Topic: Software Maintenance and Evolution

1. Software maintenance – the term has broader scope than is usually associated with “maintenance”

2. Categories of maintenance
   - Corrective – changes to fix errors (only 20% of maintenance!)
   - Adaptive – changes for use in a new environment
   - Perfective – changes to meet new or different customer needs (enhancements)
   - Preventative – changes to fix errors before they occur (reengineering)

3. Because of the broader scope, the term evolution is sometimes used instead of maintenance

4. Error reports from users should be tracked separately than requests for change

5. Maintenance costs as a percentage of overall costs have been steadily increasing for the past several decades, until it is now about 95% of the effort related to software

6. After the initial release of a software product, 10-15% of development costs should be allocated per year for maintenance

7. In a particular year, 10-15% of the code is generally “touched” during maintenance

8. Maintenance process for dealing with a single request (from the IEEE Standard for Software Maintenance)
   - Start with a Modification Request
   - Repeat as necessary the following:
     - Classification and Identification
     - Analysis
     - Design
     - Implementation
     - System Test
     - Acceptance Test
     - Delivery
9. Maintenance organization

9.1 Relationship to Software Development
- Maintenance is done within each project development team
- Separate maintenance organization

9.2 Roles and functions for maintenance of a particular project
- Maintainers
  - Software modification
  - Documentation management
- Quality Assurance Personnel
  - Testing
  - Reviews and inspections
  - Configuration management

10. Maintenance planning activities begin during development (2-3 years before release for some multiyear projects). The three major planning steps:

- Defining the maintenance concept
- Developing a maintenance plan
- Analysis to determine what resources are needed to implement the plan

11. Transition to Maintenance

11.1 At some point, the development team’s responsibility for the software has to be transitioned to the maintenance organization (assuming that the developers will not also be doing the maintenance)

11.2 The maintenance organization should develop a transition plan which addresses all of the above issues

12. The Laws of Software Evolution

12.1 Not like physical laws (e.g. Newton’s Three Laws) but are instead are facts that are often observed as happening during the evolution process

12.2 There are eight laws that have been formulated over the years by M.M. Lehman; here are the first two:

I - Continuing Change: An E-type (evolutionary type) program that is used must be continually adapted else it becomes progressively less satisfactory.

II – Increasing Complexity: As a program is evolved its complexity increases unless work is done to maintain or reduce it.