



Introduction to Baselines

Mini-Tutorial European SEPG 2009

Ralf Kneuper

`ralf@kneuper.de`

`www.kneuper.de`

Dr. KNEUPER

Dipl.-Mathematiker, Univ. of Bonn

PhD Computing Science, Univ. of Manchester

1989-1995: Software AG

- Quality assurance, quality management, ISO 9000

1995-2005: Deutsche Bahn/TLC/DB Systems

- Senior consultant, project lead
- Quality management, internal CMM(I) consultant, development processes, project management

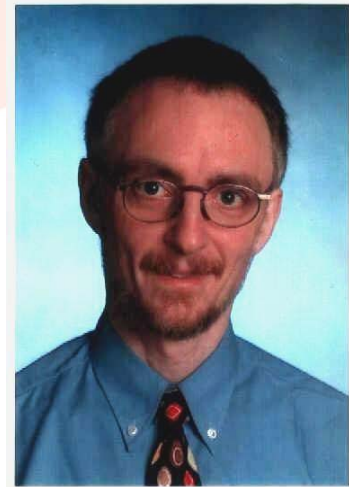
Since 2003: Independent consultant on CMMI

- Process improvement consulting
- Appraisals
- Training

Former speaker of GI (German Informatics society) SIG on software processes

SEI-certified CMM Lead Assessor, CMMI Lead Appraiser

Coordinator of the German CMM(I) Lead Appraiser and Instructor Board (CLIB)



Dr. KNEUPER

Baselines

- are a core concept of configuration management
- are difficult to understand and therefore often misinterpreted

Learner outcomes

- understanding of the properties that characterize a baseline
- understanding of typical situations when to create a baseline
- understanding the main steps needed to create a baseline

- What is a baseline
 - incl. exercise
- When to create a baseline
- Examples
- How to release a baseline
- Common misconceptions
- Building baselines
 - incl. exercise
- Changing baseline content

A large, semi-transparent grey watermark of the text "Dr. KNEUPER" is oriented vertically on the right side of the slide. A large, light orange arrow graphic points from the left towards the watermark.

Why baselines?



- Assume that you are a developer responsible for implementing a certain functionality using a defined architecture for your project
- So what do you expect when you start work?
 - The functionality is clear and there will be no changes – or at least you will be informed and hopefully asked if a change is needed
 - The architecture is defined and will stay stable – again at least you will be informed if there is a change
 - The architecture will support the functionality to be implemented
- Put differently, you expect the requirements, architecture and probably other work product to be **baselined**

A baseline is

- a set of specifications or work products
- that has been formally reviewed and agreed on,
- which thereafter serves as the basis for further development,
- and which can be changed only through change control procedures.

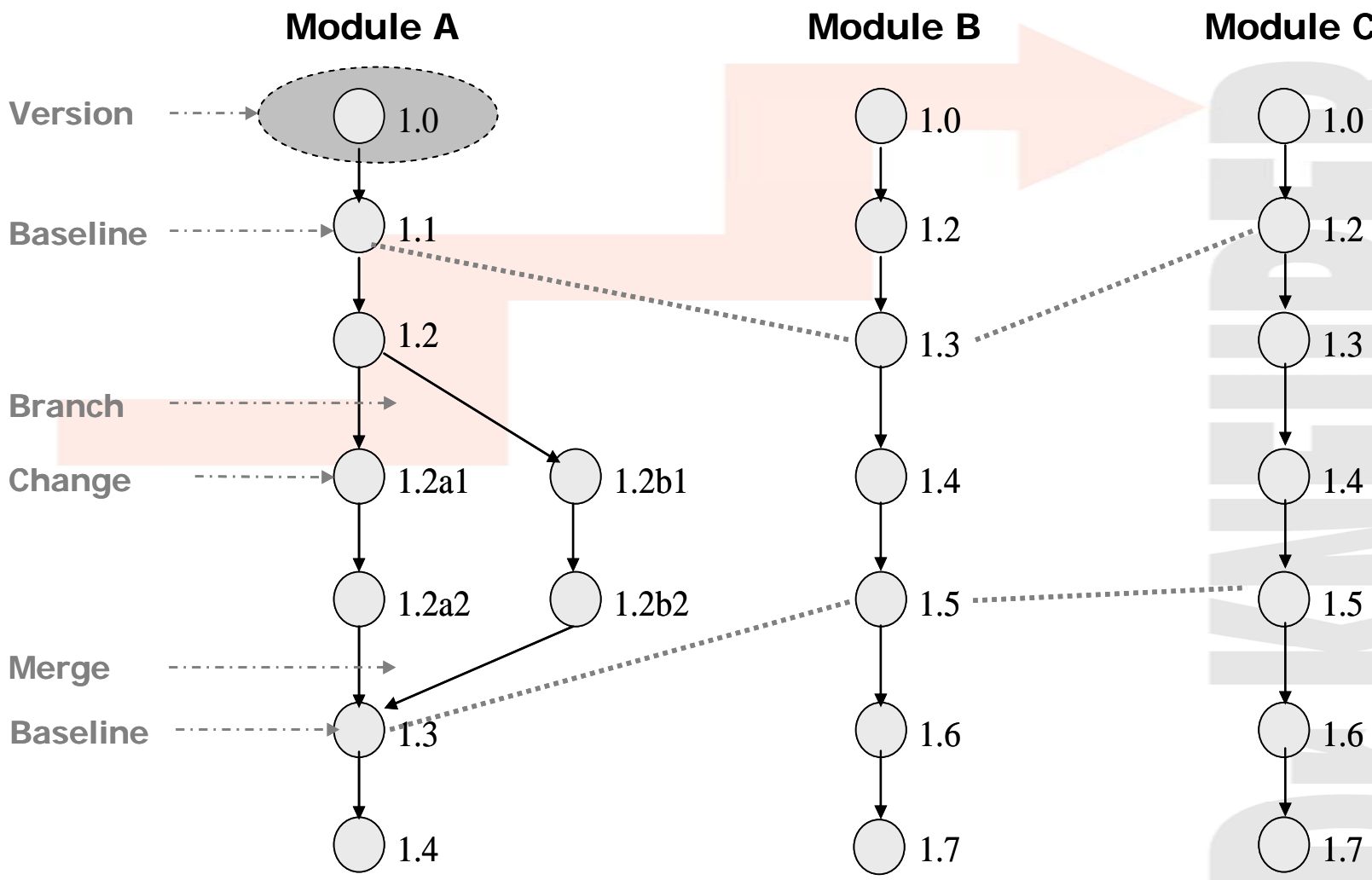
At first this sounds like rather boring CMMI-speak, but let's look again:

Dr. KNEUPER

A baseline is

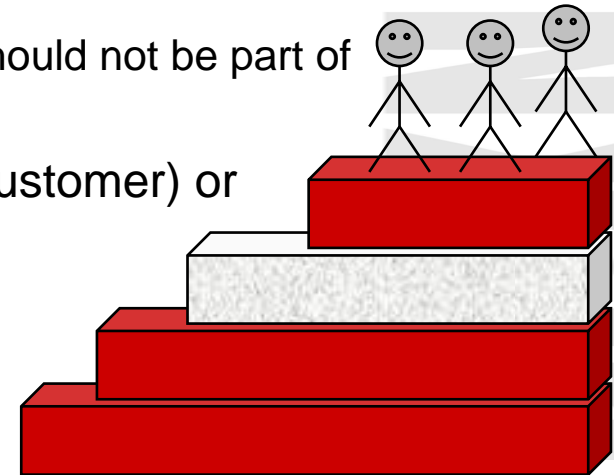
- a set of specifications or work products
 - could be anything that we created or that was provided to us (as a specification) provided that it is still relevant for future work
- that has been formally reviewed and agreed on,
 - we want to be sure that the baseline forms a sound basis for future work and is not just a collection of results that happens to exist at a certain point in time
- which thereafter serves as the basis for further development,
 - the core of the concept of baselines
- and which can be changed only through change control procedures
 - since otherwise we could not rely on it in our further work

Terms in Configuration Management



The definition implies that a baseline

- is planned
- contains results that are considered to be completed
- is reviewed for consistency, correctness and completeness
- summarizes results that belong together and is released as a single entity
- is put under change control and must be kept consistent
 - therefore files that are not to be maintained should not be part of the baseline
- can be created for external (delivery to the customer) or internal use



Get together in groups of 3-5 participants

Discuss the following questions (10 min)

Questions

- When should a baseline be built?
- When do you build a baseline in your company?

Summarize and present the results (1 min per group)

Dr. KNEUPER

When to Create a Baseline

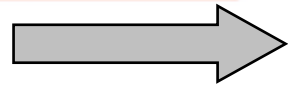


Project Planning, SP 1.3



Project lifecycle phases

Project Monitoring and Control, SP 1.7



Milestone review



Baseline

Building a baseline is event-driven
(completion of relevant results/
milestone), not date-driven



Dr. KNEUPER

Example 1

Requirements and Planning Baseline

A typical requirements/planning baseline

- is created at the end of the initial requirements / planning phase
- contains the requirements documents plus the plans based on these requirements
- is then checked for correctness, completeness, consistency etc., for example
 - does the plan contain all accepted requirements but not any additional requirements?
 - is the effort planned in the QM plan consistent with the resources planned for these activities?
- is released as a single entity (the baseline)
 - for an individual component to be included in the baseline, it may have to be released individually first.
- forms the basis for the next steps such as the design phase
 - for example, the design is created based upon the requirements baseline.

Dr. KNEUPER

Example 2

Baselines in an Agile World



A typical baseline in an agile / iterative development effort

- is created at the end of an iteration (increment, sprint ...)
- includes those results of the iteration that will still be needed after the iteration is completed

Dr. KNEUPER

Example 3

Baselines may be Needed at Different Levels

- Development project
 - develops major new functionality for product
- Change request (CR)
 - minor product improvements and bug fixes
 - implemented as part of a maintenance effort outside the development projects
 - difficulty: CRs and projects may change the same code base in parallel
- Product
 - changed by projects and change requests
 - includes those results of projects and change requests that are relevant for the longer-term handling of the product
 - code
 - user documentation
 - design documentation
 - regression test cases

Example 4 Part 1

Quarterly releases of projects and individual CRs

Release definition baseline

- Initial baseline received from customer
- Main contents:
 - list of CRs including CRS and projects

Planning baseline

- Overall planning baseline
 - revised list of CRs and projects
 - includes sub-baselines:
 - project planning baselines
 - CR planning baselines

System Requirements Specification (SRS) baseline

- one or more baselines per project
- one baseline per CR
- approval of SRS = acceptance of baseline

Example 4 Part 2

Quarterly releases of projects and individual CRs

Development baseline

- one baseline per project or CR
- includes
 - design
 - code
 - test plan
- baseline is delivered to Functional Acceptance Test

Completed FAT baseline

- Note: FAT report is part of the milestone but not of the baseline since it is not a basis of further work
- baseline includes
 - revised code
 - user documentation

Ready for production baseline

- baseline includes
 - revised code
 - user documentation
 - acceptance form

Releasing a Baseline

In order to release a baseline, it must be reviewed and accepted. Typical questions to ask:

- Is the baseline complete?
- Have the individual work products that form the baseline been individually reviewed and accepted?
- Are the individual parts of the baseline consistent?
 - As far as consistency with requirements is concerned, this may build upon the bidirectional traceability of requirements
 - Do different components use the same version of the interface definition?
- Are all bug fixes that were developed in parallel to the project merged into the final project results?

Reviewing a baseline does not imply that any checks need to be performed again that have already been performed

- The review of the baseline may consist of just verifying that all defined checks have been performed on the correct version

SP 3.2 Perform configuration audits to maintain integrity of the configuration baselines

- Confirm that baselines and documentation conform to a specified standard or requirement

Functional configuration audit:

- Verify that the as-tested functional characteristics of a configuration item have achieved the requirements specified in its functional baseline documentation and that the operational and support documentation is complete and satisfactory.

Physical configuration audit:

- Verify that the as-built configuration item conforms to the technical documentation that defines it.

Configuration management audit:

- Audits conducted to confirm that configuration management records and configuration items are complete, consistent, and accurate

So what do I need to check before releasing a baseline?

It depends ...

- on the issues that may go wrong in building the baseline
- on what I want to do with the baseline later
- on the work products that go into the baseline and their relationships
- on the risks involved
- ...
- see PPQA, VER and VAL

Some common misconceptions about baselines (1)

A baseline is created by archiving the current directory content or by creating a simple tag on the current content of the CM tool

- Such a tag would include temporary work products that should not be put under change control
- Some of the work products tagged will not form the basis for further work
- Very useful approach for other purposes (backups) but not for baselines

If results are spread over several technical platforms (e.g. documents, UML models and code, or different coding platforms), it is not possible to create a common baseline



- Even though it may not be possible to build the baseline in a single tool, a logical baseline can be created which combines several technical baselines on different platforms
- In such an environment it is even more important to clearly define baselines in order to know which results in the different platforms belong together

Dr. KNEUPER

Baselines are only created at the end of the project, consisting of the results to be delivered to the customer



Dr. KNEUPER

- Important baseline but certainly not the only one
- without baselines during project lifetime earlier results may be changed adhoc, causing a lot of rework (if noticed) or inconsistencies and bugs (if not notices in time)

A baseline review consists of testing the baseline

- in that case, what would happen to early baselines, before any code is written?
- testing will not provide all the information needed
- reviews and automated analysis provide additional information



Dr. KNEUPER

Exercise 2

Building baselines

Get together in groups of 3-5 participants

Select 1-2 technical environments that are relevant in your work

- for example Java with Subversion, Cobol on mainframe, MS Office documents without CM tool

Discuss the following question (10 min)

Question:

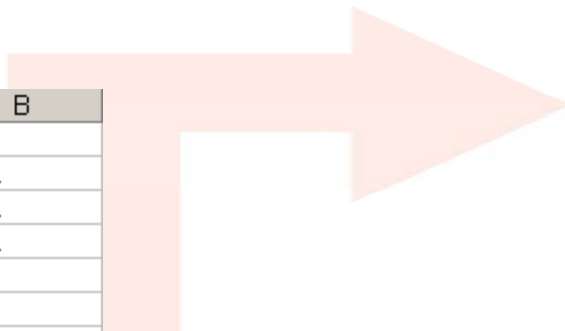
- How can you build a baseline in this environment?

Summarize and present the results (1 min per group)

Dr. KNEUPER

Building Baselines (1)

- A simple solution:



	A	B
1	Baseline 1:	
2	Work product 1a	C:\...
3	Work product 1b	C:\...
4	Work product 1c	C:\...
5	...	
6	Baseline 2:	
7	...	
8		

- Such a manual solution will be adequate for baselines consisting of a small number of results, in particular document baselines
- Source code baselines will typically require tool support

Dr. KNEUPER

Building Baselines (2)

The Standard Solution in Subversion

- In Subversion, a baseline is usually implemented as a tag:
 - perform baseline audit
 - tag repository content = copy „trunk“ to „tags“
- Disadvantage of this approach: the tag contains all the contents of the trunk which may be more than wanted for the baseline
 - If this is the case then manually copy only those work products to „tags“ that are to be part of the baseline



Dr. KNEUPER

- According to the definition, a baseline „ can be changed only through change control procedures”
 - e.g. after the customer requirements have initially been agreed, the customer realizes that an additional functionality will be needed
 - e.g. during the implementation phase it turns out that the design needs to be adapted
 - e.g. during test a number of bugs are found which need to be corrected

Dr. KNEUPER

- When deciding about a request to change a work product that is part of a baseline, the following questions need to be answered:
 - „Usual“ questions about changing this work product (cost, benefit, ...)
 - What other work products in the baseline will need to be changed if we change this item?
- Depending on the baseline affected, different people may have to be involved in adapting the contents of the baseline
 - Typically, these will be the same people that released the initial baseline
- Possible way of expressing the change:
 - A new baseline is built
 - A new version of the baseline is build

Questions?



Baseline

- A set of specifications or work products that has been formally reviewed and agreed on, which thereafter serves as the basis for further development, and which can be changed only through change control procedures.

Configuration Baseline

- The configuration information formally designated at a specific time during a product's or product component's life. Configuration baselines plus approved changes from those baselines constitute the current configuration information.

Product Baseline

- In configuration management, the initial approved technical data package (including, for software, the source code listing) defining a configuration item during the production, operation, maintenance, and logistic support of its lifecycle.