Software Quality

CHAPTER 2

What is software quality?

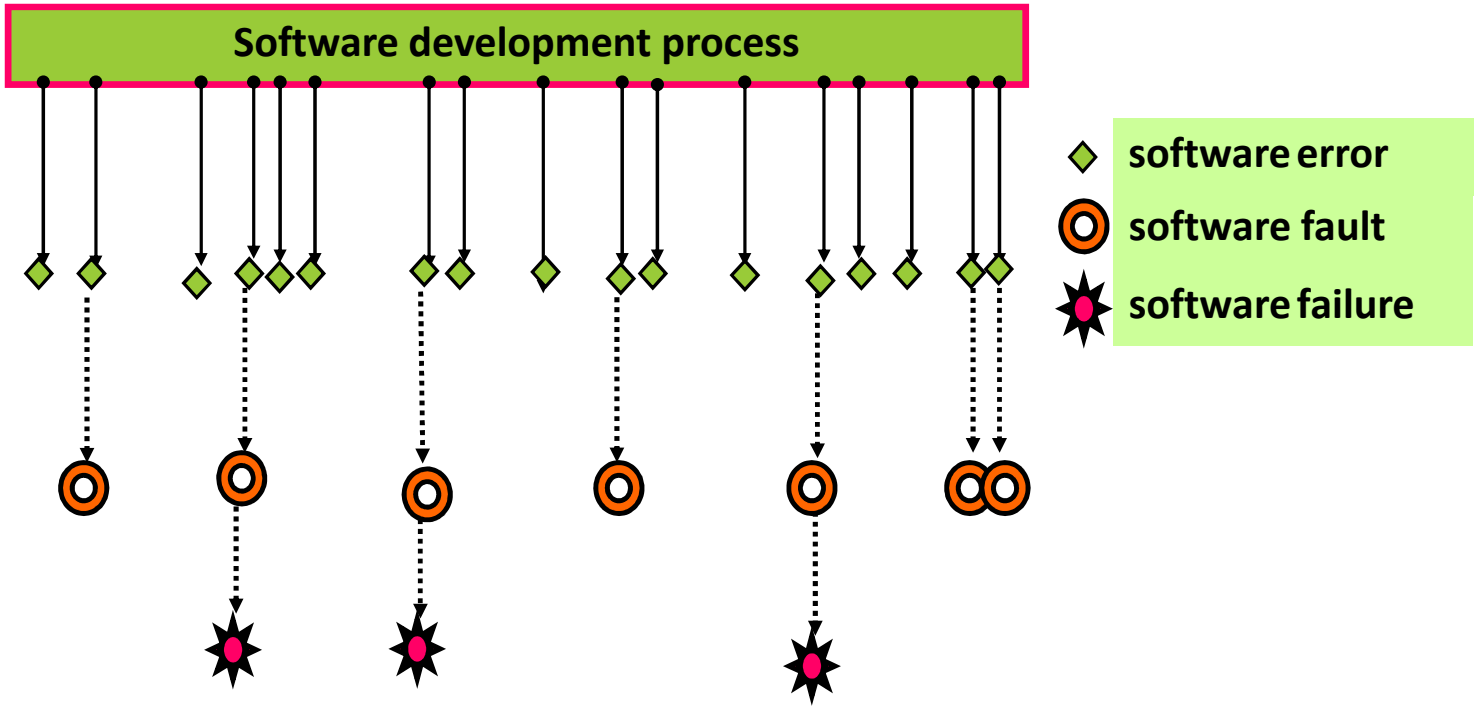
- What is software?
- Software errors, faults and failures
- Classification of the causes of software errors
- Software quality – definition
- Software quality assurance – definition and objectives
- Software quality assurance and software engineering

Software - IEEE definition

Software is:

Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.

Software errors, software faults and software failures



Software errors, Software faults and Software failures.

- **Software errors** are sections of the code that are partially or totally incorrect as a result of a grammatical, logical or other mistake made by a systems analyst, a programmer, or another member of the software development team.

- **Software faults** are software errors that cause the incorrect functioning of the software during a specific application.

- **Software faults** become **software failures** only when they are “activated”, that is, when a user tries to apply the specific software section that is faulty. Thus, the root of any software failure is a software error

The nine causes of software errors are:

1. Faulty requirements definition
2. Client-developer communication failures
3. Deliberate deviations from software requirements
4. Logical design errors
5. Coding errors
6. Non-compliance with documentation and coding instructions
7. Shortcomings of the testing process
8. User interface and procedure errors
9. Documentation errors

Software quality - IEEE definition

Software quality is:

(1) The degree to which a system, component, or process meets specified requirements.

(2) The degree to which a system, component, or process meets customer or user needs or expectations.

Software quality - Pressman's definition

Software quality is :

Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.

SQA - IEEE definition

Software quality assurance is:

1. **A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.**
2. **A set of activities designed to evaluate the process by which the products are developed or manufactured. Contrast with: quality control.**

SQA - expanded definition

Software quality assurance is:

A systematic, planned set of actions necessary to provide adequate confidence that the software development process or the maintenance process of a software system product conforms to established functional technical requirements as well as with the managerial requirements of keeping the schedule and operating within the budgetary confines.

The objectives of SQA activities in software development

(1) Assuring an acceptable level of confidence that the software will conform to functional technical requirements.

(2) Assuring an acceptable level of confidence that the software will conform to managerial scheduling and budgetary requirements.

(3) Initiation and management of activities for the improvement and greater efficiency of software development and SQA activities.

The objectives of SQA activities in software maintenance

(1) Assuring an acceptable level of confidence that the software maintenance activities will conform to the functional technical requirements.

(2) Assuring an acceptable level of confidence that the software maintenance activities will conform to managerial scheduling and budgetary requirements.

(3) Initiate and manage activities to improve and increase the efficiency of software maintenance and SQA activities.

The differences between software quality assurance and quality control.

- Quality Control is a set of activities carried out with the main objective of withholding products from shipment if they do not qualify

- Quality Assurance is meant to minimize the costs of quality by introducing a variety of activities throughout the development and maintenance process in order to prevent the causes of
errors, detect them, and correct them in the early stages of development. As a result, quality assurance substantially reduces the rates of non-qualifying products.

Review Questions (1)

1. A software system comprises four main components.
   (a) List the four components of a software system.
   (b) How does the quality of each component contribute to the quality of the developed software?
   (c) How does the quality of each component contribute to the quality of the software maintenance?

2. **(a)** List and briefly describe the various causes of software errors.
   (b) Classify the causes of error according to the groups responsible for the error: the client's staff, the systems analysts, the programmers, the testing staff – or is it a shared responsibility belonging to more than one group?

Review Questions (2)

3. George Wise is an exceptional programmer. Testing his software modules reveals very few errors, far fewer than the team's average. He keeps his schedule promptly, and only rarely is he late in completing his task. He always finds original ways to solve programming difficulties, and uses an original, individual version of the coding style. He dislikes preparing the required documentation, and rarely does it according to the team's templates.
   A day after completing a challenging task, on time, he was called to the office of the department's chief software engineer. Instead of being praised for his accomplishments (as he expected), he was warned by the company's chief software engineer that he would be fired unless he began to fully comply with the team's coding and documentation instructions.

   (a) Do you agree with the position taken by the department's chief software engineer?
   (b) If yes, could you suggest why his or her position was so decisive

Reference:

- Galin, SQA from Theory to Implementation @Pearson Education Limited 2004