1. Introduction

Emphasis on quality, productivity and repeatable processes has led to the use of innovative software development tools and techniques. In the 1960s and 1970s, software production implied a tortuous effort of writing computer code from the ground up. That old view is the basis for predicting software project time and cost, using estimated lines of code.

As more effective software tools became available, that focus shifted to functions. The dynamics of this environment continue, with heavy use of tools to generate code by computer rather than by humans, the reuse of parts of programs and other efficiencies.

In this topic, we will examine the different types of software quality tools and how they can be used to increase the efficiency of information systems (IS) projects.

Objectives: Software Quality Tools

Upon completion of this topic, you should be able to
- describe the various kinds of software quality tools
- discuss the relative advantages and disadvantages of open source software over commercially available software

2. Software Quality Tools

There are different kinds of software quality tools. Hunt and Thomas (2004) described three: version control, unit testing and automation.

Click each term to learn more about each quality tool.

Software Quality Tools

Version control

It is a means to keep track of the software developed so that prior versions can be
recovered. Developers can thus experiment with code, knowing that if a particular experiment fails, the prior version can be recovered. A version control repository saves prior versions. This can be especially helpful when multiple developers are working on the same system.

**Unit testing**

Unit testing is a primary source of developer feedback. The computer checks various outputs of a code, which guarantees that a new code still accomplishes the prior version's functionality. Unit testing code should be automatic (with unambiguous conclusions), independent (where any test can be run at any time and in any order) and repeatable.

**Automation**

Any repetitive procedure should be automated to include building code, running unit tests, creating releases and installation. Automation provides consistency, reliability and repeatability.

**Functions of software quality tools**

Software quality tools include many functions, including
- software prediction
- testing, including modelling and documentation, test creation, execution and results management
- generation of test data
- source code analysis
- quality assurance, including auditing

### 3. Open Source Software

The use of open source software represents an alternative to using commercial software sold by a vendor. Spinellis and Szyperski (2004) noted that over 70,000 projects are available on one website for shared use and over 5,000 Perl modules are available on another site for use by anyone. The proliferation and availability of open source software have changed the software development environment. Open source software has visible source code and users are granted the right to make derivatives.

The open source community provides a rich base of reusable software. This is not to say that there are no downside factors in open source code. The following table lists the benefits and drawbacks of open source software:

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reused open source code will usually be of higher quality due to error correction of flaws identified by past users.</td>
<td>Open source components often include features that are not backward compatible and may create unwanted dependencies.</td>
</tr>
<tr>
<td>Functionality will often be more complete.</td>
<td>Reusing open source components can cause licensing problems.</td>
</tr>
<tr>
<td>The cost of zero is very attractive.</td>
<td>The quality of the software varies</td>
</tr>
</tbody>
</table>
In addition, some organisations tend to shy away from open source code because the applications are not supported through technical support agreements that are possible to vendors offering commercial software.

Many tools have been developed in practice to aid the software development process. The use of these tools is very compatible with the ideas of software quality discussed earlier. They represent partial means for software development organisations to attain Level 5 in the Capability Maturity Management model.

4. **Summary**

**This topic covered the following main points:**

- There are many software quality tools available. These tools can accomplish a variety of functions, including testing, documentation and quality assurance. In many cases, they automated tedious clerical activities.
- Open source software is an alternative to commercial software. Open source software is usually cheaper and the organisation has access to the source code.
- However, open source software can suffer from quality problems as well as from a lack of support.