Configurations Management

SQZ

Which component belongs to system XYZ ?

Which components have been finished already ?

In which version has the fault been cleared ?

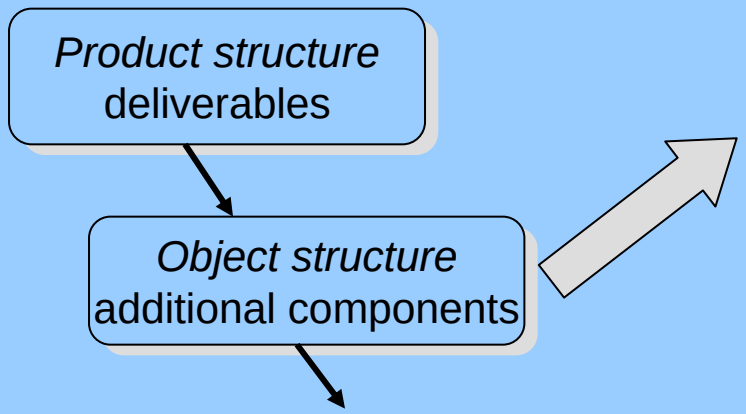
Where can I find the current version of the module ?

What's the difference between the version for Kuwait and the one for Costa Rica ?

I thought we had already cleared this fault !

CM - What's That ?

Regulations for all the tasks needed to manage the results achieved and the tools used throughout a project.



Documents: development, project, product, ...

Object code, executables, subsystems, complete product

Source elements, GUI,...

Test cases, test data, test results,...

Tools (compilers, test drivers,...)

Institution that performs these tasks

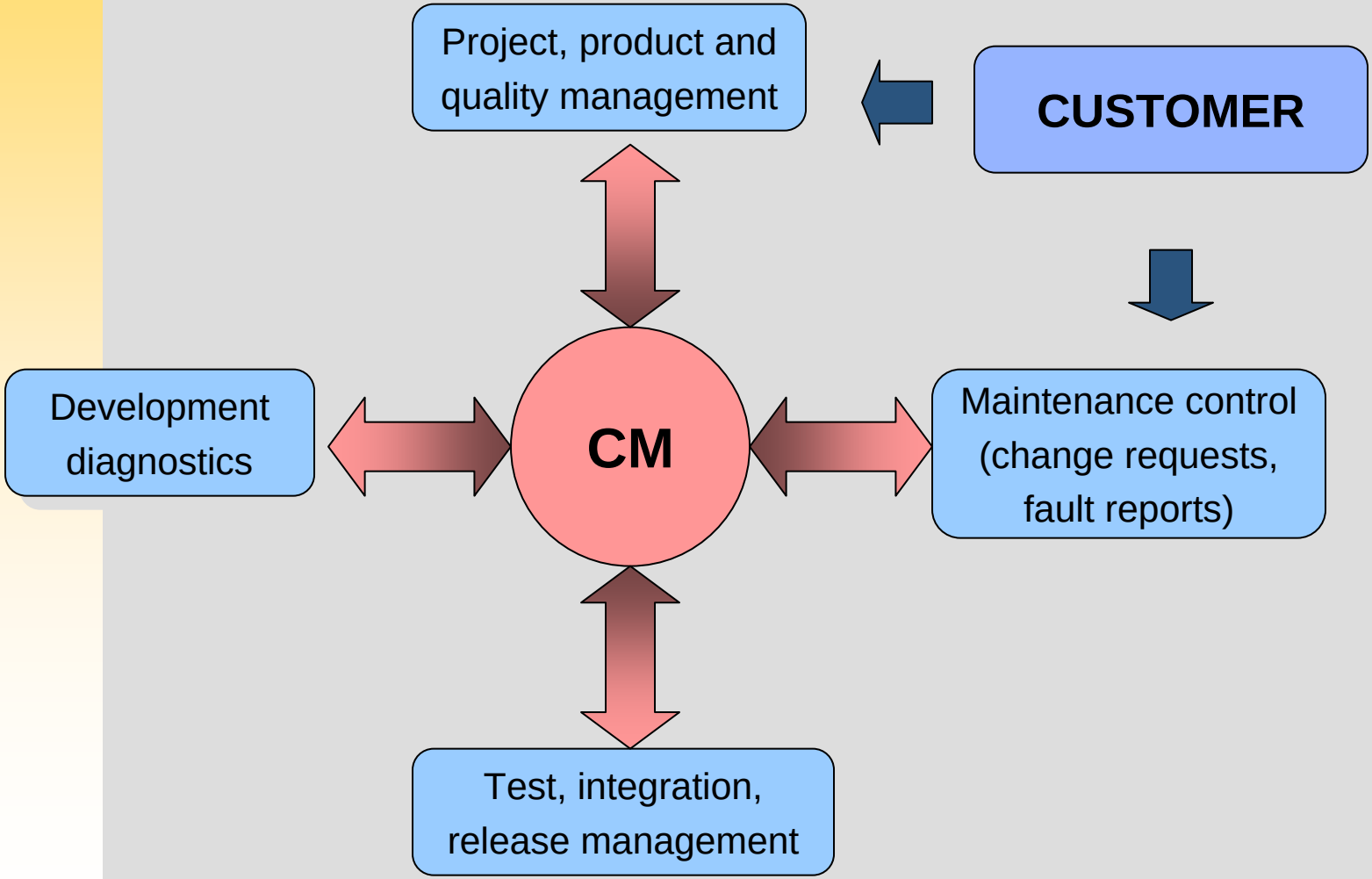
- project manager (in small-scale projects)
- project manager and developers
- a special-purpose CM group (in large-scale projects)

The project management is responsible for defining the CM !

Why CM?

- **Increasing complexity of projects**
  - large number of objects to manage
  - different types of objects
  - large staff numbers / different locations
  - shorter development cycles
  - state-of-the-art development methods and tools
  - a variety of HW platforms, heterogeneous systems,...
- **Reuse**
- **Quality assurance**
  - Requirement defined in CMM and ISO 9001

Logistics hub



HOW ?

BY WHICH MEANS ?

Too Hasty Call for Tools

1) Process model (engineering methodology) defines results

2) Definition of CM tasks in the project

3) Definition of suitable procedures and rules for performing such tasks

4) Selection and adaptation of suitable tools

   (CM cannot reasonably be handled without IT support in large-scale projects)

5) Fine-tuning of procedures and rules to adapt to tool-based requirements

6) Documentation in the CM plan

Systematic Approach

- Definition of types of configuration objects
- Definition of identification and archiving schemes
- Procedures for entering configuration objects
- Procedures for changing configuration objects
- Regulations for CM access control
- Procedures for the planning and tracking of configurations
- Resources to use (tools,...)

Documentation in the CM plan

Configuration Objects

- Configuration elements
  - "atomic" objects

    e.g.: source file, graphics
  - text objects (ASCII,...), binary objects,...
  - validated builds in i/i projects
- Configurations
  - an object (meaningful unit) created from other configuration objects such as a subsystem, a functions library, a manual (consisting of several chapters), the entire project.
- Derived objects
  - e.g.: object module, executable
- Baselines

In general, all configuration objects will appear in specific versions (variants, revisions)!

Baseline

Set of specifications or work results (architecture, source code, object code, test cases,…) that has been formally reviewed and agreed on, which thereafter serves as basis for further development, and which can be changed only through change control procedures.

(CMMI Dev 1.2)

Documentation of the validation

Archive validated software version with unique identification

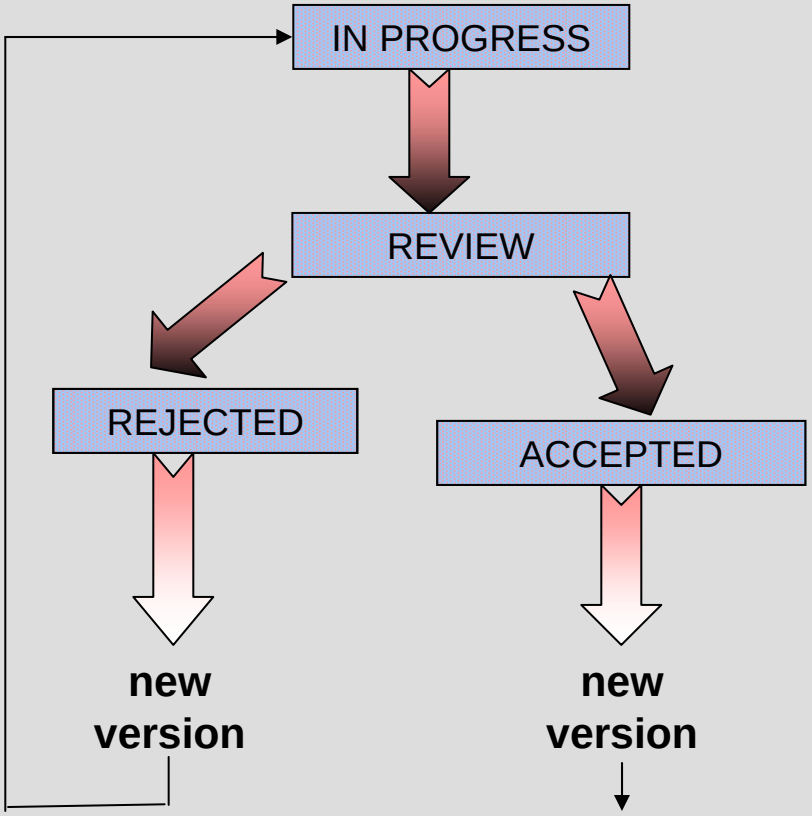
Document validation in a validation report

- Version of the validated software
- What was accepted as it was shown
  What should be changed and in which way should it be changed
- What comes in addition to previous agreements

Validation reports together with the validated software version amend the requirement specification

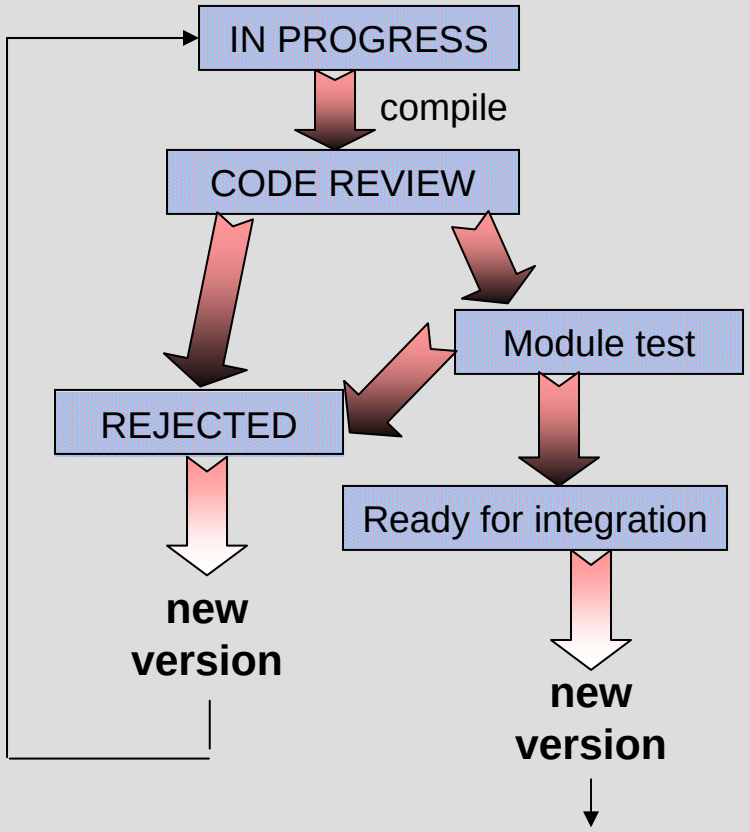
Proposal for a Status Model (1)

Development document



Proposal for a Status Model

Software module



Identification and Archiving Schemes

- Unique identification of objects
  - name
  - classification (object types,...)
  - version identifiers (variants and revisions)
- Traceability / reproducibility
  - CM tool
  - archiving schemes for office folders
  - directory trees (e.g. DOS, UNIX)...

In general, identification and archiving schemes tend to be different for different types of objects (different storage media,...)

Procedures (store, retrieve, change,...)

- Access control
  - access protection
  - user roles and rights (read only, write)
  - ...
- Coordination of teamwork
  - avoiding conflicts and obstacles
    (lock when you check out for change)
  - communication
  - ...

Rely on proven concepts

Procedures

- ensure that only planned objects and new versions of objects can be entered into the CM system
- ensure that only objects having the required degree of maturity can be reused (demand and store review or test reports)
- store notification of change
- ensure that the correct versions of objects are being used
- "parallel" / "serial" developments
- ensure that the backup and archiving concept is satisfactory

Terms

Because of different consequences for project distinguish

- Change Request (CR)
  - Change of requirement specification (also additional wishes)
- Fault Report (FR)
  - Deviation from requirement specification (functional fault, lack of performance,...)
- Modification Request
  - Generic term for CR and FR
- Problem Report
  - Report from customer (fault, error in manual, mistake of customer, wishes,...)

Procedure for Handling Changes

Typical steps (example):

	Step	Authorized
1)	Submitting a report	Integration, customer
2)	Diagnosing the report	Diagnostics unit, developers
3)	Deciding on the change	Change control board (CCB)
4)	Planning the change	Sales / marketing, developers
5)	Performing the change	Development
6)	Delivering the change	Development, integration
7)	Testing the change	Integration, production
8)	Integrating the change	Integration
9)	Releasing the change	Sales / marketing system service

Planning of a change

- Definition of a Change Set
  several fault reports and change request
- Define which objects have to be changed (new versions)
- Status tracking of the whole Change Set
- Separate lifecycle for change requests and fault reports

consistent configuration

Production

- Required objects with version identifier

  (parts list)

- Production instructions

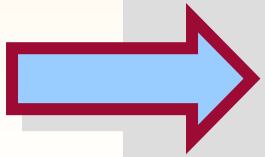
- Dependencies

  source dependencies (explicit and implicit)

  sequence of production steps, objects to use, parameters

  tools (compilers, linkers, preprocessors, filters...)
  incl. unique tool version

- Based on rules as far as possible



Production should be reproducible, hence automate as far as possible !

Tools Requirements (Selection)

- Management of both text files and binary files
- Revisions, variants
  - lock mechanisms, merge support
- Meta data
  - description of change
  - relationships, dependencies between objects
- Reproduction of any given version and its associated meta data
- Configurability / adaptability
  - compliance with / implementation of process model
- Integratability in development environment
- Supported HW and SW platforms
- Network ability
  - several different development locations

Requirements and priorities are project-specific !

Problems Encountered with CM in Projects

- Tackled too late
  When the first objects to manage appear, the CM system must already be fully functional
- Responsibility for CM not defined
- Effort underestimated
- Lacking reference to engineering methodology
- Project starts "from scratch"
  Disproportionate effort and delay (even in smaller projects)
- High evaluation and adaptation effort for tooling
  Time spent, costs incurred