

Lessons Learned in Configuration Management

CMMI Made Practical

29th April 2009 - London

Eric Mariacher – Embedded Software Manager at
Logitech

Configuration management definition

- **IEEE Std-729-1983**

- “Configuration is the process of identifying and defining the items in the system, controlling the change of these items throughout their lifecycle, recording and reporting the status of items and change requests, and verifying the completeness and correctness of items”

Files, Releases and Bugs Management

- **Identification**

- describes the system structure, the nature of its elements, their identity, and gives access to each item version

- **Control**

- organises versions and changes to system items while keeping coherency and consistency on the complete system.

Agenda

- **Configuration Management Questions**
- **Step 0: no file, nor change management tools**
- **Step 1: file management tool only**
- **Step 2: loosely linked file and change management tools**
- **Step 3: strongly linked file and change management tools, file oriented release building**
- **Step 4: strongly linked file and change management tools, change request oriented release building**
- **How CM steps help CMMi « compliance »**

Basic File Management questions

- What change did you make in this file?
- Which version of which files are included in this release?

Basic Change Requests Management questions

- Have we finished implementing this change / bug fix?
- Which bug corrections are included in this release?

Not so Basic Configuration Management questions

- Why was this change made in this file?
- Why do we release a new software version?
- Are we sure we did not forget to integrate a file in this release?
- Are we sure not to wrongly integrate a file that was not intended to be released yet?

Step 0: no file, nor change management tools

- Files back-ups and releases/baselines are managed manually.
- Change Requests are manually managed.

| title | type | description | Priority | Status | Owner |
|-----------------|-----------------|-----------------------------------|-----------------|---------------|--------------|
| CR1 description | bug | This is a short description | high | open | myself |
| CR2 description | Feature request | This is another short description | low | working | Someone else |

Step 0: Answering questions

Basic File Management

- What change did you make in this file?
- Which version of which files are included in this release?

Basic Change Request Management

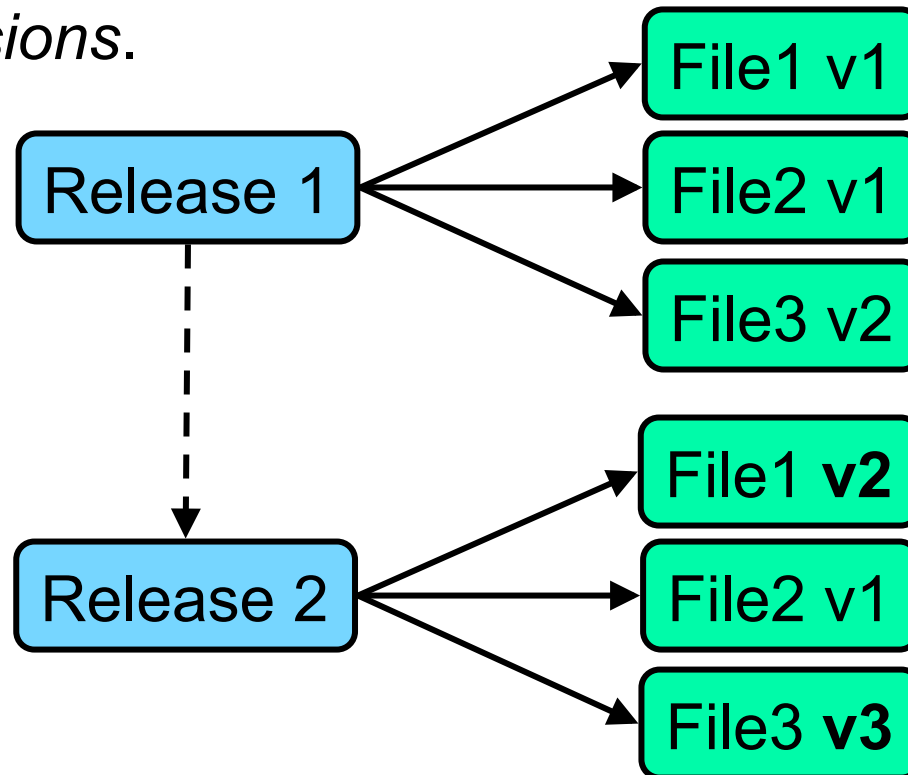
- Have we finished implementing this change / bug fix?
- Which bug corrections are included in this release?

Not so Basic Configuration Management

- Why was this change made in this file?
- Why do we release a new software version?
- Are we sure we did not forget to integrate a file in this release?

Step 1: file management tool only

- Handle file versioning: ability to retrieve older versions.
- Basic release/baseline generation: each release is made of *a list of file versions*.



Step 1: Answering questions

Basic File Management

- What change did you make in this file?
- Which version of which files are included in this release?

Basic Change Request Management

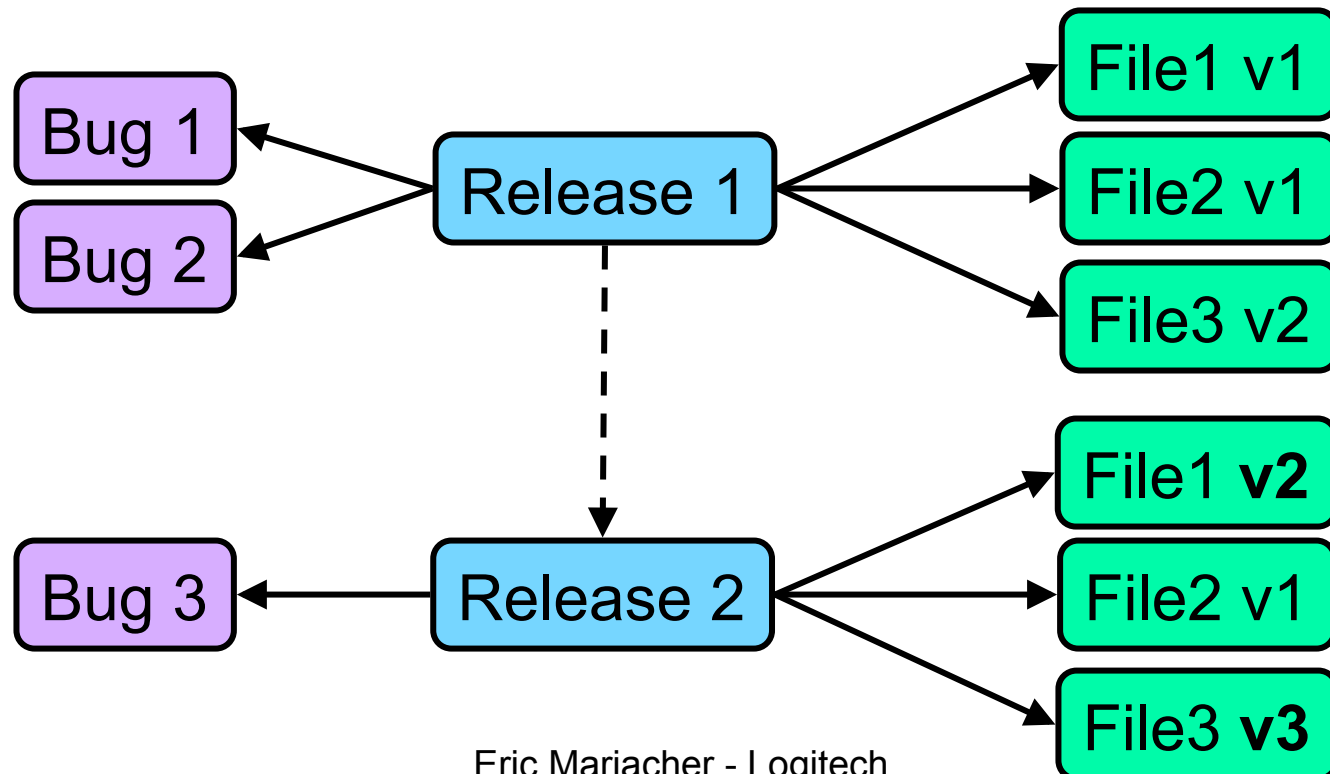
- Have we finished implementing this change / bug fix?
- Which bug corrections are included in this release?

Not so Basic Configuration Management

- Why was this change made in this file?
- Why do we release a new software version?
- Are we sure we did not forget to integrate a file in this release?

Step 2: loosely linked file and change management tools

- *Release note:*
 - *list of file versions* as done in step 1.
 - *list of change requests* based on completion date.



Step 2: Answering questions

Basic File Management

- What change did you make in this file?
- Which version of which files are included in this release?

Basic Change Request Management

- Have we finished implementing this change / bug fix?
- Which bug corrections are included in this release?

Not so Basic Configuration Management

- Why was this change made in this file?
- Why do we release a new software version?
- Are we sure we did not forget to integrate a file in this release?

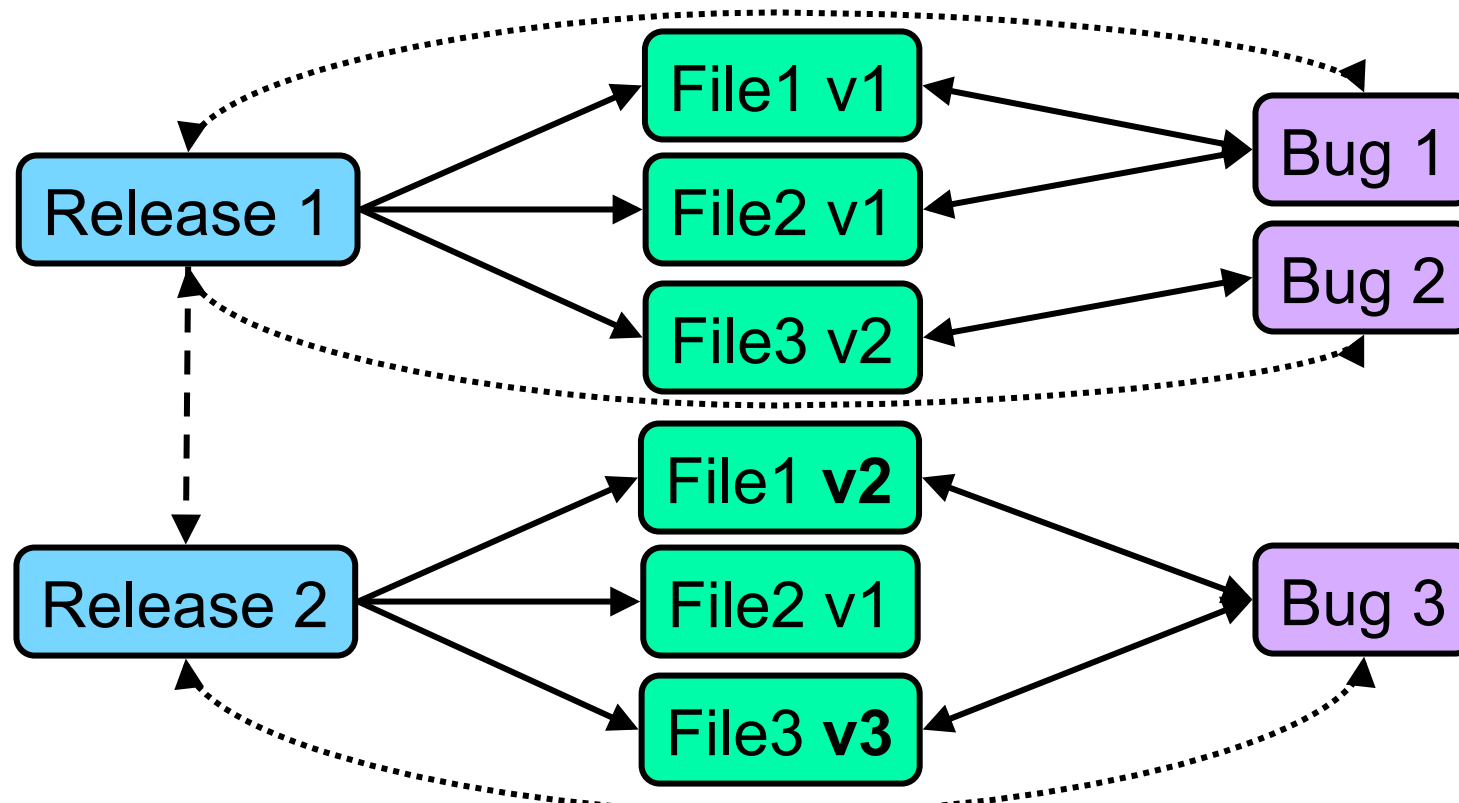
Step 3: strongly linked file and change management tools

- Each time a file check-in is done...
- File management tool **forces**...
- Developer to motivate the check-in
- i.e. **link** the check-in with a change request



Step 3: file oriented release building

- *Release note:*
 - list of file versions as done in step 1.
 - list of change requests based on completion date **and modified files versions.**



Step 3: Answering questions

Basic File Management

- What change did you make in this file?
- Which version of which files are included in this release?

Basic Change Request Management

- Have we finished implementing this change / bug fix?
- Which bug corrections are included in this release?

Not so Basic Configuration Management

- Why was this change made in this file?
- Why do we release a new software version?
- Are we sure we did not forget to integrate a file in this release?

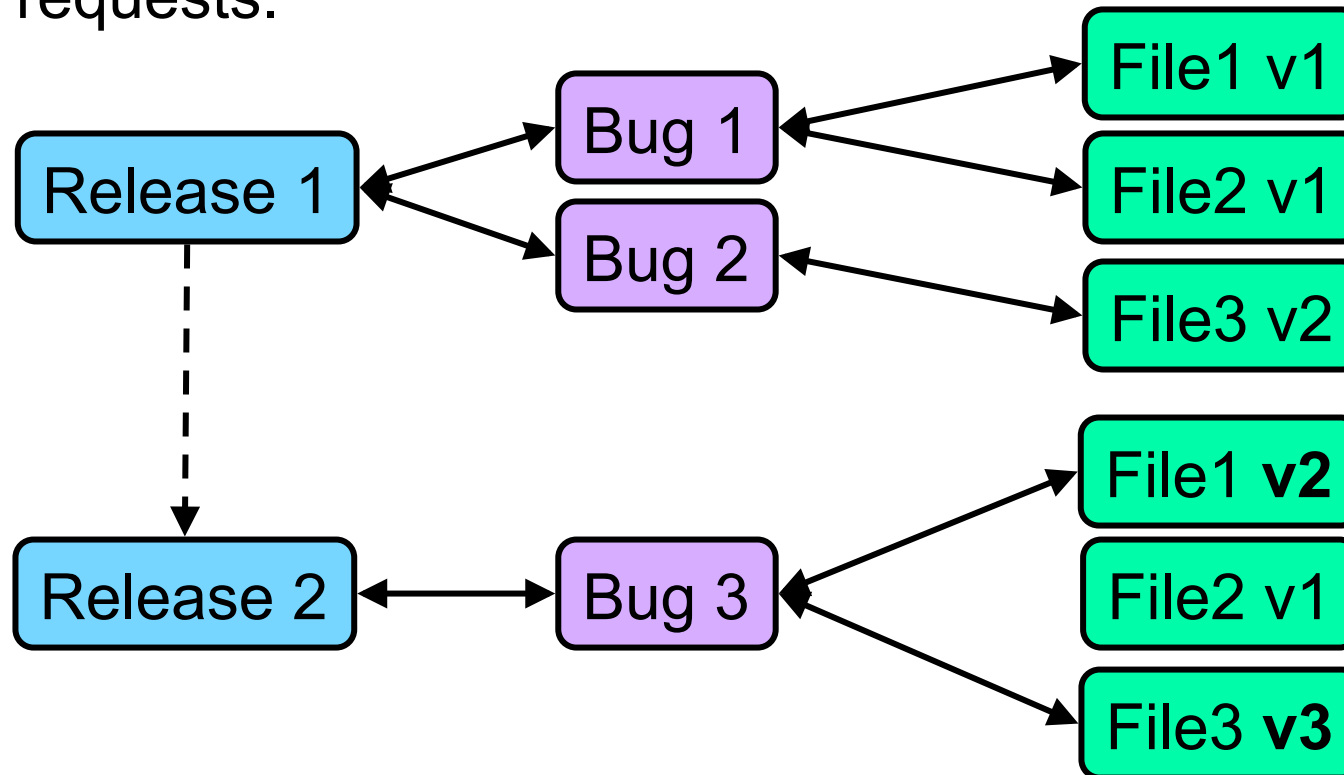
Step 4: strongly linked file and change management tools

- Each time a file check-in is done...
- File management tool **forces**...
- Developer to motivate the check-in
- i.e. **link** the check-in with a change request



Step 4: change request oriented release building

- Release note is a list of change requests.
 - list of files versions based on integrated change requests.



Step 4: Answering questions

Basic File Management

- What change did you make in this file?
- Which version of which files are included in this release?

Basic Change Request Management

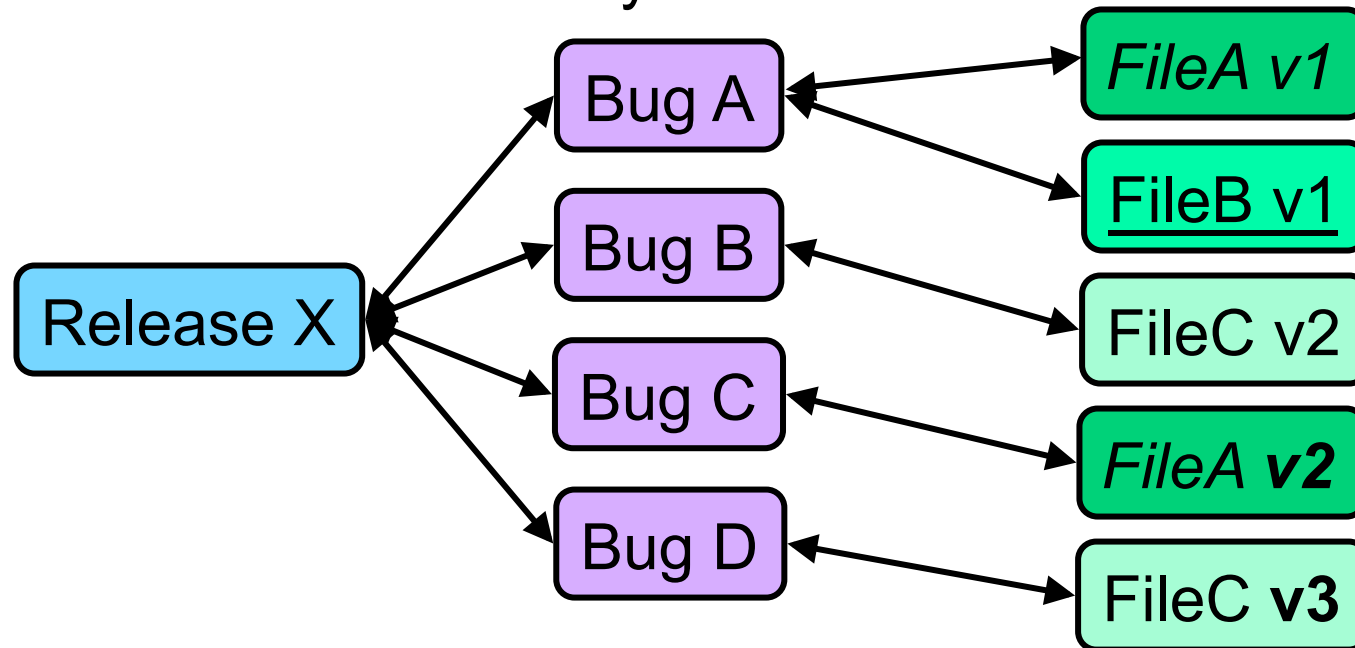
- Have we finished implementing this change / bug fix?
- Which bug corrections are included in this release?

Not so Basic Configuration Management

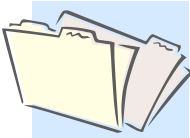

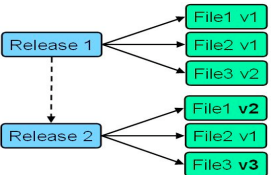


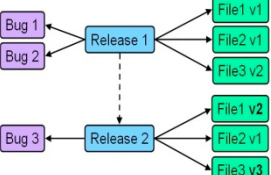


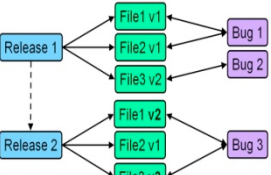



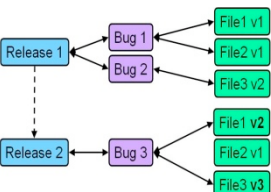



- Why was this change made in this file?
- Why do we release a new software version?
- Are we sure we did not forget to integrate a file in this release?
- Are we sure not to wrongly integrate a file that was not intended to be released yet?

Step 4: change request oriented release building

- Are we sure we did not forget to integrate a file in this release?
- Are we sure not to wrongly integrate a file that was not intended to be released yet?



Size of involved teams

| Step | Files & Bugs | developer | Change Control | tester |
|---|--|---|---|---|
| 0: no file, nor change management |  |  | | |
| 1: file management only |  |  | |  |
| 2: loosely linked file and change management tools |  |  | |  |
| 3: strongly linked tools, file oriented release building |  |  |  |  |
| 4: strongly linked tools, change request oriented release building |  |  |  |  |

Measuring vs CMMi

- How the level of integration between file and bug management tools helps measuring Configuration Management practices vs CMMi

CMMI: Configuration Management Process Area

- **SG 1 Establish Baselines**
 - SP 1.3 Create or Release Baselines
- **SG 2 Track and Control Changes**
 - SP 2.1 Track Change Requests
 - SP 2.2 Control changes to the configuration items
- **SG 3 Integrity of baselines is established and maintained**
 - SP 3.1 Establish and maintain records describing configuration items
 - SP 3.2 Perform configuration audits to maintain integrity of the configuration baselines

SG 1 Establish Baselines

SP 1.3 Create or Release Baselines

- ✓ *SubPractice 2.* Create or release baselines only using the configuration management system

- ✓ *SubPractice 3.* Document the set of configuration items that are contained in a baseline

- Use a Configuration Management tool (step 1)

- Release Notes (steps 0-3 but easier/more reliable with step 4)

SG 2 Track and Control Changes

SP 2.1 Track Change Requests

- ✓ *SubPractice 2.* Analyze the impact of changes proposed

- ✓ *SubPractice 3.* Review change requests and get agreement from stakeholders

- Link file check-ins with bugs (step 2)

- List of bugs to integrate and impacts (step 2)
- A new role : CCB (step 4)

SG 2 Track and Control Changes

SP 2.2 Control changes to the configuration items

- ✓ *SubPractice 3.* do Check ins and check outs in a manner that maintains the integrity of the configuration items

- ✓ *SubPractice 4.* Perform reviews to ensure that changes have not caused unintended effects on the baselines

- ✓ *SubPractice 5.* Record changes to configuration items and the reasons for the changes as appropriate

- (steps 1-3 but easier/more reliable with step 4)

- Integrate all the bugs and only the bugs (step 4)

- Link file check-ins with bugs (step 3)

SG 3 Integrity of baselines is established and maintained

SP 3.1 Establish and maintain records describing configuration items

- ✓ *Typical Work Product 1.* Revision history of configuration items
- ✓ *Typical Work Product 5.* Differences between baselines

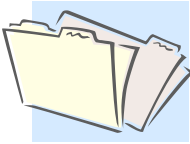

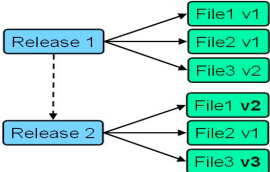


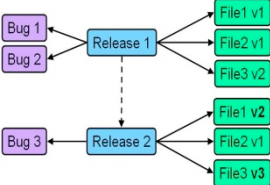


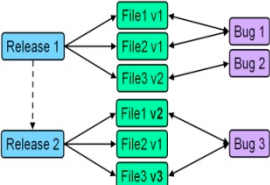



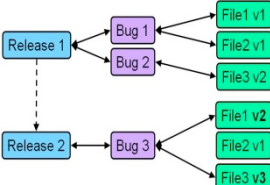



SP 3.2 Perform configuration audits to maintain integrity of the configuration baselines

- (steps 1-3 files)
- (step 4 bugs and files)
- The higher the step is achieved, the higher the accuracy is.

Conclusions

- Better integration between file and bug management tools means easier and more reliable CMMi Configuration Management measures.
- BUT
- implies « unnecessary » overhead for small projects when organization is not supporting CMMi

Size of involved teams

| Step | Files & Bugs | developer | Change Control | tester |
|---|--|---|---|---|
| 0: no file, nor change management |  |  | | |
| 1: file management only |  |  | |  |
| 2: loosely linked file and change management tools |  |  | |  |
| 3: strongly linked tools, file oriented release building |  |  |  |  |
| 4: strongly linked tools, change request oriented release building |  |  |  |  |

Step 5: releases planning?

