投票 最多可选1项

设置

Are you ready ?

- A Yes
- B No

提交

Software Engineering

Part 3
Quality Management

Chapter 21
Software Quality Assurance

These slides are designed to accompany *Software Engineering: A Practitioner's Approach, 8/e* (McGraw-Hill 2014). Slides copyright 2014 by Roger Pressman.

Reproduced by Ning Li , 2022

Contents

- 21.1 Background Issues
- 21.2 Elements of Software Quality Assurance
- 21.3 SQA Processes and Product Characteristics
- 21.4 SQA Tasks, Goals, and Metrics
- 21.5 Formal Approaches to SQA
- 21.6 Statistical Software Quality Assurance
- 21.7 Software Reliability
- 21.8 The ISO 9000 Quality Standards
- 21.9 The SQA Plan
- 21.10 A Framework for Product Metrics

21.1 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.)
  - *Most people feel that problems in these areas are caused by other people.* (If only they would take the time to do things right.)

21.2 Elements of SQA

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

DEFECT SEVERITY AND PRIORITY IN TESTING WITH EXAMPLES AND DIFFERENCE

21.2 Methods of SQA

- Defect Prevention

- Defect Removal
  - inspection: faults discovered/removed
  - testing: failures trace back to faults

- Defect Tolerance
  - local failure != global failure

- Recover

https://softwarequalityeng.blogspot.com/2018/03/defect-prevention.html

21.2 Methods of SQA

- Defect Prevention

□ Error blocking

. error: missing/incorrect actions

. direct intervention to block errors

. rely on technology/tools/etc.

□ Error source removal

. root cause analysis

-> identify error sources

. removal through education/training/etc.

□Systematic defect prevention via process improvement.

21.2 Methods of SQA

- Defect Removal: Inspection
  - □ Artifacts (code/design/test-cases/etc.) from req./design/coding/testing/etc. phases
  - □ Informal reviews
    - self conducted reviews.
    - independent reviews.
    - orthogonality of views desirable.
  - □ Formal inspections
    - Inspection and variations.
    - Process and structure.
    - Individual vs. group inspections.

Review:
meeting review
github online review

video source: https://www.youtube.com/watch?v=ocMraYgqHvg
https://www.youtube.com/watch?v=8fx-EaOUK2E

21.2 Methods of SQA

- Defect Removal: Testing
  - □ Product/Process characteristics:
    - •object: product type, language, etc.
    - •scale/order
    - •unit, component, system,
    - •who: self, independent, 3rd party
  - □ What to check
    - validation vs. verification .

      validation: do right things

      verification: do things right
    - external specifications (black-box).
    - internal implementation (white/clear-box).
  - □ Criteria: when to stop
    - coverage of specs/structures.
    - reliability -> usage-based testing.

21.2 Methods of SQA

- Fault Tolerance
  - □ Motivation:
    - fault present but
      - removal infeasible/impractical
    - fault tolerance ->contain defects
  - □ FT techniques: break fault-failure link
    - recovery: rollback and redo.
    - NVP: N-version programming.
    - voted

21.3 Role of the SQA Group-I

Give us the evidence which shows that you managed this project well
?????

- **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.

21.3 Role of the SQA Group-II

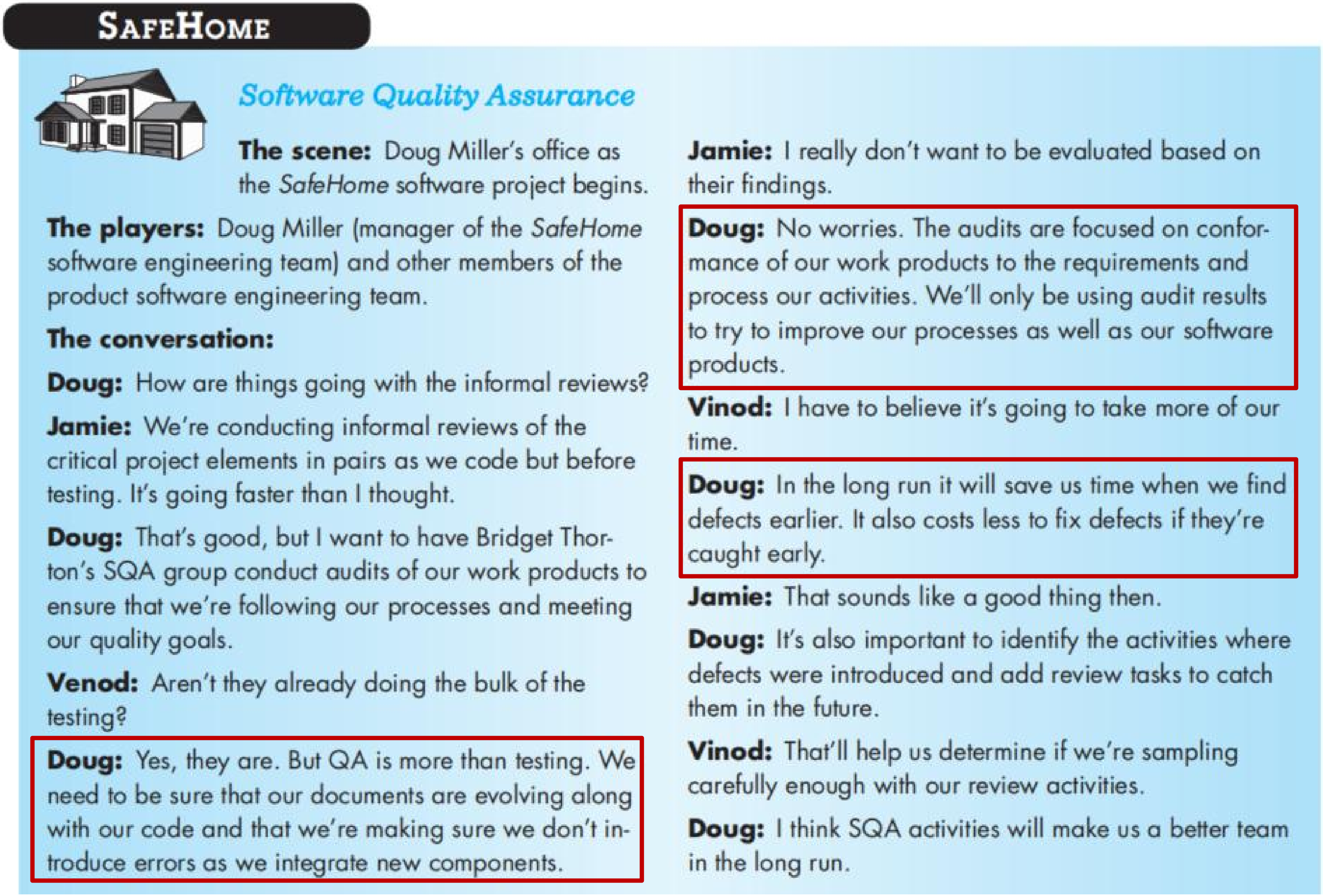
- Reviews software engineering activities to verify compliance with the defined software process.
  - identifies, documents, and tracks deviations from the process and verifies that corrections have been made.

- 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.

- Records any noncompliance and reports to senior management.
  - Noncompliance items are tracked until they are resolved.

21.3 Role of the SQA Group-II



SafeHome

Software Quality Assurance

The scene: Doug Miller's office as the *SafeHome* software project begins.

The players: Doug Miller (manager of the *SafeHome* software engineering team) and other members of the product software engineering team.

The conversation:

Doug: How are things going with the informal reviews?

Jamie: We're conducting informal reviews of the critical project elements in pairs as we code but before testing. It's going faster than I thought.

Doug: That's good, but I want to have Bridget Thorton's SQA group conduct audits of our work products to ensure that we're following our processes and meeting our quality goals.

Venod: Aren't they already doing the bulk of the testing?

Doug: Yes, they are. But QA is more than testing. We need to be sure that our documents are evolving along with our code and that we're making sure we don't introduce errors as we integrate new components.

Jamie: I really don't want to be evaluated based on their findings.

Doug: No worries. The audits are focused on conformance of our work products to the requirements and process our activities. We'll only be using audit results to try to improve our processes as well as our software products.

Vinod: I have to believe it's going to take more of our time.

Doug: In the long run it will save us time when we find defects earlier. It also costs less to fix defects if they're caught early.

Jamie: That sounds like a good thing then.

Doug: It's also important to identify the activities where defects were introduced and add review tasks to catch them in the future.

Vinod: That'll help us determine if we're sampling carefully enough with our review activities.

Doug: I think SQA activities will make us a better team in the long run.

21.4 SQA Goals

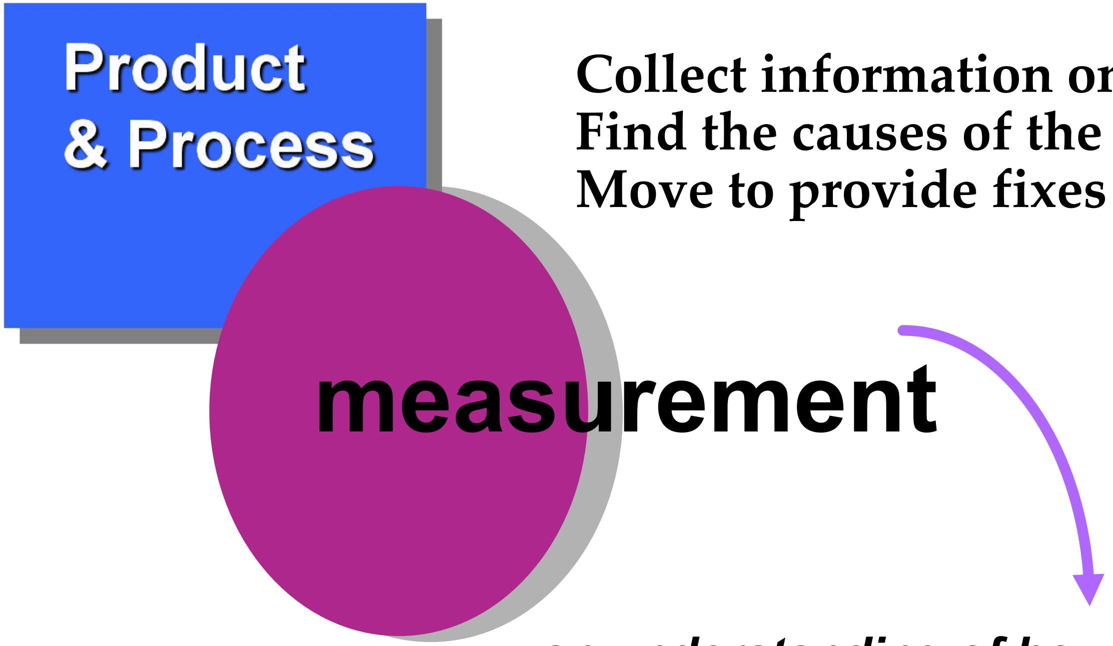
- **Requirements quality.** The correctness, completeness, and consistency of the requirements model will have a strong influence on the quality of all work products that follow.
- **Design quality.** Every element of the design model should be assessed by the software team to ensure that it exhibits 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.

21.5 Formal SQA

- Assumes that a rigorous syntax and semantics can be defined for every programming language
- Allows the use of a rigorous approach to the specification of software requirements
- Applies mathematical proof of correctness techniques to demonstrate that a program conforms to its specification

21.6 Statistical SQA

Collect information on all defects
Find the causes of the defects
Move to provide fixes for the process

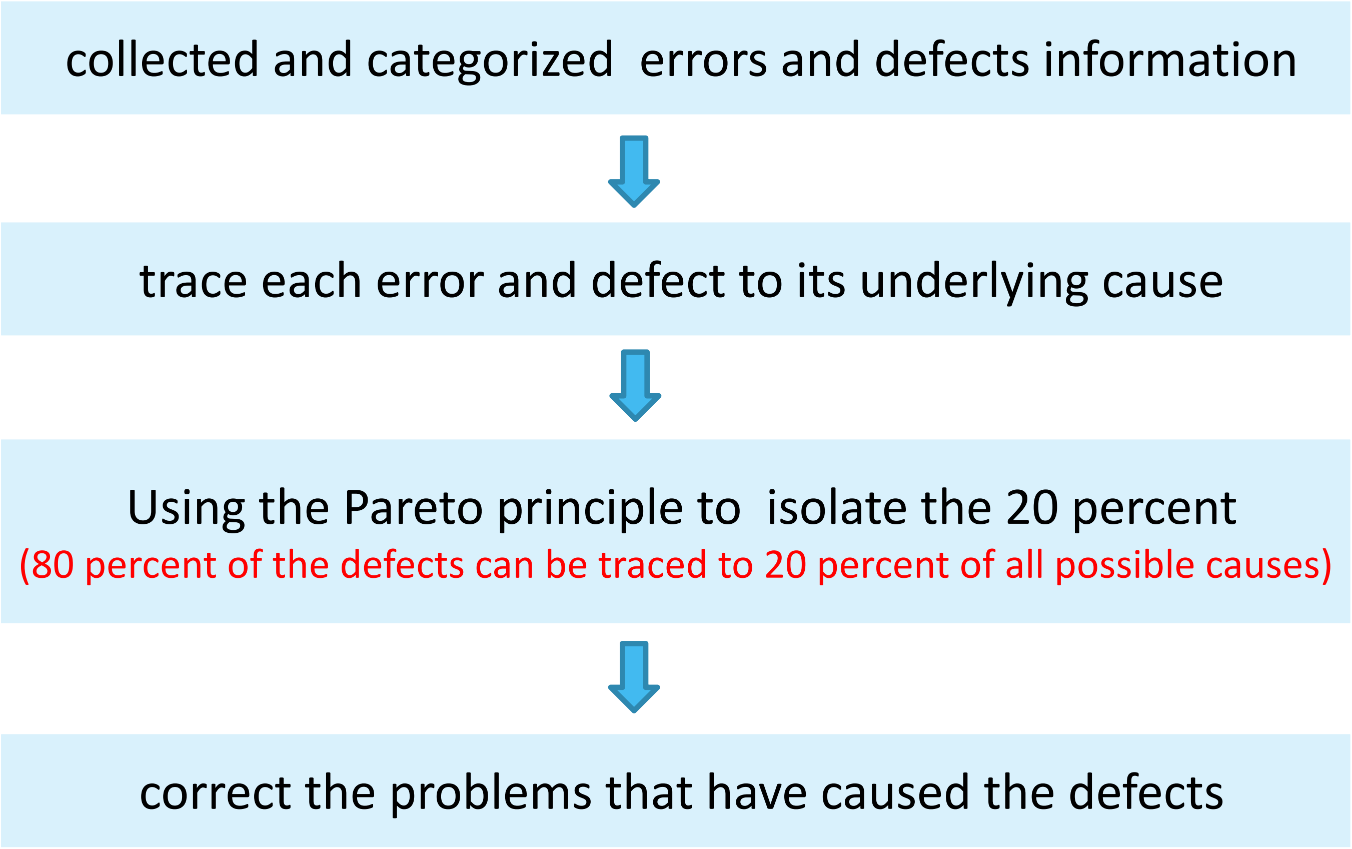


Product & Process

measurement

... an understanding of how to improve quality ...

21.6 Statistical SQA



collected and categorized errors and defects information

trace each error and defect to its underlying cause

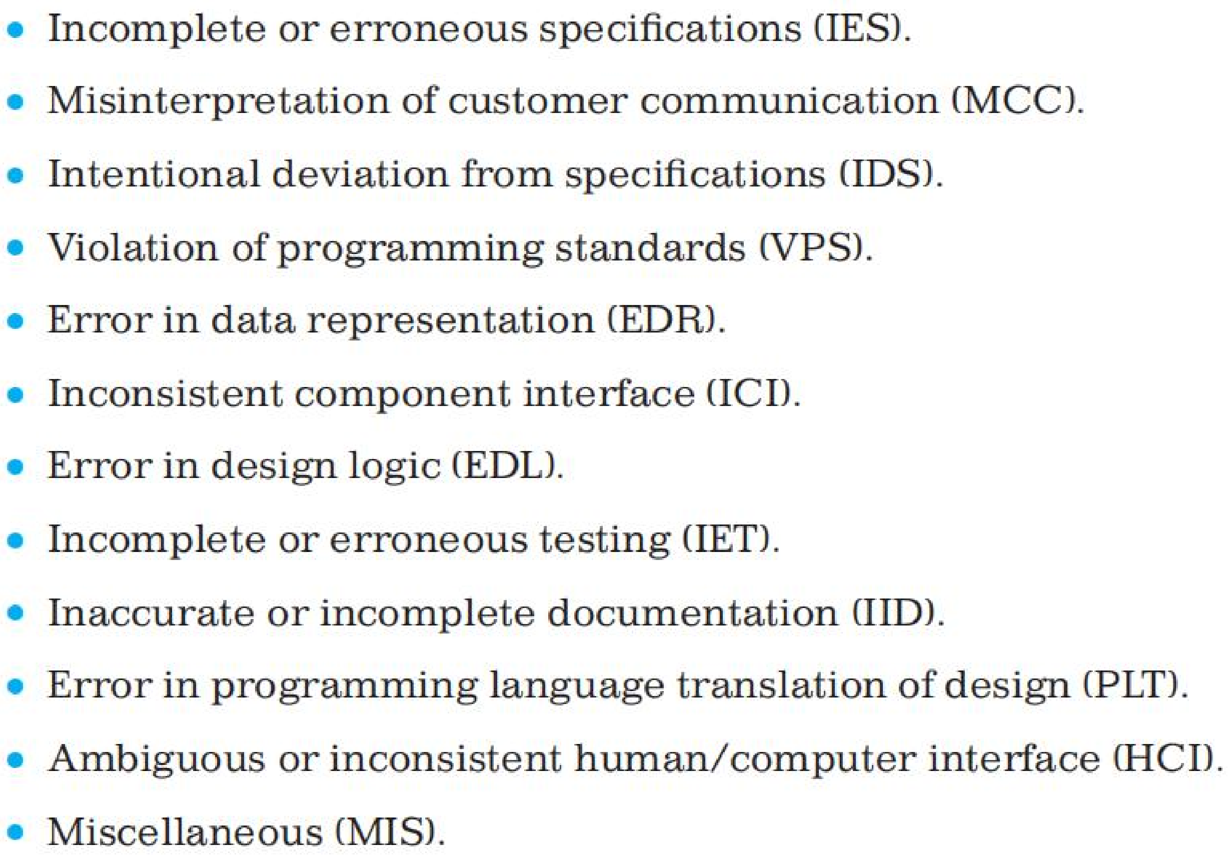
Using the Pareto principle to isolate the 20 percent

(80 percent of the defects can be traced to 20 percent of all possible causes)

correct the problems that have caused the defects

21.6 Statistical SQA

Although hundreds of different problems are uncovered, all can be tracked to one (or more) of the following causes

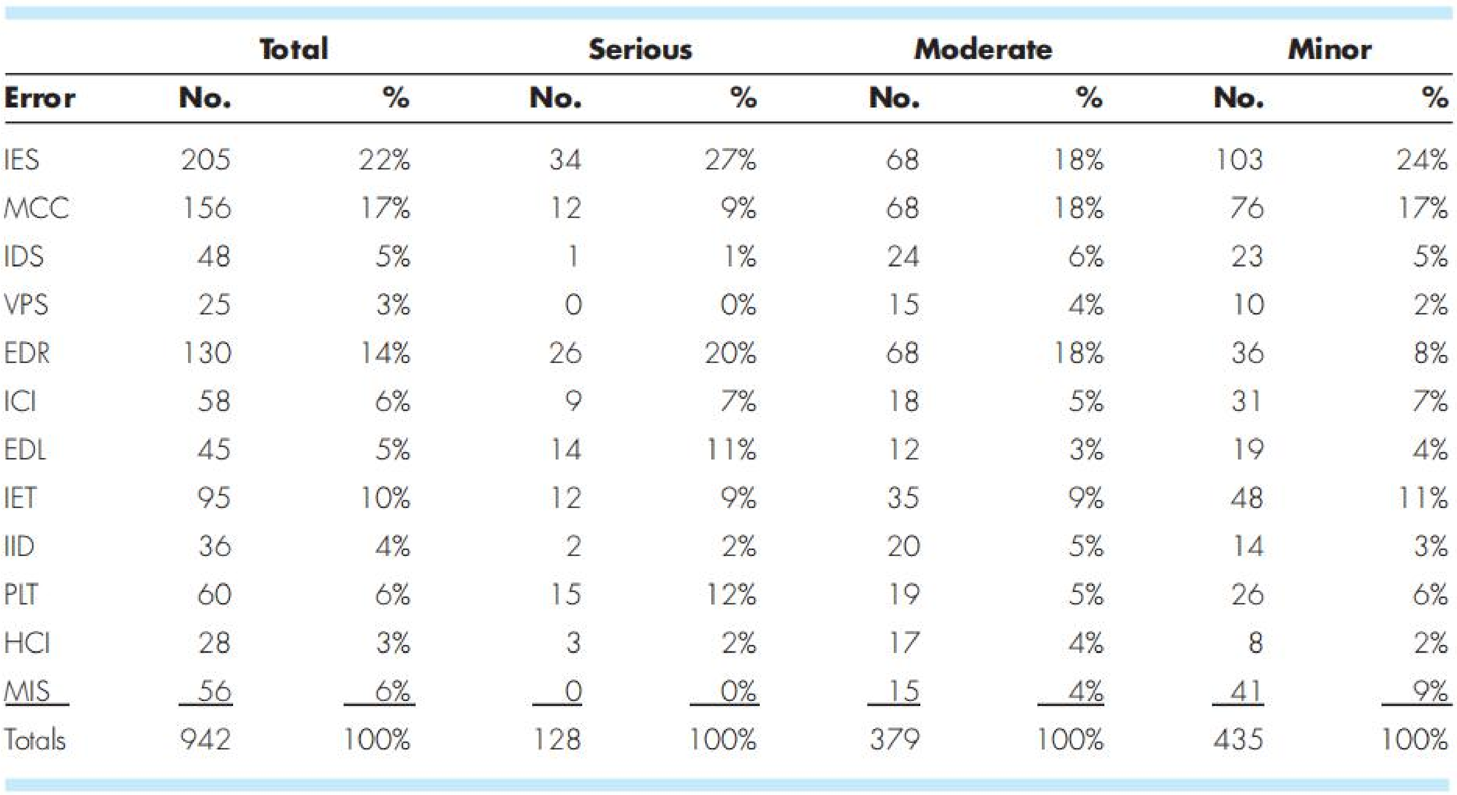


- Incomplete or erroneous specifications (IES).
- Misinterpretation of customer communication (MCC).
- Intentional deviation from specifications (IDS).
- Violation of programming standards (VPS).
- Error in data representation (EDR).
- Inconsistent component interface (ICI).
- Error in design logic (EDL).
- Incomplete or erroneous testing (IET).
- Inaccurate or incomplete documentation (IID).
- Error in programming language translation of design (PLT).
- Ambiguous or inconsistent human/computer interface (HCI).
- Miscellaneous (MIS).

21.6 Statistical SQA

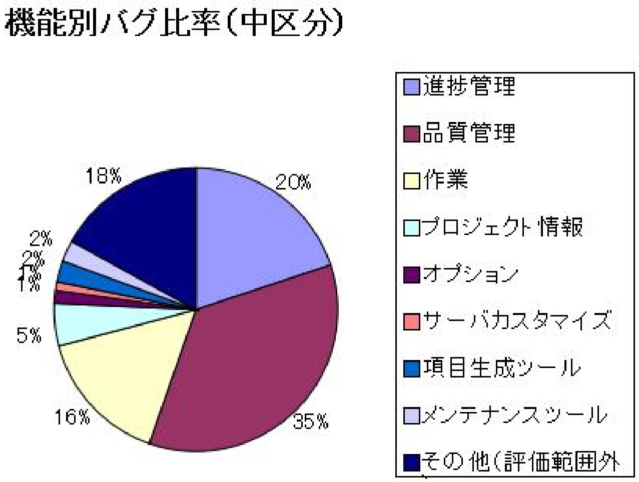
FIGURE 21.2

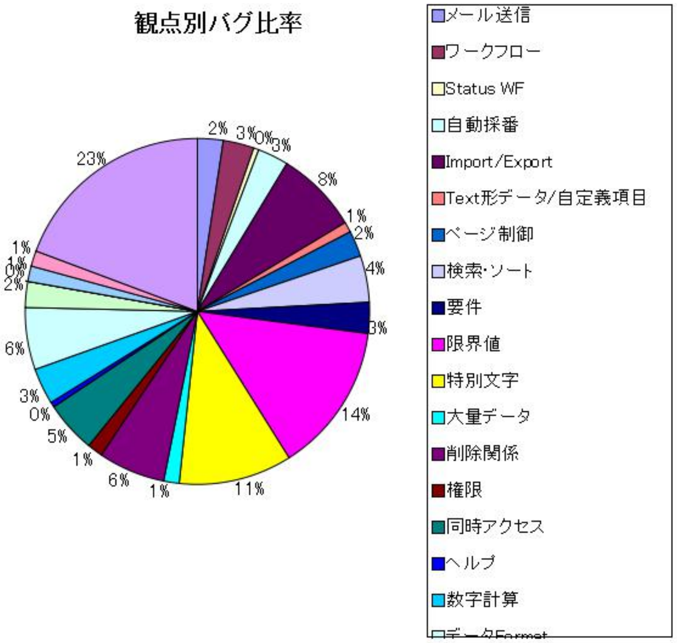
Data collection for statistical SQA



	Total		Serious		Moderate		Minor	
Error	**No.**	**%**	**No.**	**%**	**No.**	**%**	**No.**	**%**
IES	205	22%	34	27%	68	18%	103	24%
MCC	156	17%	12	9%	68	18%	76	17%
IDS	48	5%	1	1%	24	6%	23	5%
VPS	25	3%	0	0%	15	4%	10	2%
EDR	130	14%	26	20%	68	18%	36	8%
ICI	58	6%	9	7%	18	5%	31	7%
EDL	45	5%	14	11%	12	3%	19	4%
IET	95	10%	12	9%	35	9%	48	11%
IID	36	4%	2	2%	20	5%	14	3%
PLT	60	6%	15	12%	19	5%	26	6%
HCI	28	3%	3	2%	17	4%	8	2%
MIS	56	6%	0	0%	15	4%	41	9%
Totals	942	100%	128	100%	379	100%	435	100%

21.6 Statistical SQA

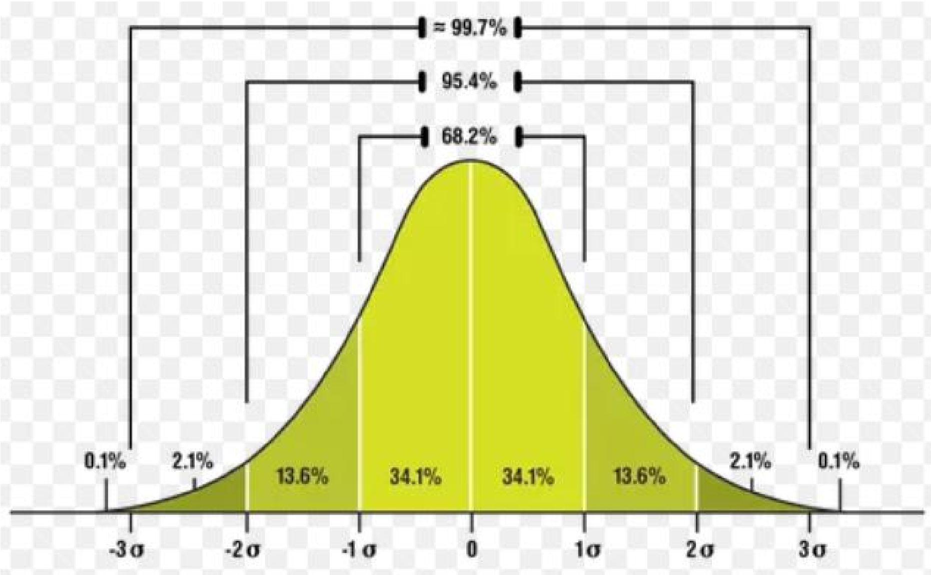




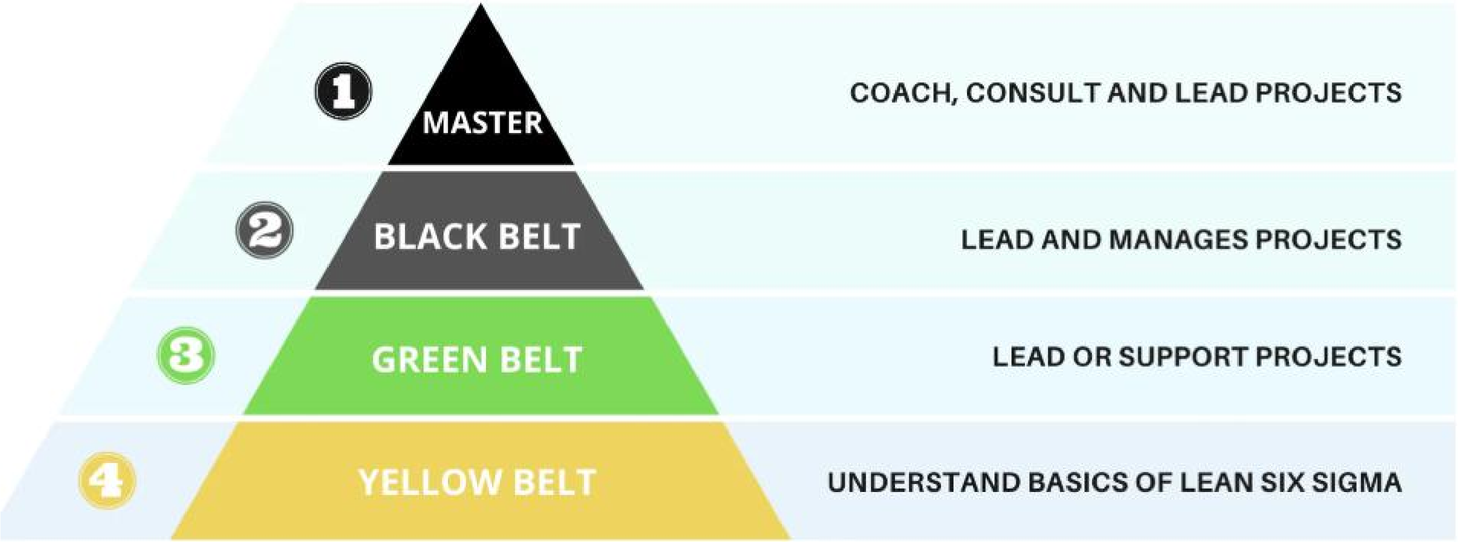
21.6 Statistical SQA

Six Sigma for Software Engineering：DMAIC

- ✓ define
- ✓ measure
- ✓ analyze
- ✓ improve
- ✓ control



21.6 Statistical SQA


1
MASTER
COACH, CONSULT AND LEAD PROJECTS
2
BLACK BELT
LEAD AND MANAGES PROJECTS
3
GREEN BELT
LEAD OR SUPPORT PROJECTS
4
YELLOW BELT
UNDERSTAND BASICS OF LEAN SIX SIGMA

- **Yellow Belt**: Serves as a basic introduction to Lean Six Sigma for those new to the domain.
- **Green Belt**: Intermediate program that prepares you to work on process improvement projects within a company.
- **Black Belt**: Advanced program that prepares you to manage and lead project teams.
- **Master Black Belt**: Prestigious program that prepares you to educate others and become a master in the domain.

https://www.6sigmacertificationonline.com/six-sigma-belts/

21.6 Statistical SQA

σ	% Good	DPMO	spelling	Time
1	30.9%	690000	170 misspelled words per page in a book	31.75 years per century
2	69.2%	308000	25 misspelled words per page in a book	4.5 years per century
3	93.3%	66800	1.5 misspelled words per page in a book	3.5 months per century
4	99.4%	6210	1 misspelled words per 30 pages in a book	2.5 Days per century
5	99.98%	230	1 misspelled words in a set of encyclopaedia	30 Minutes per century
6	**99.9997%**	**3.4**	**1 misspelled words in all the books in a small library**	**6 Seconds per century**

21.7 Software Reliability

Reliabilty: The probability of failure-free operation of a computer program in a specified environment for a specified time.[Musa87]

- Mean time to failure (MTTF)
- Mean time to repair (MTTR)
- Mean time between failures (MTBF)

  MTBF = MTTF + MTTR
- **Reliability R = MTTF/(1+MTTF)**
- Availability A = MTBF/(1+MTBF)
- Maintainability M = 1/(1+MTTR)

21.7 Software Reliability

- Example: failure time (day [execution time...])
  - SF1: 180, 675, 315, 212, 278, 503, 431
  - SF2: 477, 1048, 685, 396
  - SF3: 894, 1422

  - $MTTF_{SF1}$ = 2594/7 = 370.57 R=370.57/(1+370.57)=0.997
  - $MTTF_{SF2}$ = 2606/4 = 651.5 R=651.5 /(1+651.5) =0.998
  - $MTTF_{SF3}$ = 2316/2 = 1158 R=1158/(1+1158)=0.999

21.7 Software Safety

- *Software safety* is a software quality assurance activity that focuses on the identification and assessment of potential hazards that may affect software negatively and cause an entire system to fail.

- If hazards can be identified early in the software process, software design features can be specified that will either eliminate or control potential hazards.

21.8 ISO 9001:2008 Standard

- ISO 9001:2008 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 9001:2008 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.

Summary

- **Methods of SQA**
  - ✓ Defect prevention: Error blocking, Error source removal, process improvement
  - ✓ Defect removal: inspection and review, testing
  - ✓ Defect tolerance: NVP, out-voted,
  - ✓ recovery: recovery from failure
- **SQA Group**
  - ✓ roles: process review, work products review, audit, record and report.
  - ✓ goals: Quality of requirements, design, code; Quality control effectiveness
  - ✓ methods: review, statistical analysis
- Software Reliability: MTTF, R = MTTF/(1+MTTF)

Thinking (Ch21)

1. Can a program be correct and still not be reliable? Explain.
2. You have been given the responsibility for improving the quality of software across your organization. What is the first thing that you should do? What’s next?
3. Consider two safety-critical systems that are controlled by computer. List at least three hazards for each that can be directly linked to software failures.

Thinking (Answer)

1. Can a program be correct and still not be reliable? Explain.

   Yes. It is possible for a program to conform to all explicit functional and performance requirements at a given instant, yet have errors that cause degradation that ultimately causes the program to fail.

2. You have been given the responsibility for improving the quality of software across your organization. What is the fifi rst thing that you should do? What's next?

   Institute formal technical reviews. After these are working smoothly, any of a number of SQA activities might be implemented: change control and SCM; comprehensive testing methodology; SQA audits of documentation and related software. Also software metrics can help.

Thinking (Answer)

3. Consider two safety-critical systems that are controlled by computer. List at least three hazards for each that can be directly linked to software failures.

Classic examples include aircraft avionics systems, control systems for nuclear power plants, software contained in sophisticated medical instrumentation (e.g., CAT scanners or MRI devices) control systems for trains or subway systems; elevator control systems

THE END