UECS3383 Software Quality Assurance

Lecture 03 – Estimating and Measuring Software Quality

Changing View of Quality

Past	Present
Quality is the responsibility of blue collar workers and direct labor employees working on the product	Quality is everyone's responsibility, including, white-collar workers, the indirect labor force and the overhead staff
Quality defects should be hidden from the customers and management	Defects should be highlighted and brought to the surface for corrective actions
Quality problems lead to blame, faulty justification and excuses	Quality problems lead to cooperative solutions
Corrections-to-quality problems should be accompanied with minimum documentation	Documentation is essential for "lessons learnt" so the mistakes are not repeated.

Changing View of Quality

Past	Present
Increased quality will increase project costs	Improved quality saves money and increase business
Quality is internally focused	Quality is customer focused
Quality will not occur without close supervision of people	People want to produce quality products
Quality occurs during project execution	Quality occurs at project initiation and must be planned for within the project

Cost of Quality

- In process improvement efforts, **quality costs** or **cost of quality** is a means to quantify the total **cost of quality**-related efforts and deficiencies.
- The "cost of quality" isn't the price of creating a quality product or service. It's the cost of NOT creating a quality product or service.
- Every time work is redone, the cost of quality increases. Obvious examples include:
  - The reworking of a manufactured item.
  - The retesting of an assembly.
  - The rebuilding of a tool.
  - The correction of a bank statement.
  - The reworking of a service, such as the reprocessing of a loan operation or the replacement of a food order in a restaurant.

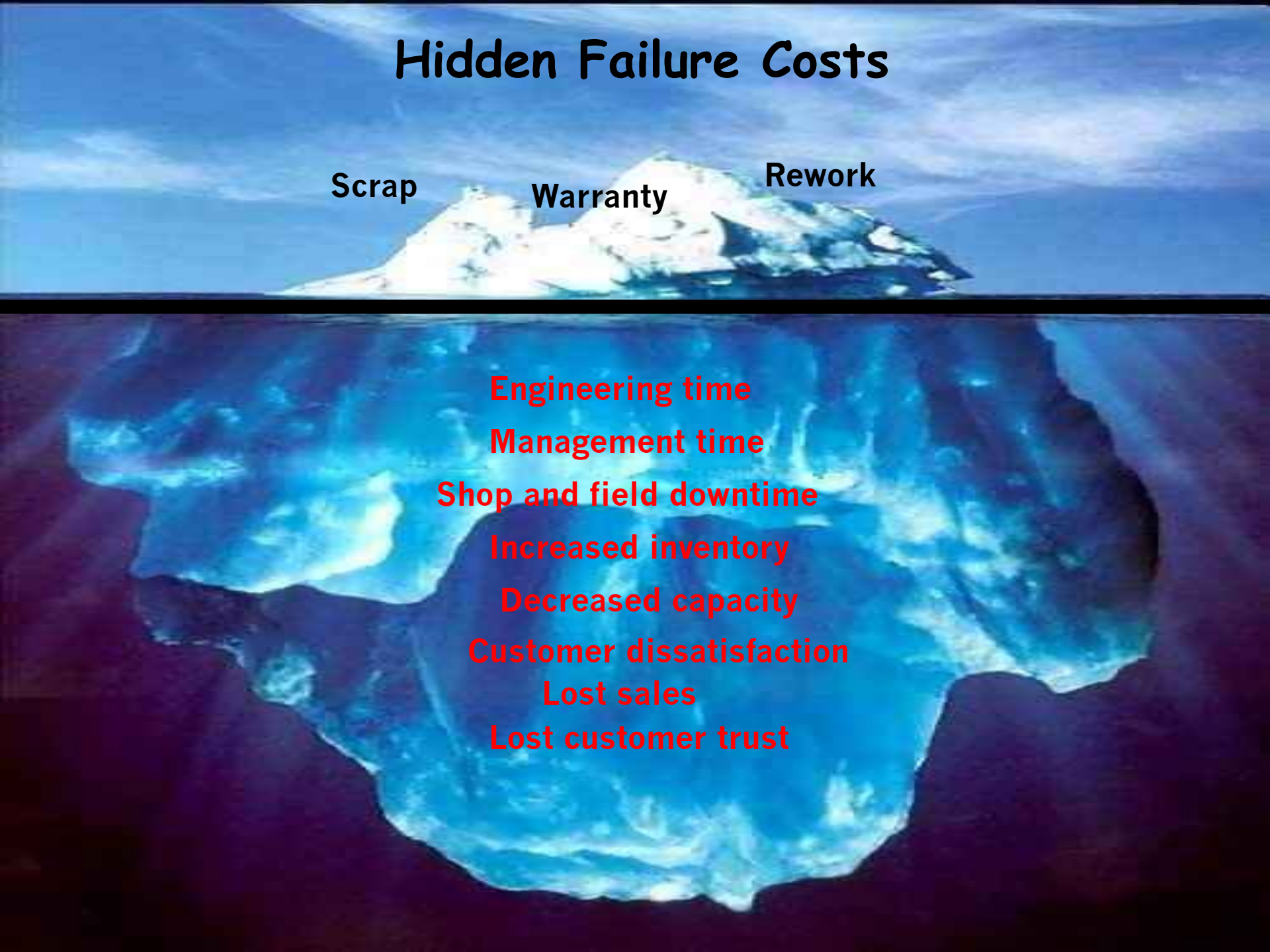
Cost of Quality (COQ)

- In short, any cost that would not have been expended if quality were perfect contributes to the cost of quality.
- Cost of quality refers to the sum of costs incurred to prevent non-conformance from happening and the costs incurred when non-conformance in products and system occurs which is commonly known as “cost of poor quality”
- Cost of poor quality is actually the cost of doing things wrong
- Cost of quality refers to the costs associated with providing poor quality product or service

Why Need to Know COQ?

- This tool speaks in the language of management $$$$
- Research shows that cost of poor quality can range from 15% to 40 % of the business costs
- It can prioritize quality improvement actions
- Cost of quality data shows how profit is affected by quality
- It helps identify the redundant activities

Hidden Failure Costs



Cost of Quality

- Total Quality Costs
  - As the figure below shows, quality costs are the total of the cost incurred by:
    - Investing in the prevention of nonconformance to requirements.
    - Appraising a product or service for conformance to requirements.
    - Failing to meet requirements.
- Categories of Cost of Quality
  - **Cost of achieving good quality**
    - **Prevention Costs**
    - **Appraisal Costs**
  - **Cost of poor quality**
    - **Failure Costs**
    - **External Failure Costs**
  - **Total Quality Costs**

Cost of Quality – Prevention Cost

- The costs of all activities specifically designed to prevent poor quality in products or services.
- These costs are incurred to keep appraisal and failure costs at minimum.
- Examples are the costs of:
  - New product review
  - Quality planning
  - Supplier capability surveys
  - Process capability evaluations
  - Quality improvement team meetings
  - Quality improvement projects
  - Quality education and training

Cost of Quality – Prevention Cost

- Examples are the costs of:
  - Process Capability studies
  - Market surveys
  - Pilot scale projects and testing
  - Procedure writing
  - Vendor evaluation and testing
  - Training and education
  - Quality improvement projects
  - Customer survey
  - House keeping
  - Design review

Cost of Quality – Appraisal Cost

- The costs associated with measuring, evaluating or auditing products or services to assure conformance to quality standards and performance requirements.
- These include the costs of:
  - Incoming and source inspection/test of purchased material
  - In-process and final inspection/test
  - Product, process or service audits
  - Calibration of measuring and test equipment
  - Associated supplies and materials
  - Internal audits
  - Incoming material inspection
  - Laboratory testing
  - Calibration costs
  - In process material inspection
  - Equipment calibration
  - Procedure evaluation
  - Final product inspection
  - Automated testing tools

Cost of Quality – Failure Cost

- The costs resulting from products or services not conforming to requirements or customer/user needs. Failure costs are divided into internal and external failure categories.

Cost of Quality – Internal Failure Cost

- Failure costs occurring prior to delivery or shipment of the product, or the furnishing of a service, to the customer.
- Examples are the costs of:
  - Scrap
  - Rework
  - Re-inspection
  - Re-testing
  - Material review
  - Downgrading

Cost of Quality – Internal Failure Cost

- **Examples:**
  - Overtime
  - Downtime
  - Excess inventory
  - Excess material handling
  - Redesign
  - Downgrading
  - Retesting
  - 100% sorting inspection
  - Scrap & rework - supplier

Cost of Quality – External Failure Cost

- Failure costs occurring after delivery or shipment of the product — and during or after furnishing of a service — to the customer.
- These are the costs incurred by a business due to failure of product or service at the customer end. These costs results into warranty claims and loss of reputation.
- Examples are the costs of:
  - Processing customer complaints
  - Customer returns
  - Warranty claims
  - Product recalls
  - Customer dissatisfaction
  - Loss of market share
  - Price concession
  - Premium freight
  - Time spent to resolve customer complaints
  - Restoking costs
  - Other penalties

Cost of Quality – Total Quality Cost

- The sum of the above costs.
- Total Quality Cost = Prevention cost + Appraisal cost +

  Failure cost + External Failure cost
- This represents the difference between the actual cost of a product or service and what the reduced cost would be if there were no possibility of substandard service, failure of products or defects in their manufacture.

Measuring Cost of Quality

- COQ data can be measured and presented in many different ways.
  - % age of sales
  - % age of profits
  - % age of manufacturing cost
  - Rs per direct labor hour
  - Rs per unit of product

Steps in Implementing Quality Cost

1. Review the literature on quality costs or consult others in similar industries who are using the same tool.
2. Select one organizational unit of the company to serve as a pilot site
3. Discuss the objectives of the study with the key people in the organization
4. Collect whatever cost data are conveniently available from the accounting system
5. Make a proposal to management for a full study

Steps in Implementing Quality Cost

6. Publish a draft of the categories defining the cost of poor quality
7. Finalize the definitions and secure management approval
8. Secure agreement on responsibility for data collection and report preparation
9. Collect and summarize the data
10. Present the cost results to management along with the results of a demonstration quality improvement project

Case study: H&S motors

The H&S motor company produces small motors for use in lawnmowers and garden equipment. The company instituted a quality improvement program in 1999 and has recorded the following quality cost data and accounting measures for 4 years.

Quality Costs	Year			
	1999	2000	2001	2002
Prevention	27,000	28,500	74,600	112,300
Appraisal	155,000	170,300	113,400	107,000
Internal Failure	386,400	469,200	347,800	219,100
External Failure	242,000	192,000	103,500	106,000
Total	**$ 810,400**	**$ 860,000**	**$ 639,300**	**$ 544,400**
Sales	**$ 4,360,000**	**$ 4,450,000**	**$ 5,050,000**	**$ 5,190,000**

Key points of study

- Approximately _____ of the H&S's total quality costs are a result of_____.
- In 2000 company spent more money on _____ that resulted into_____.
- With this strategy, H&S was able to identify more defective items, resulting _____.
- In year _____ & _____ company spent more money on prevention activities i.e. training of employees, redesigning the production process and planning how to build in product quality etc.
- Prevention costs increased by more than _____ during the 4 year period resulted into decrease in overall quality costs.

The H&S company also desired to develop index numbers using quality costs as a proportion of sales.

Quality index no. for 1999 sales is:

= (810,400/4,360,000)*100 = 18.58 and similarly for other years:

Year	Quality sales index
1999	18.58
2000	
2001	
2002	

Conclusion:

When the cost of achieving good quality increases; cost of poor quality decreases automatically.

Benefits of Using Quality Costs

- Quantify the size of the quality problem
- Identify major opportunities for cost reductions
- It helps in Identification of opportunities for reducing customer dissatisfaction and associated threats to product salability
- Measures the results of quality improvement activities
- Align quality goals with organizational goals
- Set cost reduction targets

The Uniqueness of SQA

- No developer will declare that its software is free of defects (major manufacturers of computer hardware won't claim this)
- This refusal actually reflects the essential elemental differences between software and other industrial products, such as automobiles, washing machines, radios etc.
- The differences of these two industrial can be categorized in the following areas:
  - Product complexity
  - Product visibility
  - Product development and production process

Definition – Software Defect

- Software Defect: any flaw or imperfection in a software work product or software process
  - Software work product is any artifact as part of the software process
  - Software process is a set of activities, methods, practice, and transformation that people use to develop and maintain software work product
  - A defect is frequently referred to as “fault” or “bug”

Note: Focus on predicting those defects that affect Project Performance and Product PErformance

Factors Affecting Defect Detection (Software Products vs. Other Industrial Products)

- The fundamental differences between the development and production processes of Software Products and other industrial products warrant the creation of a different SQA methodology for software (Chapter 2)
- Hence, there is a need for special tools, methods, procedures, strategies for the software industry

Factors Affecting Defect Detection (Software Products vs. Other Industrial Products)

Characteristic	Software Products	Other Industrial Products
Complexity	Usually, very complex product allowing for very large number of operational options	Degree of complexity much lower, allowing at most a few thousand operational options
Visibility of product	Invisible product, impossible to detect defects or omissions by sight (e.g. of a diskettee or CD storing the software)	Visible product, allowing effective detection of defects by sight
Nature of development and production process	Opportunities to detect defects arise in only one phase, namely product development	Opportunities to detect defects arise in all phases of development and production: • Product development • Product production planning • Manufacturing

Detection of Defects for Software Products

- In comparison to industrial products, software products DO NOT benefit from the opportunities for detection of defects at all three phases of the production process.
- The only phases when defects can be detected is the development phase.
- The uniqueness of the software development process compare to other industrial products:
  - High complexity
  - Invisibility of the product
  - Opportunities to detect defects (“bugs”) are limited to the product development phase
- The great complexity as well as invisibility of software, make the development of SQA methodology and its successful implementation a highly professional challenge

The Model for Software Defects Removal

- The model itself is to be integrated within a project's development process
- The model deals with two quantitative aspects of an SQA plan consisting of several defect detection activities:
  A. The plan's total effectiveness in removing project defects
  B. The total costs of removal of project defects

The Model of SQA Defect Removal – The Data

- The model is based on three types of data:
  - Defect origin distribution
  - Defect removal effectiveness
  - Cost of defect removal

The Model of SQA Defect Removal – Defect Origin Distribution

- Defect origins (the phase in which defects were introduced) are distributed throughout the development process, from the project's initiation to its completion.
- Survey conducted by major software developers (e.g. IBM, TRW etc.), reveal a similar pattern of defect distribution. Software development professionals believe that this pattern has not changed substantially in the last two decades.

Software Development Phase	Average % of defects originating in phase
Requirement Specification	15%
Design	35%
Coding (Coding 30%, Integration 10%)	40%
Documentation	10%

- **Sources: Boehm (1981) and Jones (1996)**

The Model of SQA Defect Removal – Defect Removal Effectiveness

- It is assumed that any QA activity filters (screens) a certain percentage of existing defects.
- This is because in most cases, the percentage of removed defects is somewhat lower than the percentage of detected defects as some corrections (about 10%) are ineffective or inadequate. The remaining defects, those undetected and uncorrected, are passed to successive development phases. The next QA activity applied confronts a combination of defects: those remaining after previous QA activities together with "new" defects, created in the current development phase.

The Model of SQA Defect Removal – Defect Removal Effectiveness

- Typical average defect filtering effectiveness rates for the various QA activities, by development phase are:

QA Activities	Average defect filtering effectiveness rate
Requirement Specification Review	50%
Design Inspection	60%
Design Review	50%
Code Inspection	65%
Unit Test	50%
Unit Test after code inspection	30%
Integration Test	50%
System Test / Acceptance Tests	50%
Documentation Review	50%

- **Sources: Boehm (1981) and Jones (1996)**

The Model of SQA Defect Removal – Cost of Defect Removal

- Data collected about development project costs show that the cost of removal of detected defects varies by development phase; while cost rise substantially as the development process proceeds.
- The representative average relative defect-removal costs are:

Defect Removal Phase	Average relative defect removal cost [cost units in Working Day] Defect Origination Phase				
	Req	Des	Uni	Int	Doc
Requirements Specification (Req)	1	-	-	-	-
Design (Des)	2.5	1 *	-	-	-
Unit Coding (Uni)	6.5	2.6	1	-	-
Integration (Int)	16	6.4	2.5	1	-
System Documentation (Doc)	-	-	-	-	1
System Testing / Acceptance Testing (Sys)	40	16 *	6.2	2.5	2.5
Operation by Customer (after release)	110	44	17	6.9	6.9

* Removal of a design defect detected in the design phase require an investment of ONE working day:
* Removal of a the same design defect in the acceptance testing require SIXTEEN (16) working day

Techniques to Improve Software Quality

- Software-quality objectives
- Explicit quality-assurance activity
- Testing strategy
- Software-engineering guidelines
- Informal technical reviews
- Formal technical reviews
- External audits
- Setting Objectives
- Improved Development Process
  - Change-control procedures
  - Measurement of results
  - Prototyping

Techniques to Improve Software Quality

- Setting Objectives
  - Programmer have high achievement motivation: they will work to the objectives specified, but they must be told what the objectives are
  - Objectives conflict and it's generally not possible to do well on all of them

Objectives Team Was Told to Optimize	Minimum memory use	Most readable output	Most readable code	Least code	Minimum programming time
Minimum memory	**1**	**4**	**4**	**2**	**5**
Output readability	**5**	**1**	**1**	**5**	**3**
Program readability	**3**	**2**	**2**	**3**	**4**
Least code	**2**	**5**	**3**	**1**	**3**
Minimum programming time	**4**	**3**	**5**	**4**	**1**

Relative Effectiveness

Removal Step	Lowest Rate	Modal Rate	Highest Rate
Informal design reviews	25%	35%	40%
Formal design inspections	45%	55%	65%
Informal code reviews	20%	25%	35%
Formal code inspections	45%	60%	70%
Modeling or prototyping	35%	65%	80%
Personal desk-checking of code	20%	40%	60%
Unit test	15%	30%	50%
New function (component) test	20%	30%	35%
Integration test	25%	35%	40%
Regression test	15%	25%	30%
System test	25%	40%	55%
Low-volume beta test (<10 sites)	25%	35%	40%
High-volume beta test (>1,000 sites)	60%	75%	85%

Higher Defect Detection Rate

Removal Stepc	Lowest Rate	Modal Rate	Highest Rate
Informal design reviews (pair programming)	25%	35%	40%
Informal code reviews (pair programming)	20%	25%	35%
Personal desk-checking of code	20%	40%	60%
Unit test	15%	30%	50%
Integration test	25%	35%	40%
Regression test	15%	25%	30%
Expected cumulative defect-removal efficiency	~74%	~90%	~97%

Recommendations

- Formal inspections of all requirements, all architecture, and designs for critical parts of a system
- Modeling or prototyping
- Code reading or inspections
- Execution testing
- Combination of Techniques
- Extreme programing Techniques

The General Principles

- The General Principle of Software Quality is that improving quality reduces development costs
  - The industry-average productivity for a software product is about 10 to 50 of lines of delivered code per person per day
  - Debugging and associated refactoring and other rework consume about 50 percent of the time on a traditional, naive software-development cycle

A Review of Defect Prediction Techniques

- Why Analyse and Predict Defects?
  - Project Management
    - Assess project progress
    - Plan defect detection activities
  - Work Product Assessment
    - Decide work product quality
  - Process Management
    - Assess process performance
    - Improve capability

Defects as the Focus of Prediction

- Distinguish between “major” and “minor” defects
  - Do not use minor or documentation defects in predictions
  - Minor defects will inflate estimate of latent product defects
- Most defect prediction techniques used in planning reply on historical data
- Defect prediction techniques vary in the types of data they require:
  - Some require little data, others require more
  - Some use work product characteristics, other require defect data only
- Techniques have strengths and weaknesses depending on the quality f the input used for prediction

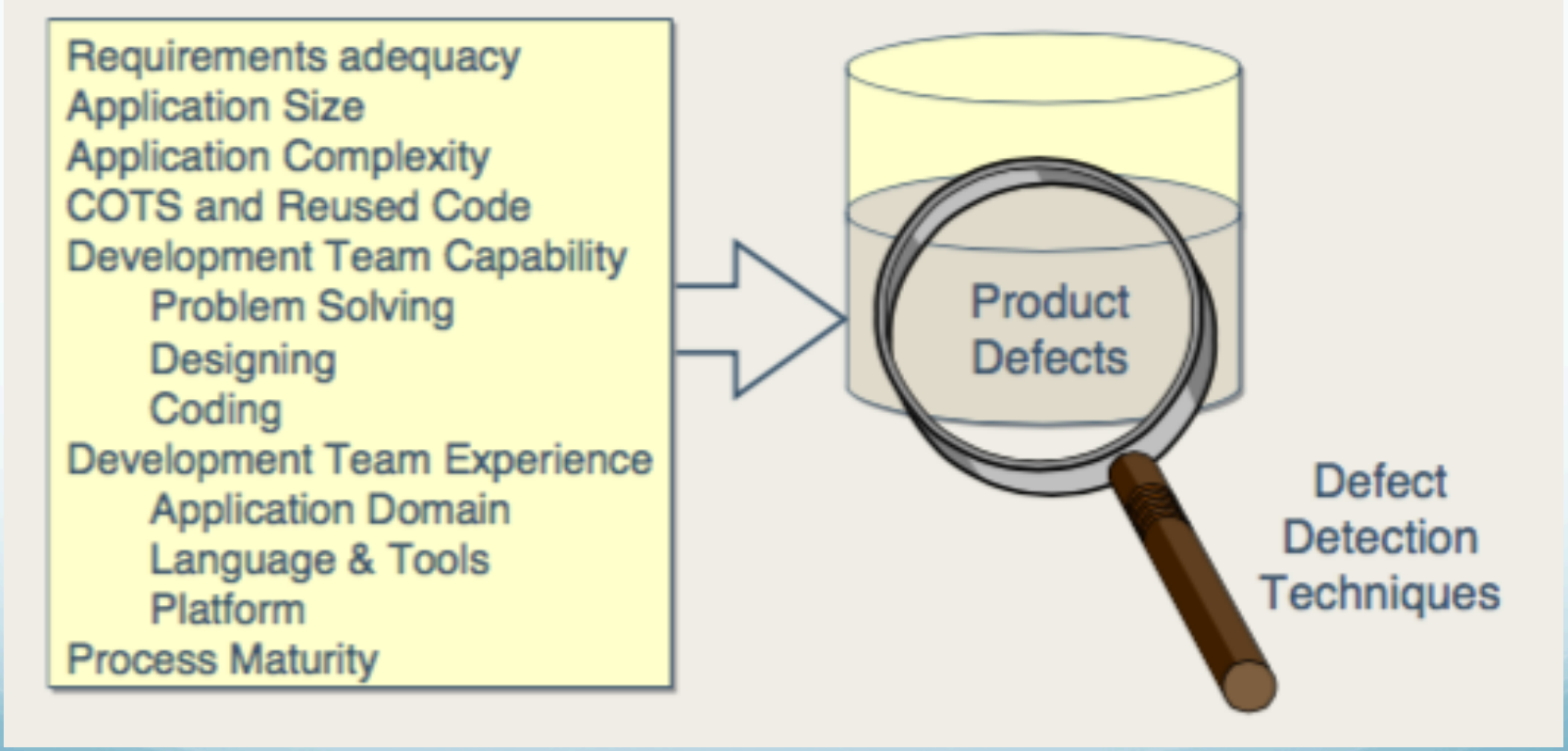
Defect Attributes Available for Analysis

- **Problem Status**
  - Open
    - Recognized
    - Evaluate
    - Resolved
  - Closed
- **Problem Type**
  - Software Defect
    - Requirement Defect
    - Design Defect
    - Code Defect
    - Operational Doc. Defect
    - Test Case Defect
    - Other Product Defect
- **Problem Type**
  - Other Problems
    - Hardware Problem
    - OS Problem
    - User Mistake
    - Operations Mistake
    - New Req / Enhancement
  - Undermined
    - Not repeatable
    - Value not identified
- **Uniqueness**
  - Original
  - Duplicate
- **Critical Level**
- **Urgency**

Additional Attributes to Consider

- Recognition
  - What is the problem?
  - When was the problem reported?
  - Who reported the problem?
- Evaluation
  - What work product caused the problem?
  - What activity discovered the problem?
  - What activity introduced the problem?
- Resolution
  - What work needs to be done?
  - What work products will be affected by the change?
  - What are the prerequisite changes?
- Closure
  - When are the changes expected?
  - What configuration contains the changes?

Project and Process Factors Correlated with Defect Insertion



Defect Prediction Techniques

- Project Management
  - Empirical Defect Prediction
  - Defect Discovery Profile
  - COQUALMO
  - Orthogonal Defect Classification
- Work Product Assessment
  - Fault Proneness Evaluation (Size, Complexity, Prior History)
  - Capture/Recapture Analysis
- Process Management
  - Defect Prevention Program
  - Statistical Process Control