Unit 7: Using Tools

Objectives

- To introduce the use of automated support for software engineering.

Automated Support for SE

- Why we need it
  - Clerical scale
  - Dim computers catch dim errors! Liberate the developer to do the "hard" work (skilled people are expensive)
  - Analytical techniques (and fancy graphics) which are not feasible manually
  - Distributed work and virtual organisations which require electronic communication
  - Save time - save money
  - Enforcement of good practice
  - Management oversight

Automated Support for SE

- Automated support for software engineering can be divided into three broad categories
  - Tools (and toolkits)
  - Workbenches
  - Environments

Tools

- A tool is a software aid for accomplishing a particular activity within the development process.

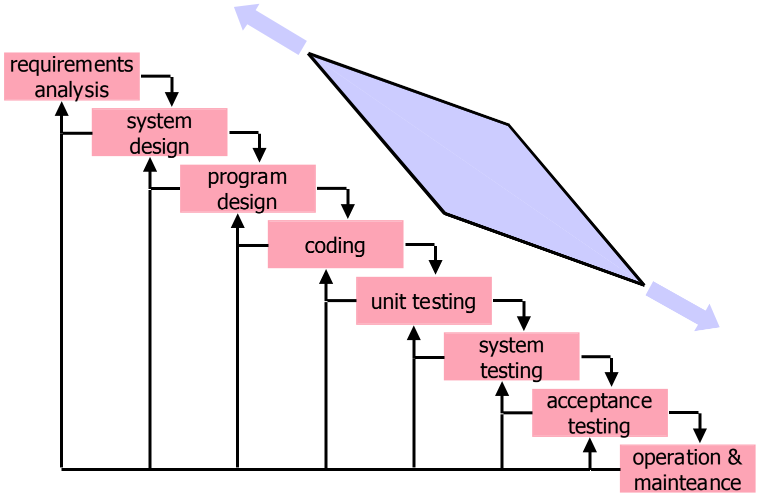
- A toolkit is a loose personal collection of such tools

Tools

- There is a very wide variety of tools in software engineering:
  - Documentation and editing tools
  - Testing tools
  - Analysis tools
  - Design tools
  - Metrics tools
  - Costing and estimation tools
  - Debugging tools
  - Reengineering tools
  - Bug tracking tools

and many, many more ...

Tools Trends



requirements analysis

system design

program design

coding

unit testing

system testing

acceptance testing

operation & mainteance

Workbenches

- A workbench is where tools are collected to support a coherent set of activities but where tool interworking is not fully supported

Workbenches

- For example Microsoft Visual Studio
  - Visual Basic® 6.0 development system
  - Visual C++® 6.0 development system
  - Visual J++® 6.0 development system
  - Visual InterDev™ 6.0 Web development system
  - Visual FoxPro® 6.0 database development system
  - Visual Modeler 2.0
  - Visual Studio Analyzer
  - Oracle and SQL Server Database Designer
  - Visual Database Tools
  - Visual SourceSafe 6.0
  - Microsoft Repository 2.0
  - Visual Component Manager 2.0.

Environment

- An environment is a software framework within which a group of tools is organised to support a coherent set of activities and which supports tool invocation and interworking

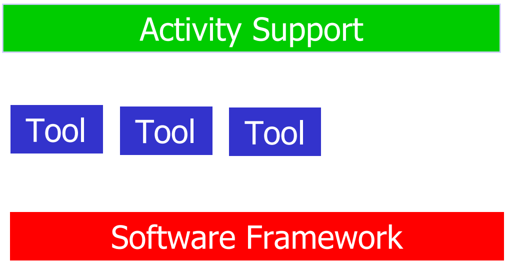
Environment

workflow

user interface management

version & configuration management

data management



Activity Support

Tool

Tool

Tool

Software Framework

Rational Suite: the nearest example

- Project Management
  - > Rational Unified Process
  - > Rational ClearQuest
  - > Rational ClearDDTS
- System Definition
  - > Rational Suite AnalystStudio
  - > Rational RequisitePro
  - > Rational Rose family
- Software Development
  - > **Rational Suite DevelopmentStudio**
  - > Rational ClearCase family
  - > Rational Rose family
  - > Rational ClearQuest
  - > Rational ClearDDTS
  - > Rational Purify
  - > Rational Quantify
  - > Rational PureCoverage
  - > Rational Apex family
- System Testing
  - > Rational Suite TestStudio
  - > Rational Suite PerformanceStudio
  - > Rational SiteLoad
  - > Rational TeamTest
  - > Rational Visual Test
  - > Rational TestFoundation
  - > Rational Purify
  - > Rational Quantify
  - > Rational PureCoverage
  - > Rational ClearQuest
  - > Rational ClearDDTS
- Team Unifying Platform
  - > Rational Unified Process
  - > Rational ClearCase family
  - > Rational ClearQuest
  - > Rational ClearDDTS
  - > Rational RequisitePro
  - > Rational SoDA

CASE

- Automated support for software engineering is sometimes called
  - – Computer Aided Software Engineering or CASE
- CASE is often divided into
  - – Upper CASE (support for requirements and design activities)
  - – Lower CASE (support for implementation, testing and maintenance)

You should know these terms

Illustrative Examples

- Analysis and Design Tool
  - Rational ROSE: provides diagram editing, consistency checking and skeleton code generation
- Requirements Management Tool
  - DOORS: provides document construction and management, report generation, information linking and traceability

Its easier to understand when you see a demo ...

Typical SE Tool Problems

- Integration with “office” tools
- Lack of standardisation
- Complexity and user interface difficulty
- Cost
- Speed
- Robustness
- Scalability

How to select a tool

- First
  - Decide on a process
  - Determine desired coverage
  - Determine key interworking requirements
- Identify candidates based on these
- Then
  - Check general features (robustness, scalability, support etc.)
  - Check tailorability
  - Check training
  - Check consulting and process support

Key Points

- Use tools. It is quicker, cheaper and better.
- Get automated support for as much of the process as possible. Don’t forget where the effort goes.
- Make sure your tools interwork to maximise the benefit.
- A good student set of free tools can be found at http://burks.bton.ac.uk/burks/index.htm
- “The cobblers children are oft ill shod!”