Software Maintenance

Modified fromShari L. Pfleeger, Software Engineering Theory and Practice Second Edition, Prentice-Hall, 2001

The Changing System

- The system development is completed when used by the users in an actual production environment.
- Any work done to change the system after it is in operation is considered to be **maintenance**.
- Why the software system change?
  - A customer make a decision to do something in a different way.
  - The nature of the system itself changes.
- The more dependent a system is on the real world for its requirements, the more likely it is to change.

The System Life Span

- At some points, we may decide to replace a legacy system with a new one or retire it because no longer needed.
- System Evolution (Maintenance) vs System Decline
  - Is the cost of maintenance too high?
  - Is the system reliability unacceptable?
  - Can the system no longer adapt to further change?
  - Is the system performance still beyond the constraints?
  - Can other system do the same job better, faster or cheaper?
  - Is the cost of maintaining the hardware great enough to justify replacing it with cheaper, newer hardware?

Maintenance Activities and Roles

- Maintenance activities are similar to those of development:
  - Analyzing requirements, evaluating system and program, writing and reviewing code, testing changes, updating documentation.
- Maintenance focuses on 4 major aspects:
  - Maintaining control over the system's day-today functions.
  - Maintaining control over system modifications.
  - Perfecting existing acceptable functions.
  - Preventing system performance from degrading to unacceptable levels.

Corrective Maintenance

- To control day-to-day system function, respond to problems resulting from faults,
- As failures occur, they brought o the team's attention:
  - Resolve problems resulting from faults, find the failure's cause and make corrections.
  - Change to requirement, design, code, test suites, and documentation as necessary
- Often, the initial repair is temporary: something to keep the system running, but not the best fix
- Long-range changes may be implemented later

Adaptive Maintenance

- Sometimes a change introduced in one part of the system requires changes to other parts; **Adaptive maintenance** is the implementation of these secondary changes.
  - For example, the existing DBMS, part of a larger HW or SW system, is upgraded to a new version.
- Also, can be performed for changes in HW or environment.
  - For example, a system originally designed to work in a dry environment is chosen for use on a tank or submarine.

Perfective Maintenance

- As we maintain a system, we examine documents, design, code, and tests, looking for opportunities for improvement
  - For example, as functions are added to a system the design may become confused and difficult to follow.
- A redesign may enhance future maintenance and make it easier to add new functions in the future.
- **Perfective maintenance** involve making changes to improve some aspect of the system, even when the changes are not suggested by faults.
  - For example, documentation changes to clarify items, test suite changes to improve test coverage, code and design modifications to enhance readability

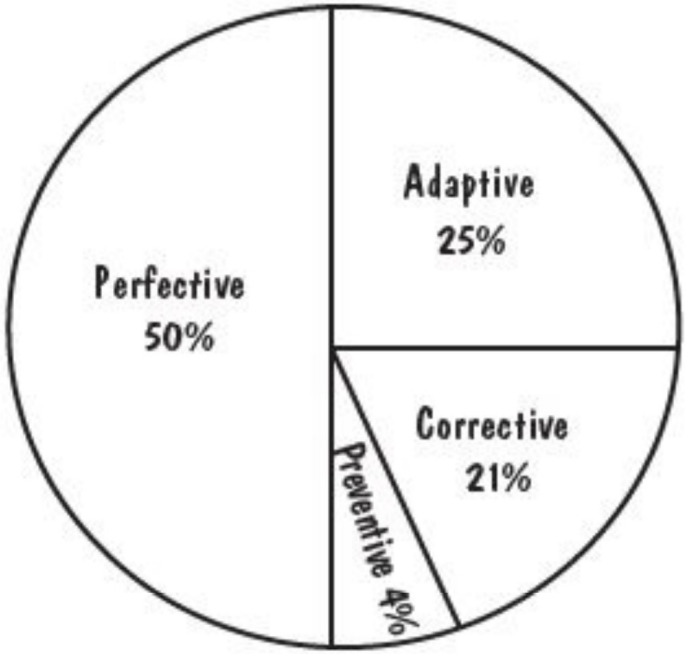
Preventive Maintenance

- **Preventive maintenance** involves changing some aspect of the system to prevent failures.
- It may include:
  - The addition of type checking
  - The enhancement of fault handling
  - The additional of a “catch-all” statement

  To make sure the system can handle all possibilities.
- Usually results when a programmer or code analyzer finds an actual or potential fault that
  - has not yet become a failure
  - takes an action to correct the fault before damage is done

Use of Maintenance Time

- Surveyed by Lientz and Swanson (1981)



Who Performs Maintenance

- The development team is not always used to maintain the system once it is operational; Often, a separate maintenance team is employed.
- There are positive and negative aspects of using a development team (using a separate team) in maintenance.
- positive:
  - The team is familiar with the system.
- negative:
  - The team feels so confident in their understanding of the system that they tend not to keep the documentation up to date.
  - Their lack of care in writing and revising documentation may result in the need for more people or resources to tackle a problem

Team Responsibilities

- Maintaining a system involves all team members:
  - users, operators, or customer representatives
  - analysts or programs: determine affected part and the impact on the design
- The maintenance team activities:
  1. understanding the system
  2. locating information in the system documentation
  3. keeping system documentation up to date
  4. extending existing functions to accommodate new or changing requirements
  5. adding new functions to the system

Team Responsibilities (cont.)

6. finding the source of system failures or problems
7. locating and correcting faults
8. answering questions about the way the system works
9. restructuring design and code components
10. rewriting design and code components
11. deleting design and code components that are no longer useful
12. managing changes to the system as they are made