Configuration management

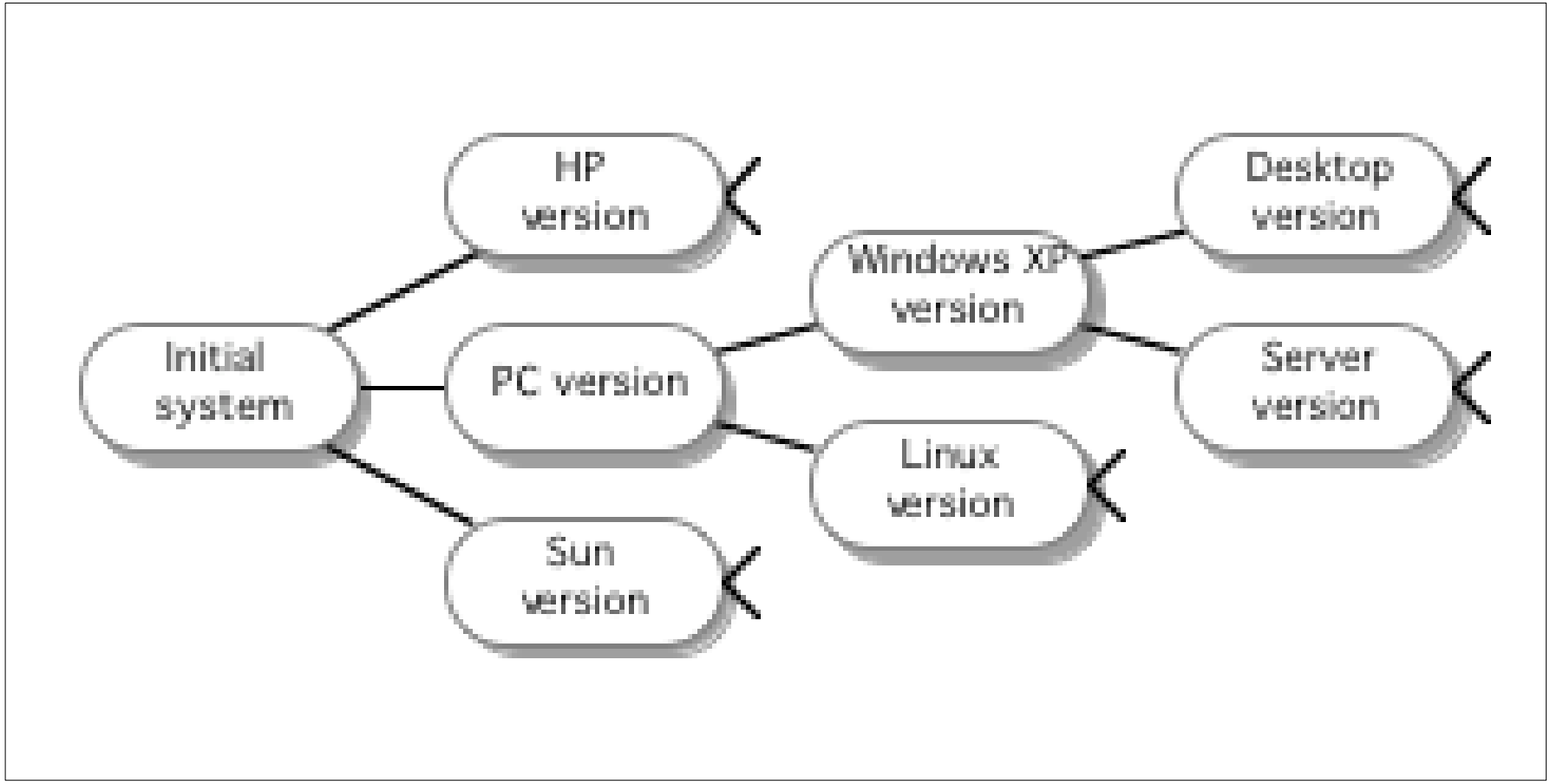
Configuration management

- New versions of software systems are created as they change:
  - For different machines/OS;
  - Offering different functionality;
  - Tailored for particular user requirements.
- Configuration management is concerned with managing evolving software systems:
  - System change is a team activity;
  - CM aims to control the costs and effort involved in making changes to a system.

Configuration management

- Involves the development and application of procedures and standards to manage an evolving software product.
- CM may be seen as part of a more general quality management process.
- When released to CM, software systems are sometimes called *baselines* as they are a starting point for further development.

System families



Configuration management planning

- All products of the software process may have to be managed:
  - Specifications;
  - Designs;
  - Programs;
  - Test data;
  - User manuals.
- Thousands of separate documents may be generated for a large, complex software system.

The CM plan

- Defines the types of documents to be managed and a document naming scheme.
- Defines who takes responsibility for the CM procedures and creation of baselines.
- Defines policies for change control and version management.
- Defines the CM records which must be maintained.

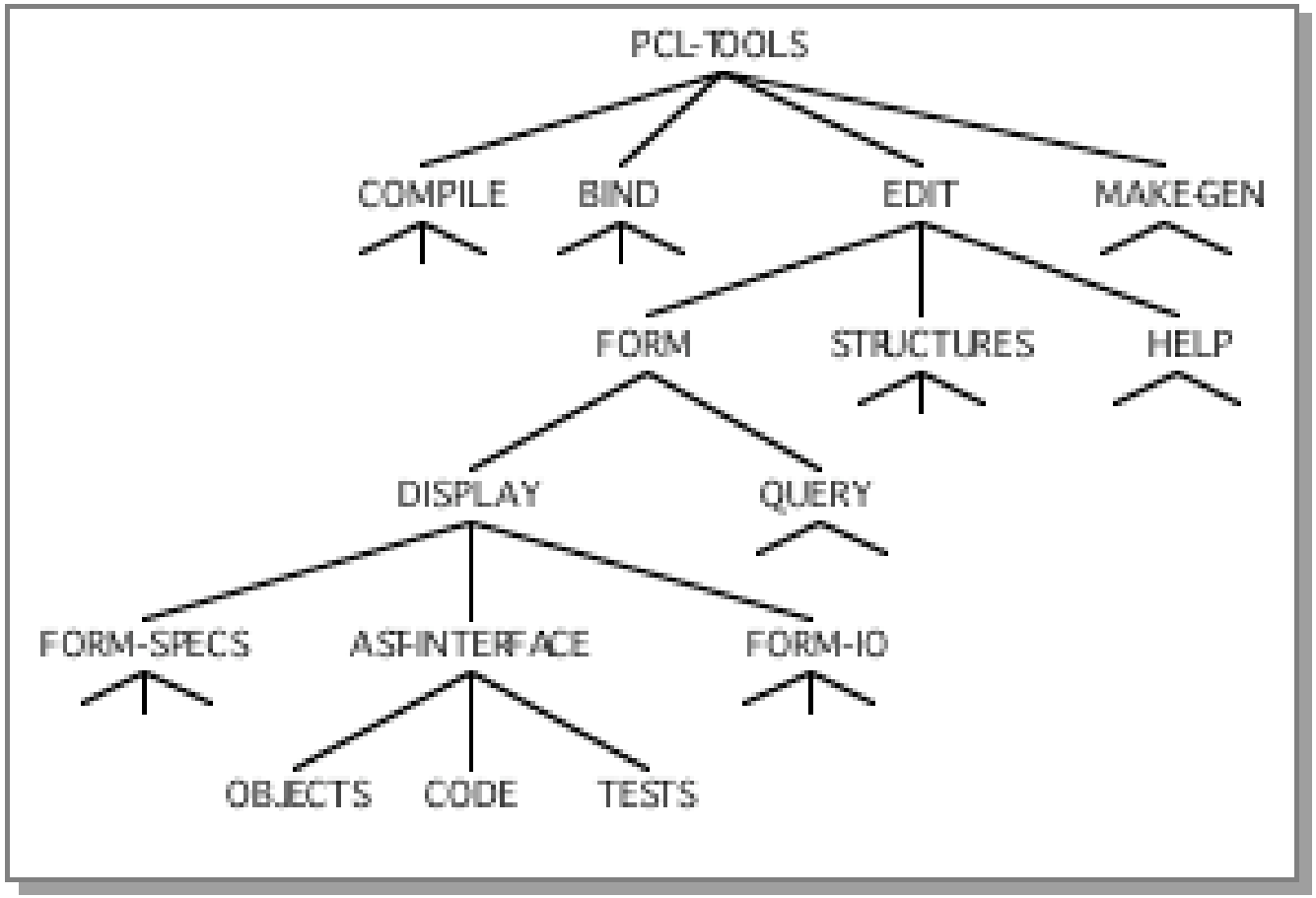
The CM plan

- Describes the tools which should be used to assist the CM process and any limitations on their use.
- Defines the process of tool use.
- Defines the CM database used to record configuration information.
- May include information such as the CM of external software, process auditing, etc.

Configuration item identification

- Large projects typically produce thousands of documents which must be uniquely identified.
- Some of these documents must be maintained for the lifetime of the software.
- Document naming scheme should be defined so that related documents have related names.
- A hierarchical scheme with multi-level names is probably the most flexible approach.
  - PCL-TOOLS/EDIT/FORMS/DISPLAY/AST-INTERFACE/CODE

Configuration hierarchy



The configuration database

- All CM information should be maintained in a configuration database.
- This should allow queries about configurations to be answered:
  - Who has a particular system version?
  - What platform is required for a particular version?
  - What versions are affected by a change to component X?
  - How many reported faults in version T?
- The CM database should preferably be linked to the software being managed.

Change management

- Software systems are subject to continual change requests:
  - From users;
  - From developers;
  - From market forces.
- Change management is concerned with keeping track of these changes and ensuring that they are implemented in the most cost-effective way.

Change tracking tools

- A major problem in change management is tracking change status.
- Change tracking tools keep track the status of each change request and automatically ensure that change requests are sent to the right people at the right time.
- Integrated with E-mail systems allowing electronic change request distribution.

Change control board

- Changes should be reviewed by an external group who decide whether or not they are cost-effective from a strategic and organizational viewpoint rather than a technical viewpoint.
- Should be independent of project responsible for system. The group is sometimes called a change control board.
- The CCB may include representatives from client and contractor staff.

Derivation history

- This is a record of changes applied to a document or code component.
- It should record, in outline, the change made, the rationale for the change, who made the change and when it was implemented.
- It may be included as a comment in code. If a standard prologue style is used for the derivation history, tools can process this automatically.

Component header information

Version and release management

- Invent an identification scheme for system versions.
- Plan when a new system version is to be produced.
- Ensure that version management procedures and tools are properly applied.
- Plan and distribute new system releases.

Versions/variants/releases

- Version An instance of a system which is functionally distinct in some way from other system instances.
- Variant An instance of a system which is functionally identical but non-functionally distinct from other instances of a system.
- Release An instance of a system which is distributed to users outside of the development team.

Version identification

- Procedures for version identification should define an unambiguous way of identifying component versions.
- There are three basic techniques for component identification
  - Version numbering;
  - Attribute-based identification;
  - Change-oriented identification.

Release management

- Releases must incorporate changes forced on the system by errors discovered by users and by hardware changes.
- They must also incorporate new system functionality.
- Release planning is concerned with when to issue a system version as a release.

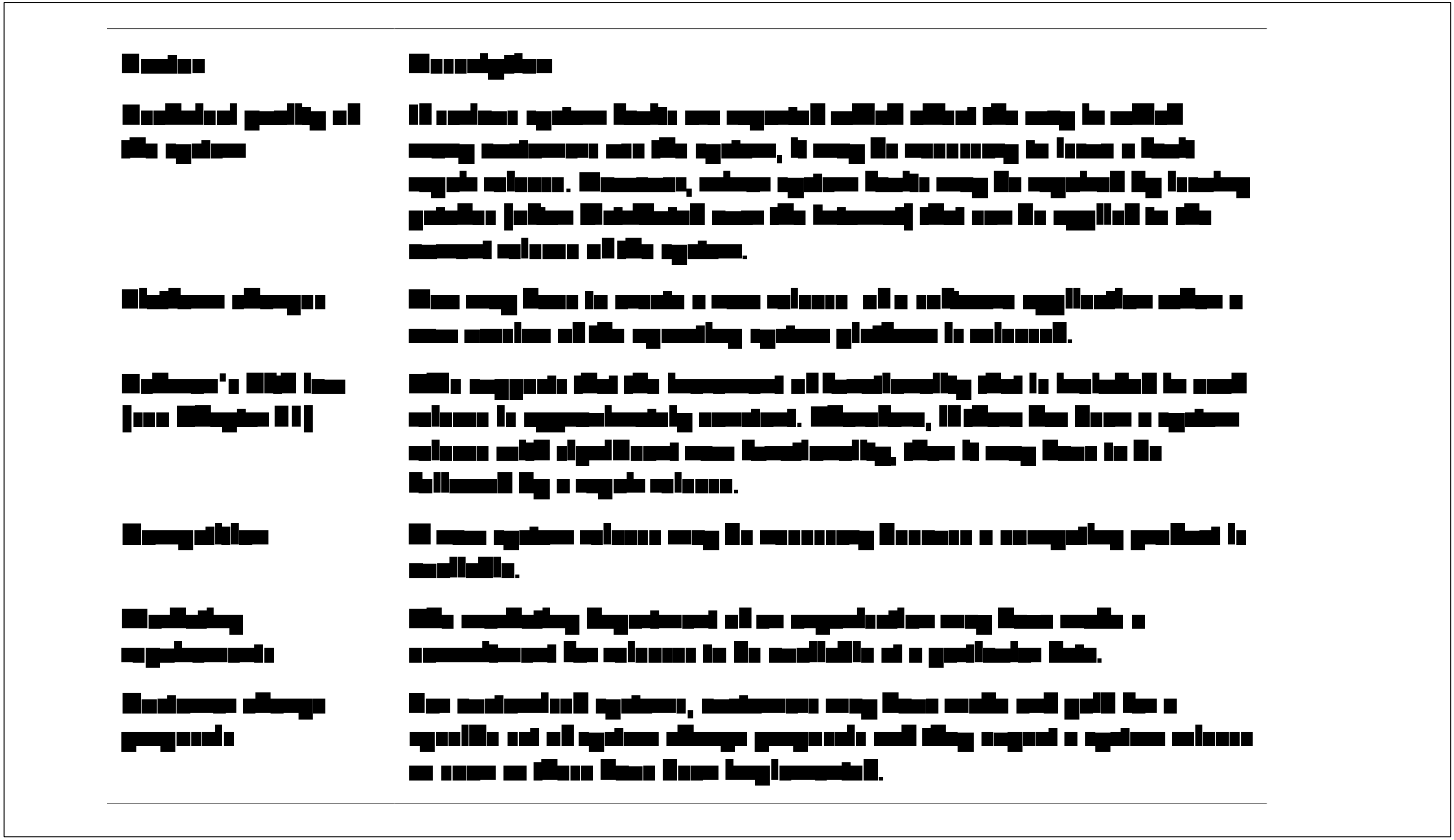
System releases

- Not just a set of executable programs.
- May also include:
  - Configuration files defining how the release is configured for a particular installation;
  - Data files needed for system operation;
  - An installation program or shell script to install the system on target hardware;
  - Electronic and paper documentation;
  - Packaging and associated publicity.
- Systems are now normally released on optical disks (CD or DVD) or as downloadable installation files from the web.

Release problems

- Customer may not want a new release of the system
  - They may be happy with their current system as the new version may provide unwanted functionality.
- Release management should not assume that all previous releases have been accepted. All files required for a release should be re-created when a new release is installed.

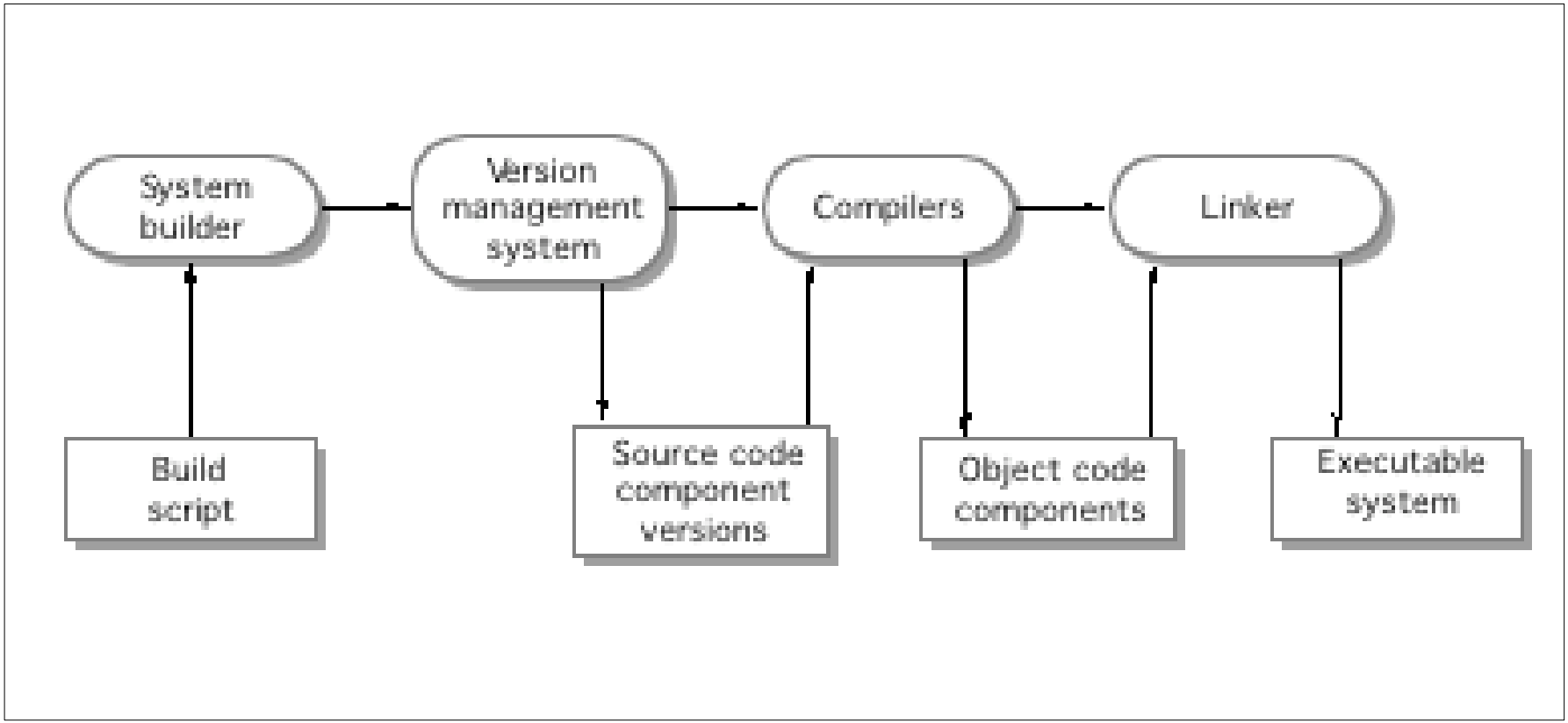
System release strategy



System building

- The process of compiling and linking software components into an executable system.
- Different systems are built from different combinations of components.
- This process is now always supported by automated tools that are driven by ‘build scripts’.

System building



CASE tools for configuration management

- CM processes are standardised and involve applying pre-defined procedures.
- Large amounts of data must be managed.
- CASE tool support for CM is therefore essential.
- Mature CASE tools to support configuration management are available ranging from stand-alone tools to integrated CM workbenches.

Change management tools

- Change management is a procedural process so it can be modelled and integrated with a version management system.
- Change management tools
  - Form editor to support processing the change request forms;
  - Workflow system to define who does what and to automate information transfer;
  - Change database that manages change proposals and is linked to a VM system;
  - Change reporting system that generates management reports on the status of change requests.

Version management tools

- Version and release identification
  - Systems assign identifiers automatically when a new version is submitted to the system.
- Storage management.
  - System stores the differences between versions rather than all the version code.
- Change history recording
  - Record reasons for version creation.
- Independent development
  - Only one version at a time may be checked out for change. Parallel working on different versions.
- Project support
  - Can manage groups of files associated with a project rather than just single files.

System building

- Building a large system is computationally expensive and may take several hours.
- Hundreds of files may be involved.
- System building tools may provide
  - A dependency specification language and interpreter;
  - Tool selection and instantiation support;
  - Distributed compilation;
  - Derived object management.

A quick look...

- BugZilla
  - Bug Tracking
  - Change Management
  - …
- CVS
  - Concurrent Versioning System
- Makefile and ANT
  - Build-Distribute tool