

Software Configuration Management



Lecture: Configuration identification
Chapter: 7

Niels Veerman
René Krikhaar

Lecture objectives



- ◆ Learn about configuration identification processes
 - ◆ Learn about system decomposition
 - ◆ Learn about unique identification
 - ◆ Learn about baselines

The aim of configuration identification



- ◆ The goal is to support project management by breaking down the development effort into manageable items that are uniquely identified
- ◆ Too many items result in
 - ◆ Increased administrative overhead
 - ◆ Inefficient status reporting to support project management
- ◆ Too few items result in
 - ◆ Inefficient modifications and collaborative development
 - ◆ Inefficient status reporting to support project management
- ◆ Configuration identification is closely related to system architecture design

Concepts of configuration identification



- ◆ Selection (system decomposition into CIs)
- ◆ Identification (numbering/naming scheme for CIs)
- ◆ Baselines (consistent set of CIs)

CI selection

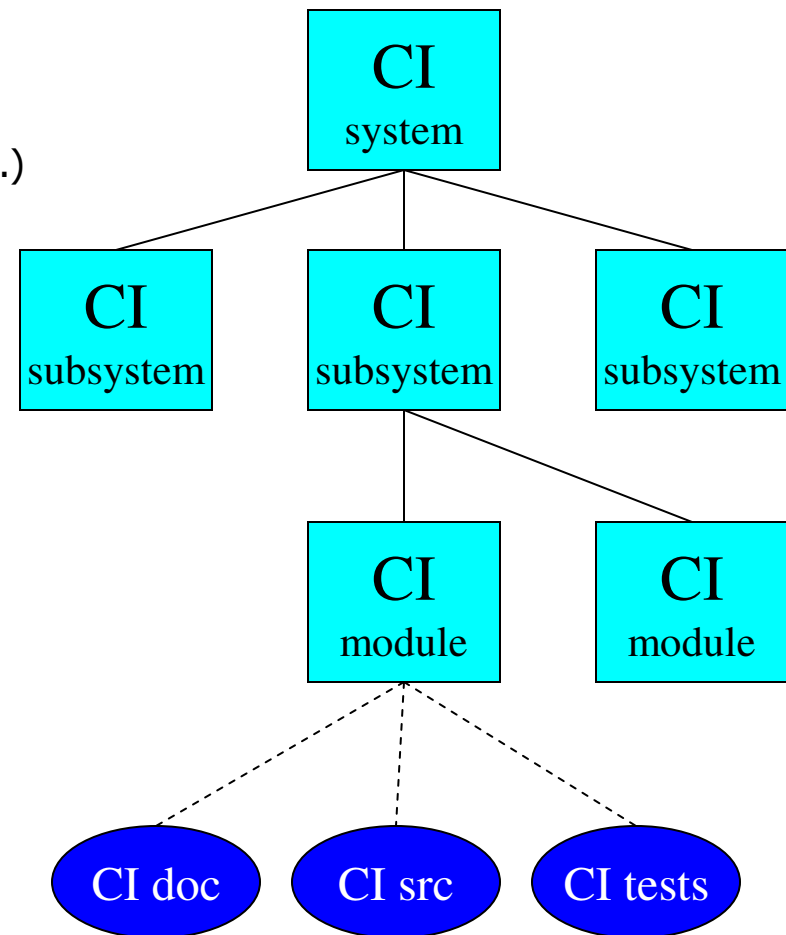
◆ Product CIs

- ◆ In-house items
 - ◆ Documents
 - ◆ Specifications
 - ◆ User documents (installation,usage,..)
 - ◆ Source code
 - ◆ C, C++, Java, C#, Cobol,..
 - ◆ Build files
 - ◆ Makefile, Antfile, MSVS project file
 - ◆ Test files
 - ◆ Releases

- ◆ Third-party items

◆ Support CIs (mostly third-party)

- ◆ IDEs
- ◆ Compilers
- ◆ Build tools
- ◆ Test tools
- ◆ SCM tools

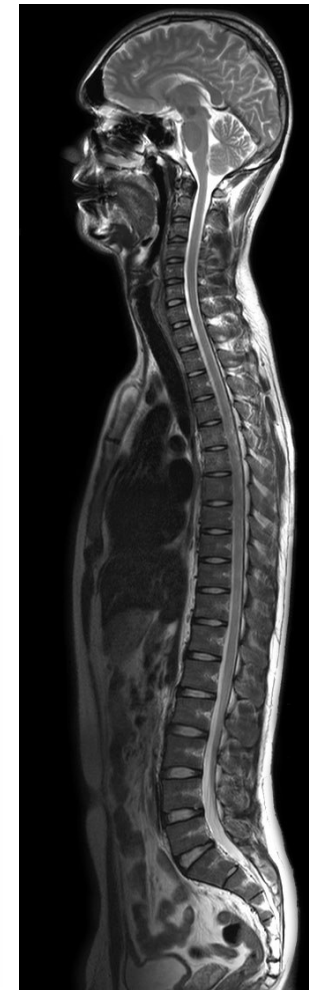
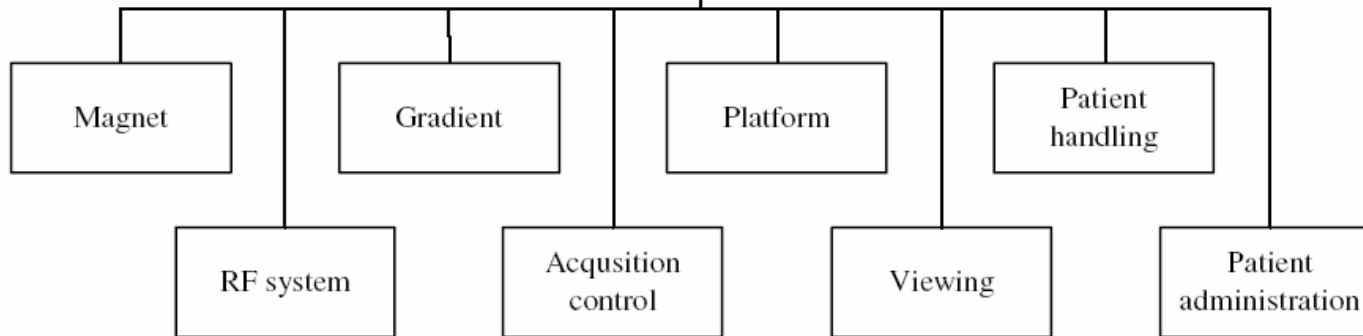


CI selection

- ◆ High-level CI selection in an MRI scanner



MRI system



CI identification/numbering

- ◆ CI identification defines numbering/naming schemes for CIs to ensure unique identification
- ◆ Files are uniquely identified by their location, path, branch, name and version/revision number

serverX: \project1\trunk\car\engine\control\ fuelpump.c Rev:4586

location	path and branch	name	revision
----------	-----------------	------	----------


- ◆ Releases can be internal or external, and numbered with X.Y.Z[A|Bn]
 - ◆ X: major release number
 - ◆ Y: feature release number
 - ◆ Z: defect repair number
 - ◆ A|B: alpha / beta release with a number 'n'
- ◆ Third-party items are usually identified by their name and release
 - ◆ Windows XP SP2, GNU C Compiler 4.3.0, Subversion 1.4.6, McAfee 8.5

CI example



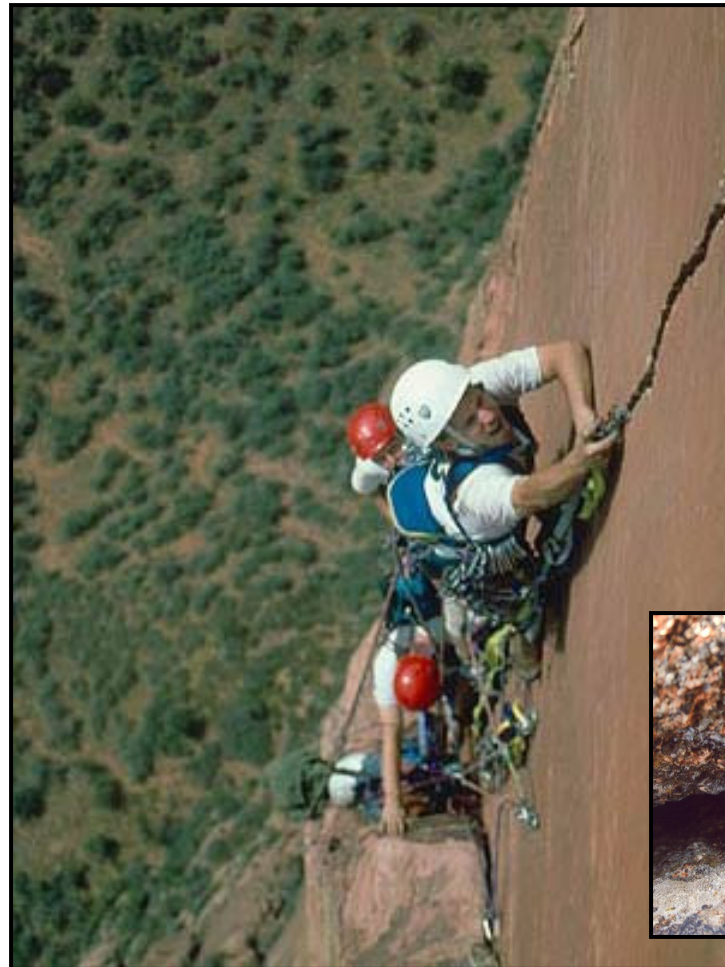
- ◆ Configuration items real-life example (not included in sheet)

Baselines – definitions

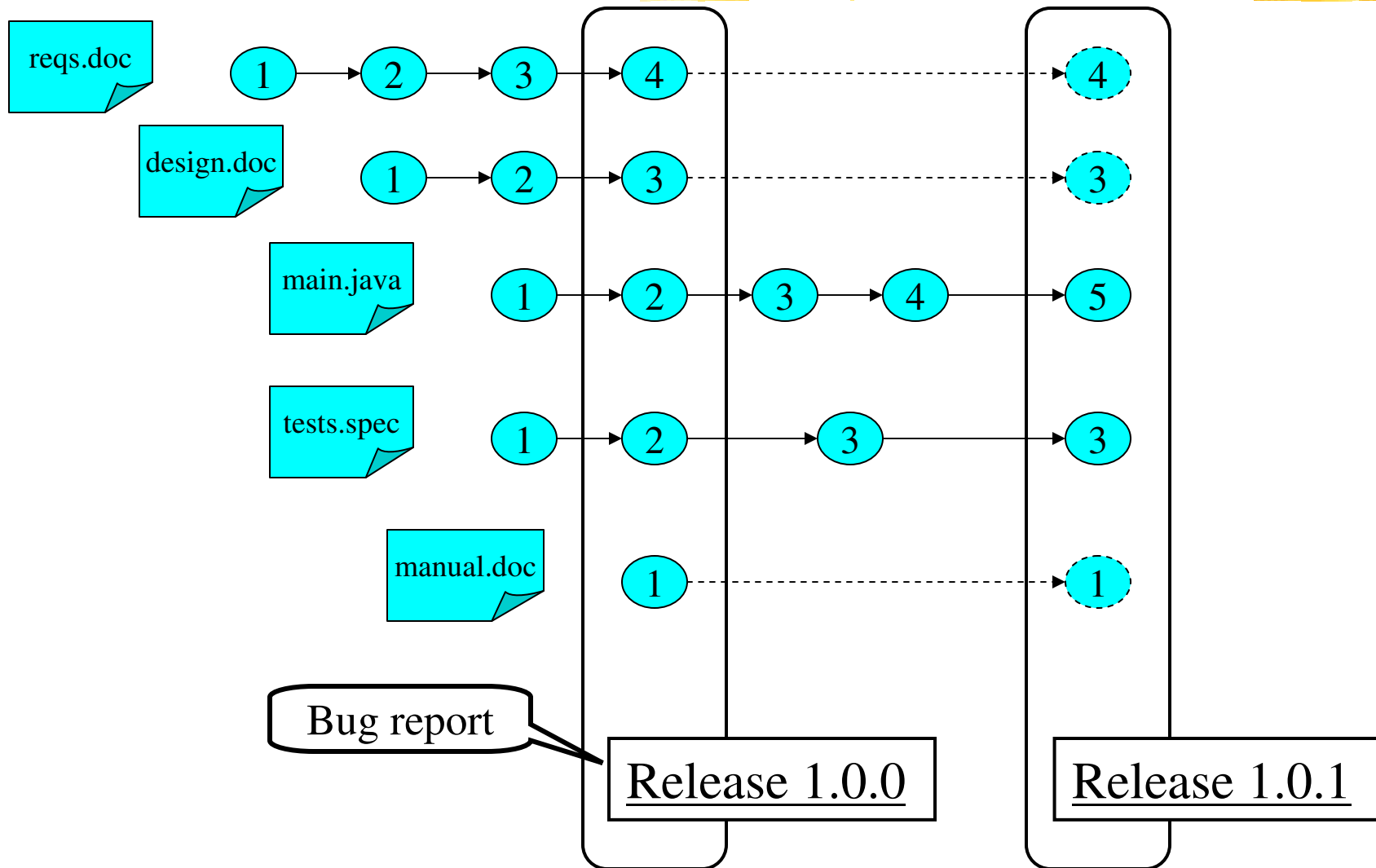


- ◆ **IEEE:** *“a specification or product that has been formally reviewed and agreed on, which thereafter serves as the basis for further development and which can only be changed through formal change control procedures”*
- ◆ A baseline is consistent set of CIs
 - ◆ Also called Tagging, Labeling, ...
- ◆ A baseline is a reference basis for evolution and release in any lifecycle phase
- ◆ The baseline concept is the foundation for configuration management and the start for configuration control

Baselines



Baselines – examples



Baselines & dev. process

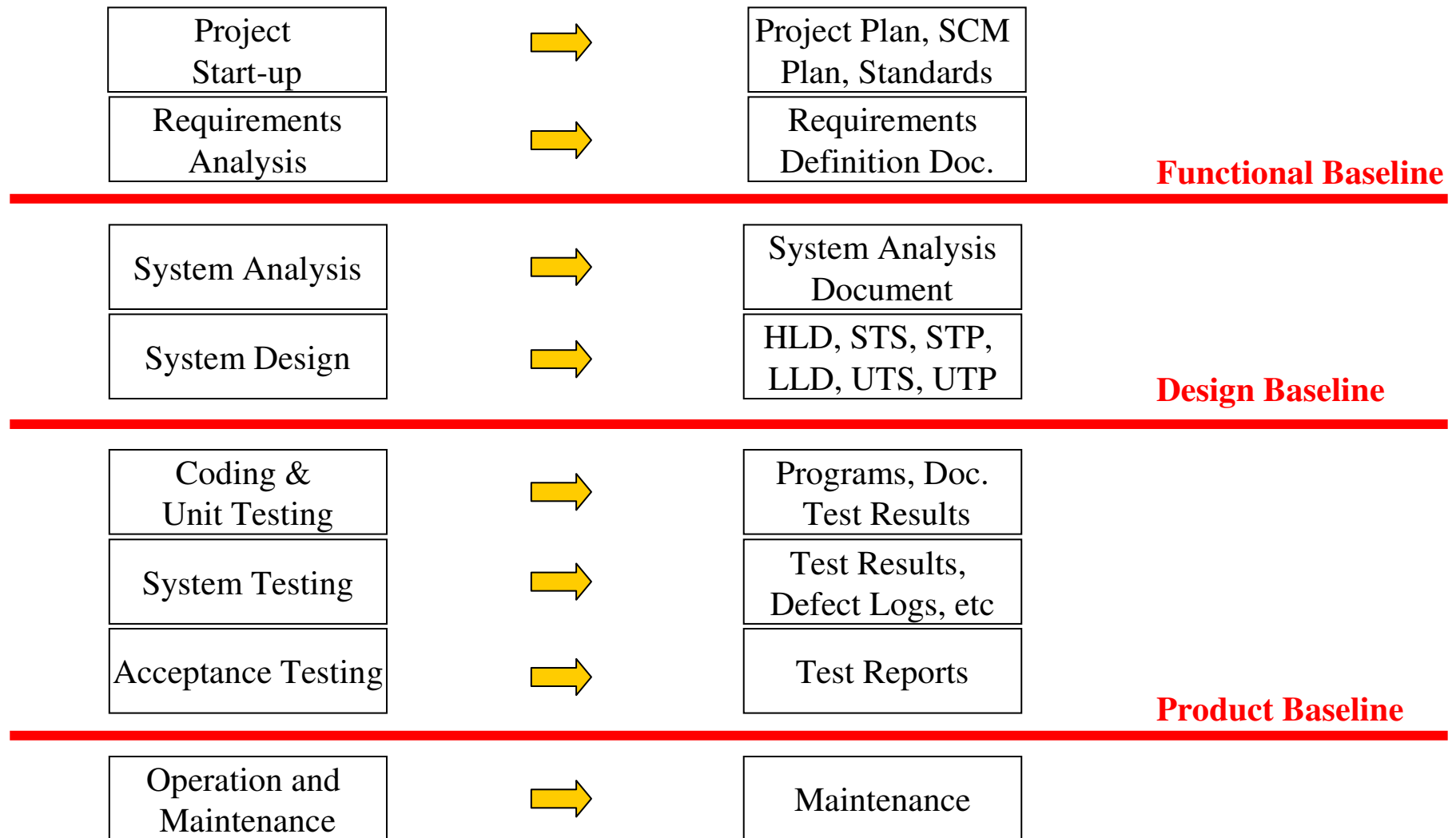


- ◆ **Waterfall.** The waterfall strategy, also called “once-through”, consists of performing the development process a single time. Simplistically: determine user needs, define requirements, design the system, implement the system, test, fix, and deliver.
- ◆ **Incremental.** The incremental strategy determines user needs and defines the system requirements, then performs the rest of the development in a sequence of builds. The first build incorporates part of the planned capabilities, and so on, until the system is complete.
- ◆ **Evolutionary.** The evolutionary strategy also develops a system in builds but differs from the incremental strategy in acknowledging that the user need is not fully understood and all requirements cannot be defined upfront. In this strategy, user needs and system requirements are partially defined at the begin, then are further defined in each succeeding build.

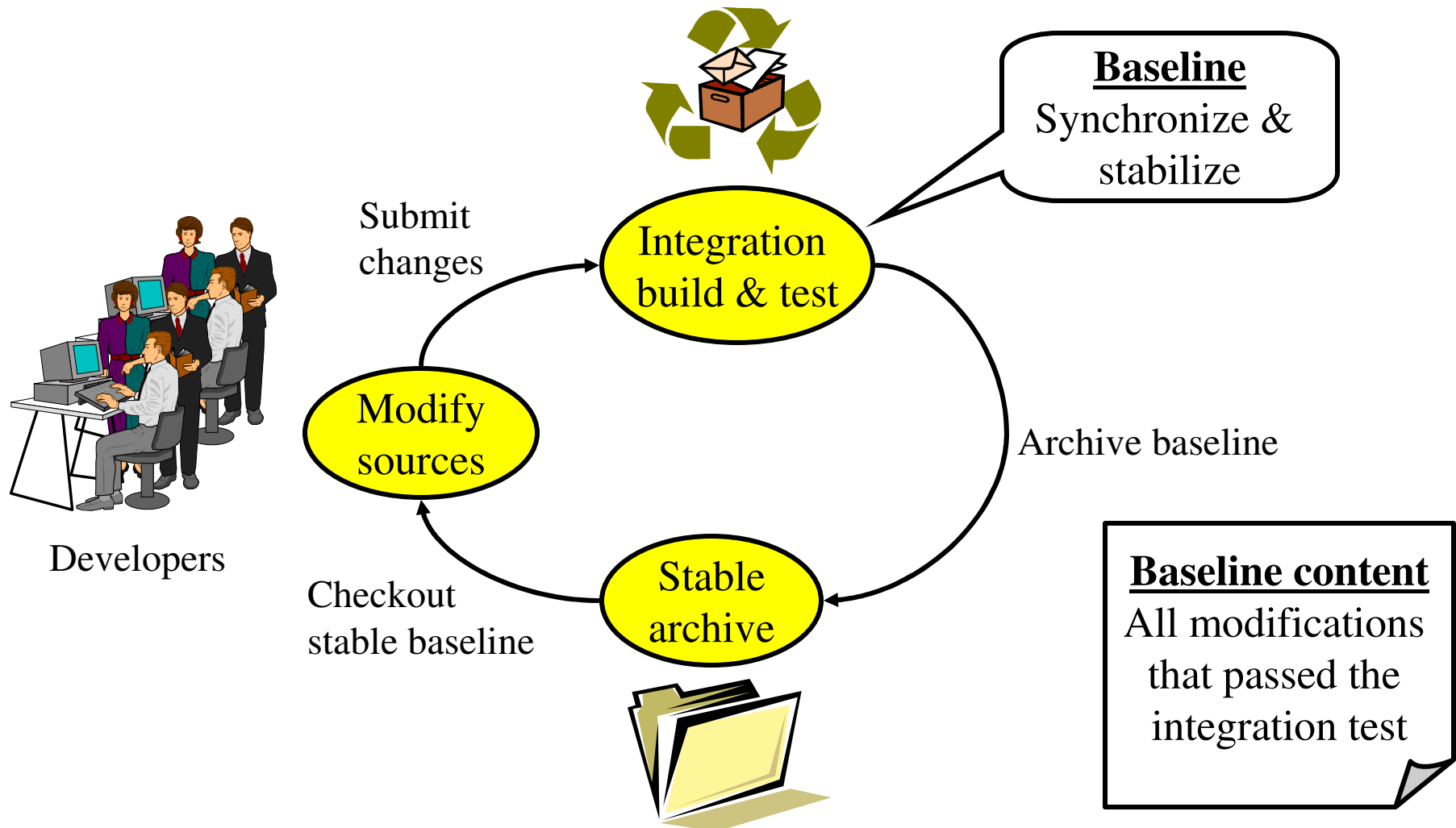
Waterfall baselines

Process phase

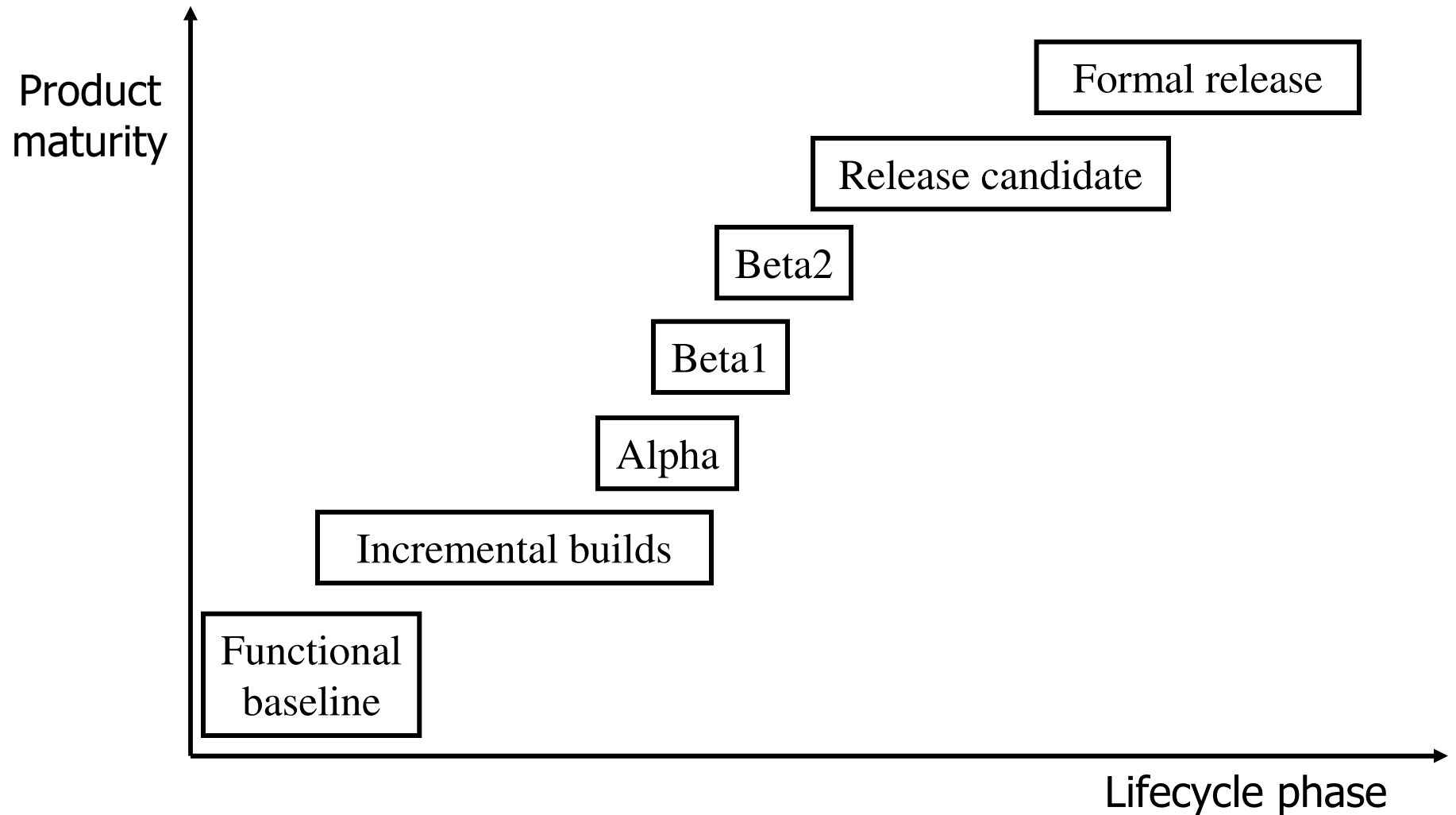
Baseline content



Incremental/evolutionary baselines



Incremental baselines



Summary



- ◆ Relation between configuration identification and system architecture design
- ◆ Difference between Product CIs and Support CIs
- ◆ Baselines types in different development processes
- ◆ A baseline has an associated level of configuration control