Software Maintenance
Software Maintenance Terminology

• **Software Evolution**
  – a continuous change from a lesser, simpler, or worse state to a higher or better state

• **Software Maintenance**
  – consists of the activities required to keep a software system operational and responsive after it is accepted and placed into production.
Software Evolution

‘Software development does not stop when a system is delivered but continues throughout the lifetime of the system’. (Sommerville, 2004)

- The system changes relate to changing business and user needs
- The system evolves throughout its lifetime through a seamless process
- The process is a spiral process involving requirements, design & implementation throughout the lifetime of the system
Software Maintenance

‘Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment’.

(SWEBOK, 2004)

• 40-60% of the maintenance effort is devoted to understanding the software to be modified
• Defect and change tracking is essential
Software Maintenance Problems

- Most computer systems are difficult and expensive to maintain
- Software changes are poorly designed and implemented
- The repair and enhancement of software often injects new bugs that must later be repaired
Relative Costs of Maintenance

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>3%</td>
<td>Requirements definition</td>
</tr>
<tr>
<td>3%</td>
<td>Preliminary design</td>
</tr>
<tr>
<td>5%</td>
<td>Detailed design</td>
</tr>
<tr>
<td>7%</td>
<td>Implementation</td>
</tr>
<tr>
<td>15%</td>
<td>Testing</td>
</tr>
<tr>
<td>67%</td>
<td>Operations and Maintenance</td>
</tr>
</tbody>
</table>

- The majority of a software budget in large companies is devoted to maintaining systems.
- Sommerville (2004) states that 90% of software costs are evolution costs.
  - 60-80% of software cost is spent on maintenance
Differences:
Software Development & Maintenance

- Constraints of an existing system
  » Changes must conform or be compatible with an existing architecture, design and code constraints

- Shorter time frames
  » Development spans 1 or more years
  » Maintenance spans hours or up to 6 months

- Available test data
  » Development creates all test data from scratch
    Maintenance uses existing test data with regression testing creating new data for the changes
Types of Maintenance

• Corrective Maintenance
  • Reactive process focused on fixing defects

• Adaptive Maintenance
  • Reactive process to keep product viable in a changing environment

• Perfective Maintenance
  • Proactive process to improve software quality

• Preventative Maintenance
  • Proactive process to detect/correct faults before users
Software Maintenance Process

- Seven-step approach:
  - Step 1 - Change Management
  - Step 2 - Impact Analysis
  - Step 3 - System Release Planning
  - Step 4 - Design the Changes
  - Step 5 - Code the Changes
  - Step 6 - Test the Changes
  - Step 7 - System Release