

Software evolution

Software change

- Software change is inevitable
 - New requirements emerge when the software is used;
 - The business environment changes;
 - Errors must be repaired;
 - New computers and equipment is added to the system;
 - The performance or reliability of the system may have to be improved.
- A key problem for organisations is implementing and managing change to their existing software systems.

Importance of evolution

- Organisations have huge investments in their software systems - they are critical business assets.
- To maintain the value of these assets to the business, they must be changed and updated.
- The majority of the software budget in large companies is devoted to evolving existing software rather than developing new software.

Software maintenance

- Modifying a program after it has been put into use.
- Maintenance does not normally involve major changes to the system's architecture.
- Changes are implemented by modifying existing components and adding new components to the system.

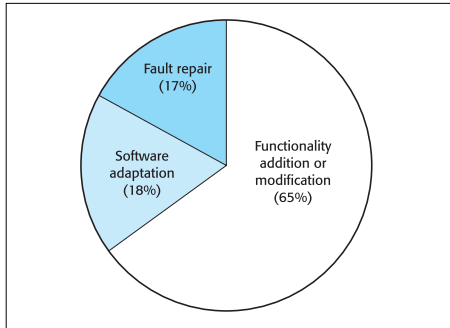
Maintenance is inevitable

- The system requirements are likely to change while the system is being developed because the environment is changing. Therefore a delivered system won't meet its requirements!
- Systems are tightly coupled with their environment. When a system is installed in an environment it changes that environment and therefore changes the system requirements.
- Systems **MUST** be maintained therefore if they are to remain useful in an environment.

Types of maintenance

- Maintenance to repair software faults
 - Changing a system to correct deficiencies in the way meets its requirements.
- Maintenance to adapt software to a different operating environment
 - Changing a system so that it operates in a different environment (computer, OS, etc.) from its initial implementation.
- Maintenance to add to or modify the system's functionality
 - Modifying the system to satisfy new requirements.

Distribution of maintenance effort



©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 21

Slide 7

Maintenance costs

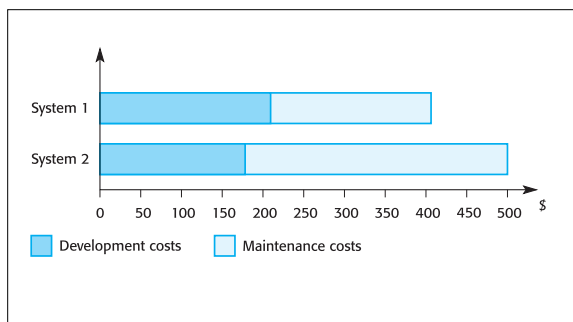
- Usually greater than development costs (2* to 100* depending on the application).
- Affected by both technical and non-technical factors.
- Increases as software is maintained. Maintenance corrupts the software structure so makes further maintenance more difficult.
- Ageing software can have high support costs (e.g. old languages, compilers etc.).

©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 21

Slide 8

Development/maintenance costs



©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 21

Slide 9

Maintenance cost factors

- Team stability
 - Maintenance costs are reduced if the same staff are involved with them for some time.
- Contractual responsibility
 - The developers of a system may have no contractual responsibility for maintenance so there is no incentive to design for future change.
- Staff skills
 - Maintenance staff are often inexperienced and have limited domain knowledge.
- Program age and structure
 - As programs age, their structure is degraded and they become harder to understand and change.

©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 21

Slide 10

Maintenance prediction

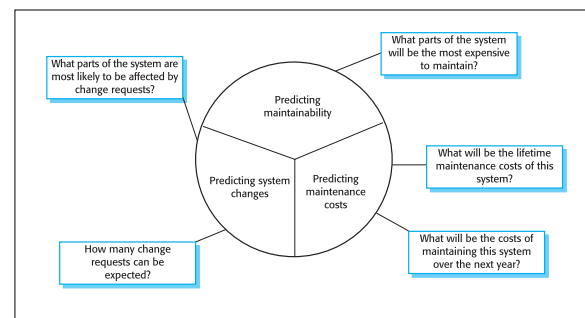
- Maintenance prediction is concerned with assessing which parts of the system may cause problems and have high maintenance costs
 - Change acceptance depends on the maintainability of the components affected by the change;
 - Implementing changes degrades the system and reduces its maintainability;
 - Maintenance costs depend on the number of changes and costs of change depend on maintainability.

©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 21

Slide 11

Maintenance prediction



©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 21

Slide 12

Change prediction

- Predicting the number of changes requires and understanding of the relationships between a system and its environment.
- Tightly coupled systems require changes whenever the environment is changed.
- Factors influencing this relationship are
 - Number and complexity of system interfaces;
 - Number of inherently volatile system requirements;
 - The business processes where the system is used.

Evolution processes

- Evolution processes depend on
 - The type of software being maintained;
 - The development processes used;
 - The skills and experience of the people involved.
- Proposals for change are the driver for system evolution. Change identification and evolution continue throughout the system lifetime.

The system evolution process

