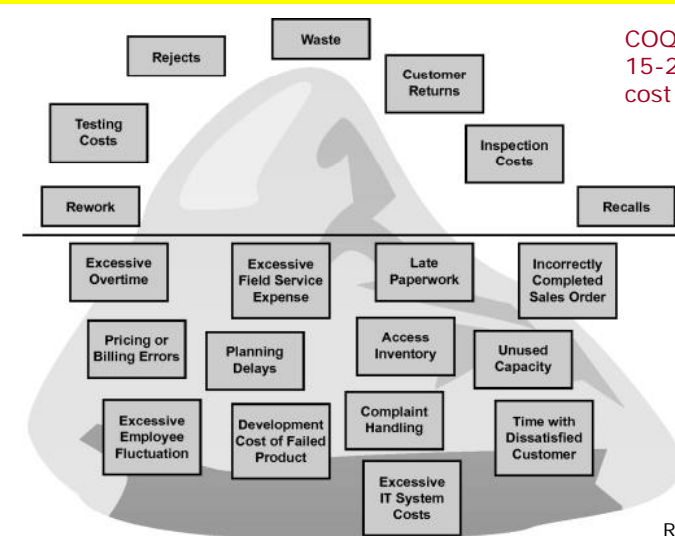| Prevention	Appraisal
Represents everything a company spends to prevent software errors, documentation errors, and other product-related errors. • Staff training • Requirements analysis • Early prototyping • Fault-tolerant design • Defensive programming • Usability analysis • Clear specifications • Accurate internal documentation • Pre-purchase evaluation of the reliability of development tools	Includes the money spent on the actual testing activity. Any and all activities associated with searching for errors in the software (and associated product materials) fall into this category. • Design reviews • Code inspection • Glass box testing • Black box testing • Beta testing • Test automation • Usability testing • Pre-release out-of-box testing by customer service staff
Internal Failure	**External Failure**
The cost of coping with errors discovered during development and testing. These are bugs found before the product is released. • Bug fixes • Regression testing • Wasted in-house user time • Wasted tester time • Wasted writer time • Wasted marketer time • Wasted advertisements • Direct cost of late shipment • Opportunity cost of late shipment	The costs of coping with errors discovered after the product is released. These are typically errors found by your customers. • Technical support calls • Answer books (for support) • Investigating complaints • Refunds and recalls • Interim bug fix releases • Shipping product updates • Warranty, liability costs • Public relations to soften bad reviews • Lost sales • Lost customer goodwill • Supporting multiple versions in the field • Reseller discounts to keep them selling the product

---

Iceberg Model of COQ

COQ is typically 15-25% of total cost

Ref: George Webb, LTC N Patton

SQA

High Payback in Prevention

According to QAI,

- Prevention costs have a **10:1** payback.

- Appraisal cost (cost of detecting defects) have a **3:1** payback.

SQA

Calculating the ROI based on COQ

	A	B	C	D
1	**Testing Investment Options: ROI Analysis**			
2				
3		**No Formal**	**Manual**	**Automated**
4	**Testing**	**Testing**	**Testing**	**Testing**
5	Staff	$0	$60,000	$60,000
6	Infrastructure	$0	$10,000	$10,000
7	Tools	$0	$0	$12,500
8	Total Investment	$0	$70,000	$82,500
9				
10	**Development**			
11	Must-Fix Bugs Found	250	250	250
12	Fix Cost (Internal Failure)	$2,500	$2,500	$2,500
13				
14	**Testing**			
15	Must-Fix Bugs Found	0	350	500
16	Fix Cost (Internal Failure)	$0	$35,000	$50,000
17				
18	**Customer Support**			
19	Must-Fix Bugs Found	750	400	250
20	Fix Cost (External Failure)	$750,000	$400,000	$250,000
21				
22	**Cost of Quality**			
23	Conformance	$0	$70,000	$82,500
24	Non-conformance	$752,500	$437,500	$302,500
25	**Total CoQ**	**$752,500**	**$507,500**	**$385,000**
26				
27	**Return on Investment**	**#N/A**	**350%**	**445%**

245000/70000

367500/82500

SQA

Exercise

Classify the following costs into Prevention, Failure and Appraisal:

(a) Cost of doing inspections
(b) Cost of testing after a possible fix is made but before the fix is shipped
(c) Customer downtime costs
(d) Re-making the CDs for software distribution after a fix is verified.
(e) Time spent by the programmers to arrive at a fix for a reported defect
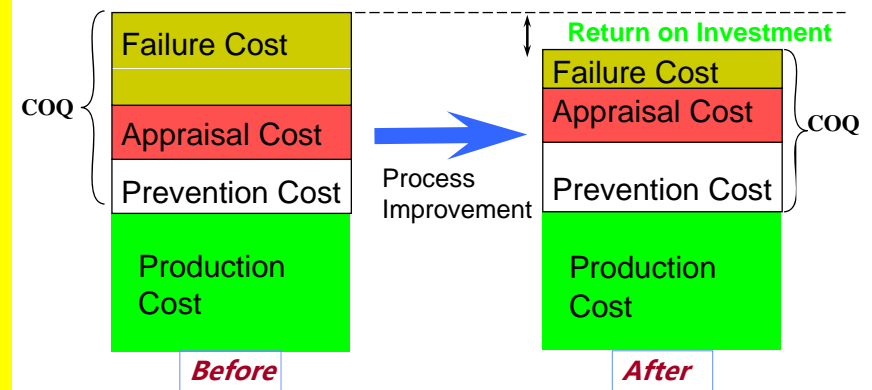(f) Cost of ISO9001 audits

---

Quality is Free

Return on Investment

COQ { Failure Cost | Appraisal Cost | Prevention Cost }

Production Cost

Before

Process Improvement

Failure Cost | Appraisal Cost | Prevention Cost } COQ

Production Cost

After

Cost of implementing Process Improvement can be funded by the saving from reduction in COQ

"It takes money to save money."

Quality Control (QC)

Quality Control: the set of activities (inspections, reviews, and testing throughout the development cycle) intended to detect, document, analyze, and correct product defects and to manage product changes

Down to earth definition: *QC is defect control, error control, frustration control, and the control of all negative problems related to quality.*

QC people
• concerned with **products**
• a <u>data collector</u> for QA's process analysis

Common Misconceptions

• Quality can only be tested when the code is done. --- NO

• Quality can only be improved by removing defects from the code. --- NO

How can we remove errors earlier?

• Do reviews, inspections, and walkthroughs (static analysis) of the **requirements** and **design**!
• Try to prevent defects getting into the source code.

Software Quality Assurance (SQA)

SQA: <u>Set of activities</u> designed to evaluate the **process** by which software work products are developed and maintained (focus on **Process**, rather than product)

SQA encompasses:
- A quality management approach
- Effective software engineering technology
- Formal technical reviews
- Testing strategy
- Document change control
- Software development standard and control procedure
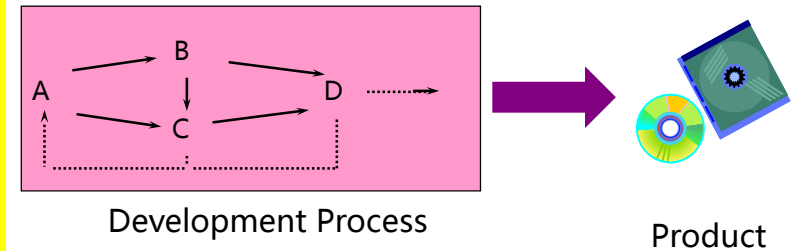- Measurement and reporting mechanism

---

Assumption of SQA

Development Process   Product

A <u>quality process</u> produces a <u>quality product</u>

Thus, only need to look at the "Process"

SQA

Key goal: achieve high quality software product.   *How?*
- ⌘ Monitor the software and the development process
- ⌘ Ensure compliance with standards and procedures
- ⌘ Bring needs for improvement to managers.

Benefits
- ☺ reduce the number of defects and thus reduce the testing time and effort
- ☺ cause higher reliability that will result in greater customer satisfaction
- ☺ reduce maintenance costs
- ☺ reduce life cycle cost of software

Software Quality System or
Software quality management system

<u>Software quality system</u> consists of the managerial structure, responsibilities, activities, capabilities and resources to ensure that the developed software products have the desired quality.

Details of the quality management system is contained in a **quality manual**, which *contains* **standards, procedures and guidelines** (*influenced by external standards, such as ISO 9001.*)

- a **standard** is instruction of how a project document or program code is to be developed
- a **procedure** is a <u>step-by-step set of instructions</u> describing how a particular activity is to be done
- a **guideline** consists of <u>advice</u> on best practice.

Quality Planning

Quality Manual

Quality planning

Input: *Risk assessment, Specific requirements of project*
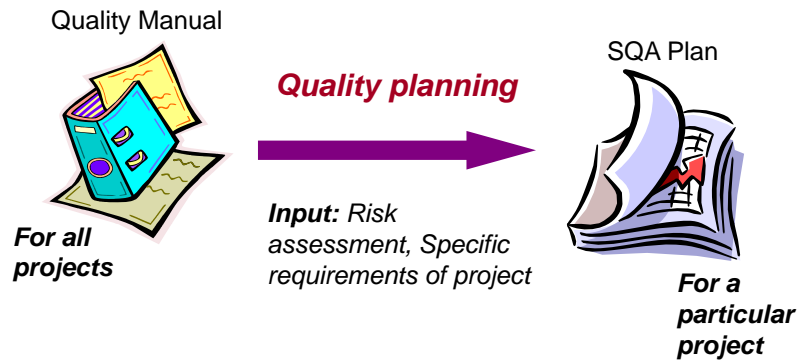
SQA Plan

For all projects

For a particular project

Most companies should have a Quality Manual defined for their quality management system, which should be applicable for almost all projects

---

SQA Plan (for a project)

⌘ A plan for conducting the QA activities

⌘ Provides a road map for doing SQA.

⌘ Defines how adherence to standards will be monitored (e.g., coding standards)

⌘ Include
- ➜ quality standards, methodologies, procedures, and tools for performing the QA activities
- ➜ procedures for contract review and coordination
- ➜ procedures for identification, collection, filing, maintenance, and disposition of quality records
- ➜ resources, schedule, responsibilities
- ➜ plans for monitoring each activity
- ➜ checklist for activities that have to be carried out to assure the quality of the product.

IEEE Standard for SQAP

- IEEE Std 730-1998
  - Standard for Software Quality Assurance Plans
- IEEE Std 730.1-1995
  - Guide for Software Quality Assurance Planning

Plan to do quality work. If you don't plan for quality, you probably won't get it.   *Watts S. Humphrey*

Contents of SQA Plan

Section 1 - Purpose
- list software covered
- state portion of software life cycle covered

Section 2 - Reference Documents
- complete list of documents referenced elsewhere

Section 3 - Management
- organization structure
- tasks
  - tasks to be performed
  - relationship between tasks and checkpoints
  - sequence of tasks
- Responsibilities of each organizational unit

Section 4 - Documentation
- identify required documents
- state how documents will be evaluated
- minimum documents
  - SRS - Software Requirements Specification
  - SDD - Software Design Description
  - SVVP – S. Verification and Validation Plan
  - SVVR - S. Verification and Validation Report
  - User documentation - manual, guide
  - SCMP – S. Configuration Management Plan

Introduction
Definition
Cost of Quality
QC
SQA
▶**Plan**
Tools
Roles

Section 5- Standards, Practices, Conventions and Metrics

- Identify standards, practices, conventions and metrics to be applied
- How compliance is to be monitored and assured
- Minimum
  - documentation standards, logic structure standards, coding standards, testing standards
  - selected SQA product and process metrics
    - e.g., branch, decision points coverage

Section 6 - Reviews and Audits

Purpose:
- define what reviews/ audits will be done,
- how they will be accomplished
- what further actions are required

Minimum set of activities:
- Software Requirements Reviews
- Preliminary Design Review
- Critical Design Review
- SVVP Review
- Functional Audit (check all requirements in SRS have been met)
- Physical Audit (software and documents are consistent and ready)
  In Process Audit
- Managerial Reviews

Introduction
Definition
Cost of Quality
QC
SQA
▶**Plan**
Tools
Roles

Section 7 - Test

- All tests that are included in SVVP

Section 8 - Problem Reporting

- Practices and Procedures for reporting, tracking, and resolving problems
- Organizational responsibilities

Section 9 - Tools, Techniques & Methodologies

- identify the special software tools, techniques and methodologies
  - purpose
  - describe use

Section
10 - Code Control
11 - Media Control
12 - Supplier Control (for outsourcing)
13 - Records - collection, maintenance and retention
14 - Training
15 - Risk Management

Sample

Introduction
Definition
Cost of Quality
QC
SQA
Plan
▶**Tools**
Roles

Tools of SQA

1. **Measurements** - draw conclusions on process adherence based on measurement. SQA does not collect data, but use the raw and analysed data.

   Techniques used in analysis include:
   trend data, direct measurement, failure analysis, process cause and effect, Pareto analysis, and task entrance/exit criteria.

2. **Audits** - check adherence to working procedures and to verify project progress
3. **Review** - inspect project documents

Introduction
Definition
Cost of Quality
QC
SQA
Plan
▶**Tools**
Roles

Use of Measurement Example : Defects by Program

Pareto Chart

100 %
96%
84%

Defects

27
4
1

Program B   Program A   Program C
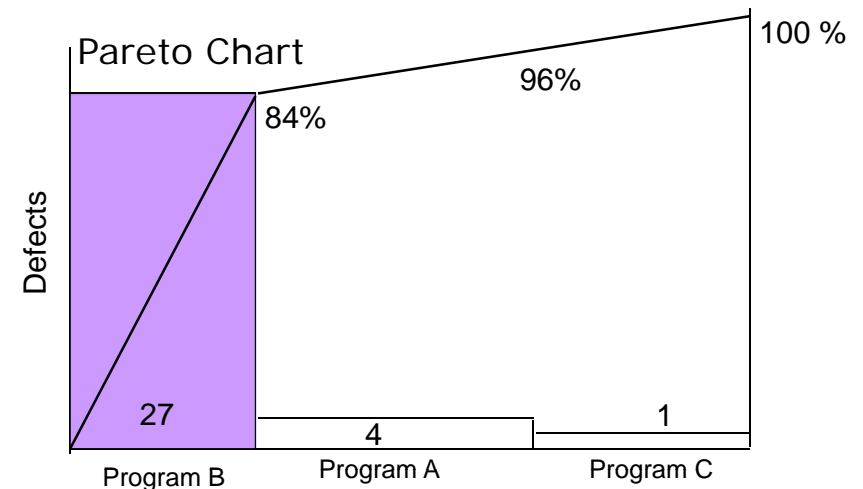
Use of Measurement Example : Defect Types Distribution

National Software Quality Experiment: 1992-2001

Percent of Defect Types
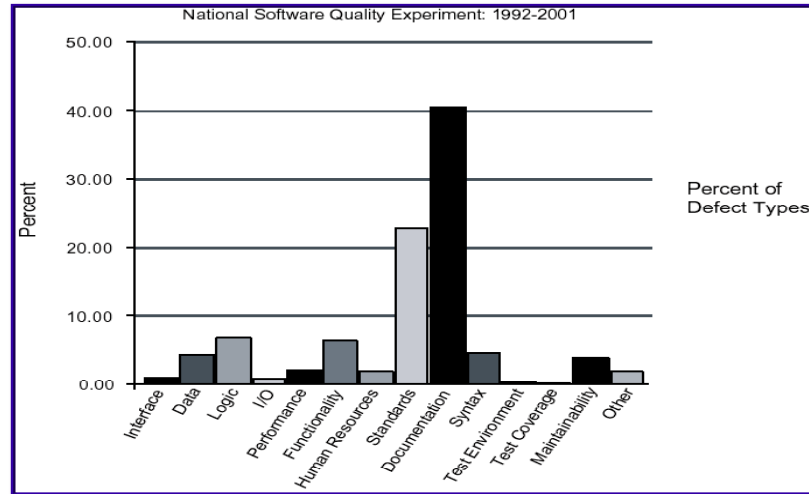
Another Example: Use of Measurement

A company use 1, 2, 3, and 4 to prioritize customer reported problems. (1 is most severe). But, priority 3 and 4 problems are easier to locate/fix than priority 1 and 2 problems. The following table show the <u>problem report</u> for June.

No.	priority	report date	start diagnosis	fix date
101	3	1/6	5/6	6/6
102	2	2/6	4/6	8/6
103	4	2/6	2/6	3/6
104	1	2/6	4/6	7/6
105	1	4/6	4/6	8/6
106	3	4/6	4/6	5/6
107	4	10/6	10/6	11/6
108	2	10/6	14/6	16/6

- What do you infer from this data?
- What corrections does the organization need?

Quality Audit (1)

A systematic and <u>independent</u> examination to determine whether quality activities and related results comply with planned arrangements and whether these arrangements are implemented effectively and are suitable to achieve objectives.

Type of Quality Audit
- <u>Internal</u> / External
- <u>Planned</u> / Ad hoc / Follow-up
- <u>Quality System Audit</u> / Process Quality Audit / Product Quality Audit / Service Quality Audit

Most common is internal, planned, quality system audit.

Quality Audit (2)

Purpose: Evaluate the need for improvement or corrective action

Characteristics
- Involve 2 parties: auditor and audited party (project team)
- Require trained and competent audit team
- Provide Audit Report containing findings to the <u>senior management</u>
- Lead to corrective actions and consequent improvements
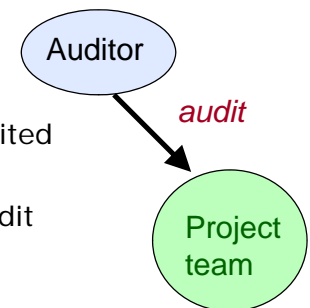
Auditor

audit

Project team

Software Reviews

Purposes: *Reviews are applied at various points during software development and serve to* **uncover errors** *in* analysis, design, code and testing.

A software review uses a group of people to:
- point out needed improvements to **product**
- confirm those parts of the product in which improvement is needed
- achieve technical work of more uniform quality

2 types of reviews:
- **informal reviews**: informal meetings and informal desk checking
- **formal reviews**: walkthroughs, inspection

Benefit:
The early discovery of software defects so that each defect may be corrected prior to the next phase of development.

Formal Technical Reviews

A formal technical review (FTR) is a SQA activity performed by software engineers with the objectives:
- to uncover errors in function, logic or implementation of the software;
- to verify that the software meets its requirements;
- to ensure that the software has been developed according to the standards;
- to achieve uniform software;
- to make projects manageable.

FTR also serves to educate new team members.

Each FTR is considered successful only if it is properly planned, controlled and attended.

Review Meeting

Review meeting follows these "rules:
- 3 - 5 people should be involved;
- advance preparation should occur but should require no more than 2 hours of work for each person
- Duration of meeting: less than 2 hours.
- Focus on a specific part of the product

Who participate in the review meeting:
- review leader, reviewers (one of them is recorder), and the producer.

The producer organizes a "walk through" of the product, explaining the material, while the reviewers raise issues based on their advance preparation.

Review Reporting & Record Keeping

During the FTR, the reviewer records all issues that have been raised

At the end of the review, make decisions:
- Accept the work product with further modification
- Reject the work product due to errors
- Accept the work under some conditions

Finally, a review summary report is produced, with the following information:
- What was reviewed
- Who reviewed it
- What were the findings and conclusions

Review Guidelines

Guidelines for the conduct of FTR must be established in advance, distributed to all reviewers, agreed upon, and then followed.

Guidelines:
- review the product, not the producer
- set an agenda and maintain it
- limit debate and rebuttal
- don't attempt to solve every problem noted
- take written notes
- limit the number of participants and insist upon advance preparation
- develop a checklist for each product that is likely to be reviewed
- allocate resources and time schedule for FTRs
- conduct meaningful training for all reviewers
- Learn from early reviews

SQA

Responsibilities of SQA

- Review plans for completeness
- Participate in design and code reviews
- Review test plans for completeness
- Audit samples of test results
- Audit Software Configuration Management performance
- Participate in project management reviews

Other Duties:

- Tailor QA procedures to the specific project
- SQA will never evaluate technical content itself, but the adherence to standards, practices and processes.
- SQA will say that code is not documented or reviewed properly, but will never say that the logic is incorrect.

SQA

Relation between QA and QC

QA checks on QC

QA Staff
- QA representative
- Auditors
- QA management
- Assessment body

• Prevent defects

QC Staff
- Project management
- Project staff

- Collect data
- Ensure that the required qualities are built into the product
- find and address defects that have not been prevented

- In theory, QA engineers and testers should do different things, but in reality, QA engineers are usually testers
- A common business practice uses "QA engineer" and "Test engineer" interchangeably.
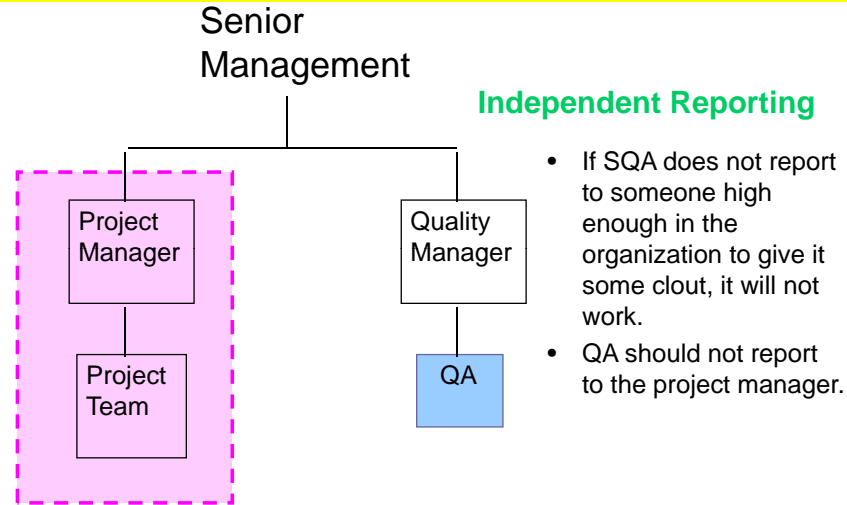
SQA
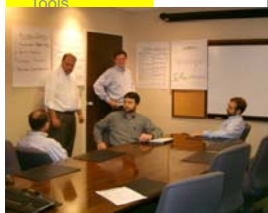
QC | QA

- QC is done after the product is built. Hence it is usually reactive.
- Expensive. For example, correcting the design at the system testing stage.
- Oriented towards **defect detection** rather than prevention.

- QA is proactive. Defect prevention oriented rather than defect detection oriented. We know defect prevention is always better than defect detection.
- Intended to catch the defects as close to the point of injection as possible rather than let the defects trickle down to subsequent levels.
- Applies to the **process** rather than the end-product.

Both QA and QC take places throughout the entire software development process.

SQA

QA Independence & Reporting

Senior
Management

Independent Reporting

Project
Manager

Project
Team

Quality
Manager

QA

- If SQA does not report to someone high enough in the organization to give it some clout, it will not work.
- QA should not report to the project manager.

Different Roles

Role of QA: identify process flaws and weakness, improve the process; QA planning, record keeping, analysis and reporting

Role of management: use information from QA as a guide to improve the software development process, such as:

➔ test process

➔ maintenance process

➔ support process: quality, CM, and other

In small organization, software managers can do the work of SQA, in larger ones they cannot.
Beyond 15-20 developers, a SQA group is a must!

Question

Which activity is QC, which QA :

(a)   Add the code reviews as a means of defect prevention.
(b)   Doing acceptance testing
(c)   Audits for ISO9001
(d)   Running the tests for a product after it is built
(e)   Verifying that all scheduled tests are run

A Story

In ancient China there was a family of doctors.
A doctor was asked which of his family was the best.
"I tend to the sick and dying with drastic and dramatic treatments, and on occasion someone is cured and many people know me."
"My elder brother cures sickness when it just begins, and his skills are known among the local peasants and neighbors."

My eldest brother is able to sense the spirit of sickness and eradicate it before it takes form. His name is unknown outside our home."

Same Story (in Chinese)

魏文王问名医扁鹊说："你们家兄弟三人，都精于医术，到底哪一位最好呢？" 扁鹊答说："长兄最好，中兄次之，我最差。"

文王再问："那么为什么你最出名呢？"

扁鹊答说："我长兄治病，是治病于病情发作之前。由于一般人不知道他事先能铲除病因，所以他的名气无法传出去，只有我们家的人才知道。我中兄治病，是治病于病情初起之时。一般人以为他只能治轻微的小病，所以他的名气只及于本乡里。而我扁鹊治病，是治病于病情严重之时。一般人都看到我在经脉上穿针管来放血、在皮肤上敷药等大手术，所以以为我的医术高明，名气因此响遍全国。"
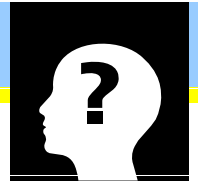
Why Management Don't Support SQA

Solving problems is a high-visibility process.

Preventing problems is low-visibility.

But, which is more beneficial?

SQA

Review Questions

1. What is the definition of quality?
2. How many quality attributes?
3. COQ has how many components?
4. What is the purpose of SQA?
5. Compare Audit and Technical Review.

SQA

Summary

- Software quality is conformance to explicit stated and implicit requirements.
- Software quality may be modeled as a set of attributes, including correctness, reliability, efficiency, testability, usability.
- Cost of quality includes 3 components: COP, COF and COA.
- Quality is free because the saving from improvement can be used to fund the improvement project.
- SQA focuses on the process.
- To be successful, SQA should be independent of the project team

SQA

Reference
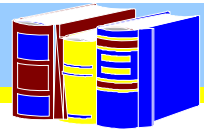
Information technology - Software life cycle processes, ISO/IEC 12207

Lewis, W. E., Software Testing and Continuous Quality Improvement, CRC Press, 2008

O'Regan, G., A Practical Approach to Software Quality, Springer-Verlag, 2002

Schulmeyer, G. G., Handbook of Software Quality Assurance, 4th ed., Artech House, 2008

Software Engineering Standards Committee of the IEEE Computer Society. *IEEE Standard for Software Quality Assurance Plans, IEEE Std 730-2002*.

Software QA Testing and Test Tool Resources, http://www.aptest.com/resources.html

http://www.rspa.com/reflib/QualityMgmt.html#SQA

SQA