R 1 : allowing pluggable scheduler

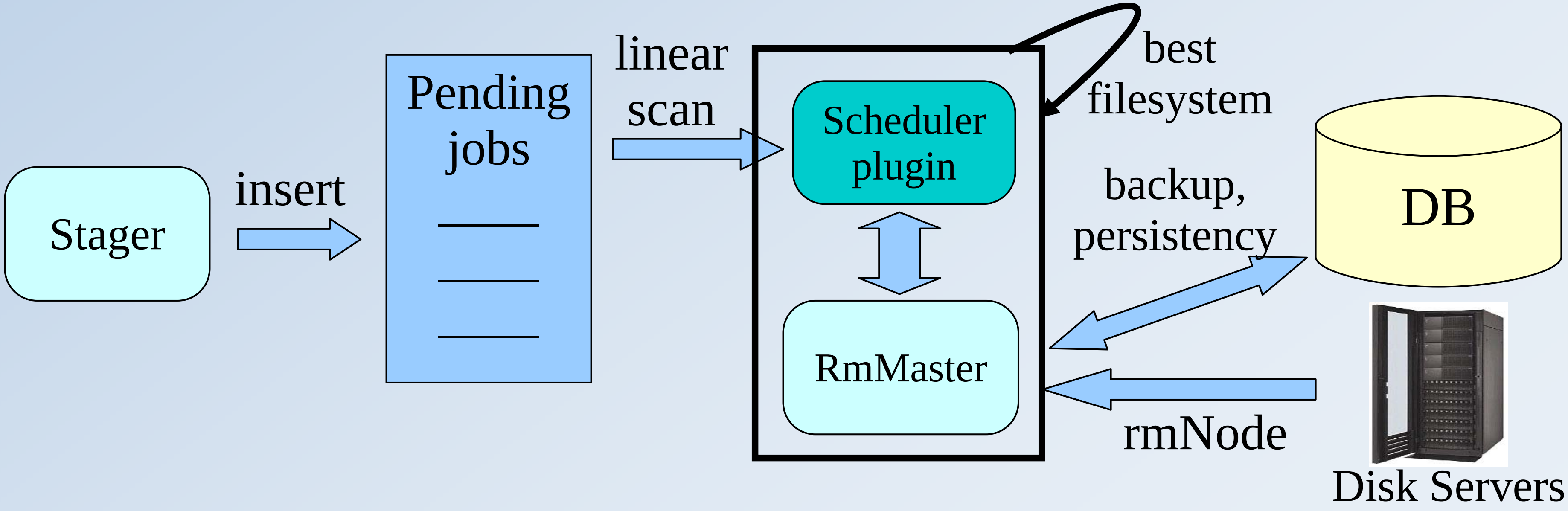
“without enforcing a strong binding to LSF”

- MAUI binding was ported to latest revisions
  - Credits to Russian contributors
  - Used by them in production (small Tier2)
  - Very limited scalability unless we switch to moab

R 1 : allowing pluggable scheduler

"until clear positive results about LSF scaling are obtained"

- LSF plugin rewrite is going on
  - New architecture for the monitoring and the LSF plugin
  - Based on shared memory with new RmMaster
  - No connection to the Database for FS choice (bottleneck)

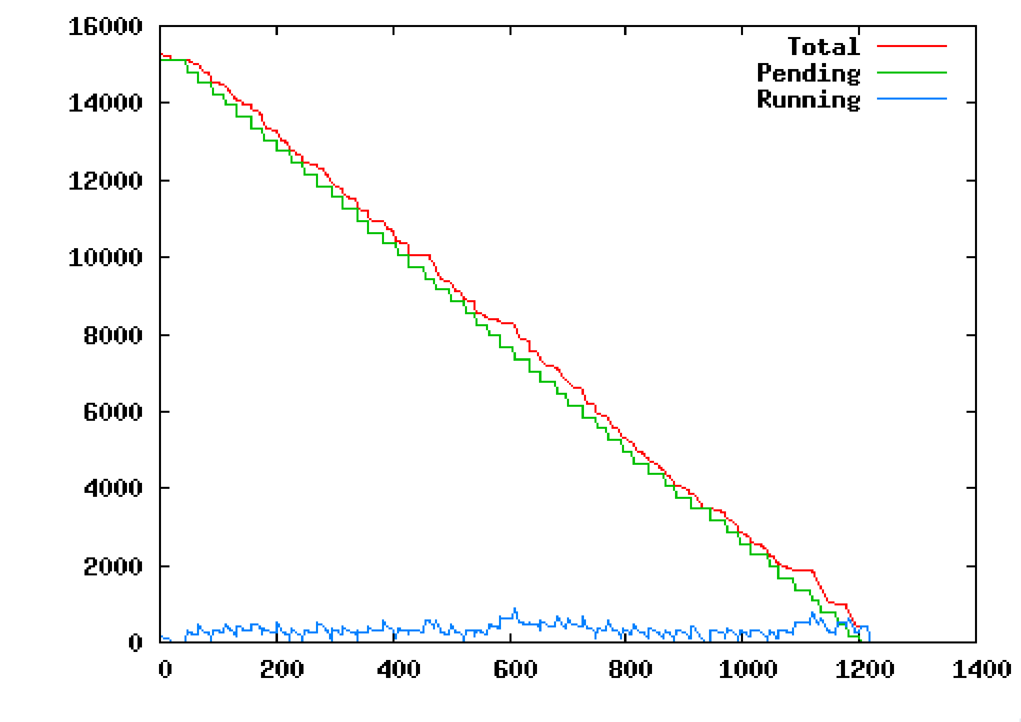


R1 : What will improve

- LSF
  - plugin limitations gone
    - expecting > 100K queuing jobs is fine. To be measured
  - code better supported now as we understand it
  - improvements in the scheduling made possible
    - limiting number of retries
    - returning proper errors from the plugin
- RmMaster, RmNode
  - fully rewritten (monitoring part), so better supported
  - RmMaster made almost “stateless”
    - DB sync allows to restart after a crash
    - machine states not lost

R1 : Prototype testing

- Scalability already tested
  - Proto sustained 50K queued requests (did not test more)
  - Final release expected to sustain >100K



R1 : Current status

- Shared memory
  - Integrated into the CASTOR framework
  - New plugin implemented and tested
- New monitoring
  - RmMasterDeamon and RmNode daemons implemented
  - Currently under tests
  - ORACLE updates missing
- Command lines
  - rmGetNodes implemented
  - rmAdminNodes to be done
    - but protocol and server part are already there
    - only the client part is missing
- Production ready 1st quarter of next year ?

R 2 : XRoot protocol

“to be supported by CASTOR 2 as an internal protocol”

- XRoot has been integrated as an internal protocol
  - XRoot will only run on dedicated pools
  - XRoot will be responsible for load balancing
  - The daemons run independently of CASTOR
  - Modifications integrated since CASTOR 2.1.1.0
- Still under tests
  - Reading part has been successfully tested
  - Writing part still under testing
- CASTOR expectations
  - Solution for analysis facilities
  - Low foot print of the protocol on the diskservers

R 2 : XRoot protocol

“the development, integration and support model must be agreed with XRoot developers”

- Agreed that development, integration and support are provided by the XRoot team
- The CASTOR team has only a consultancy role

- Progress were very slow since June. Commitment from the XRoot team is low as they have been given other priorities
- May become a problem if CASTOR relies on it for e.g. the CERN Analysis Facility
- SLAC may insure long term support

R 3 :Process and technologies

“the committee is happy to endorse the process and technologies adopted which clearly compared favourably to those in use in other software projects”

“The CASTOR team is grateful for this endorsement”

- What to say more ?

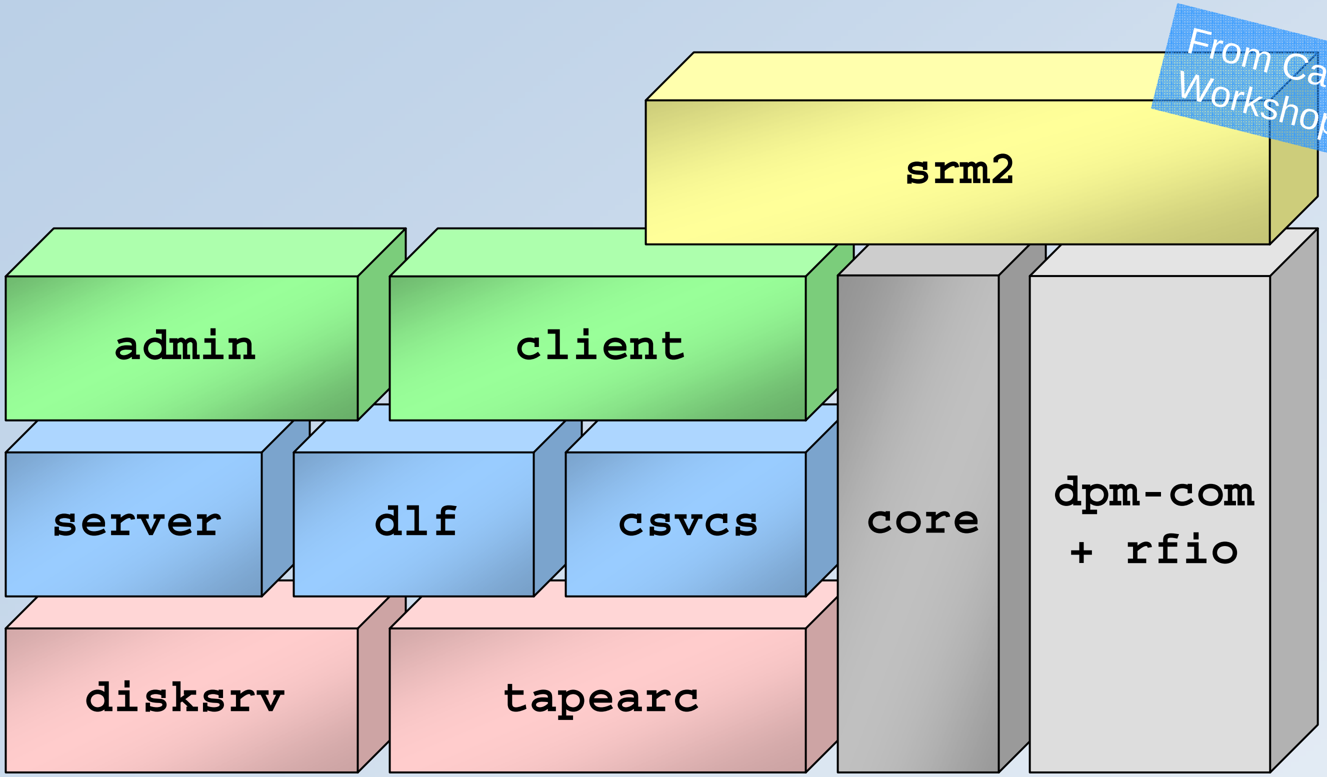
R4 : Builds and releases

- *Consideration should be given to use of new configuration management and build tools in order to better automate the release process and to facilitate support of different platforms.*

Current status

- Review of the build process still underway
  - Identified main components
  - Initial evaluation of build alternatives
- Support for different platforms depends on the build process
- Improved test/release process
- ETICS integration foreseen by Q2 2007

R4 : Castor Components Breakdown



From Castor External Workshop, Nov 2006

- Identified **9** Components
  - `dpm-common` is an external component including rfio clients – we won't maintain it
    - *more later*
  - `admin` is a "simple" component providing tools and scripts for operation
  - The other components are not yet well separated in the CVS codebase
    - Going to be made separated with a clear dependency graph

R4 : Component Structure

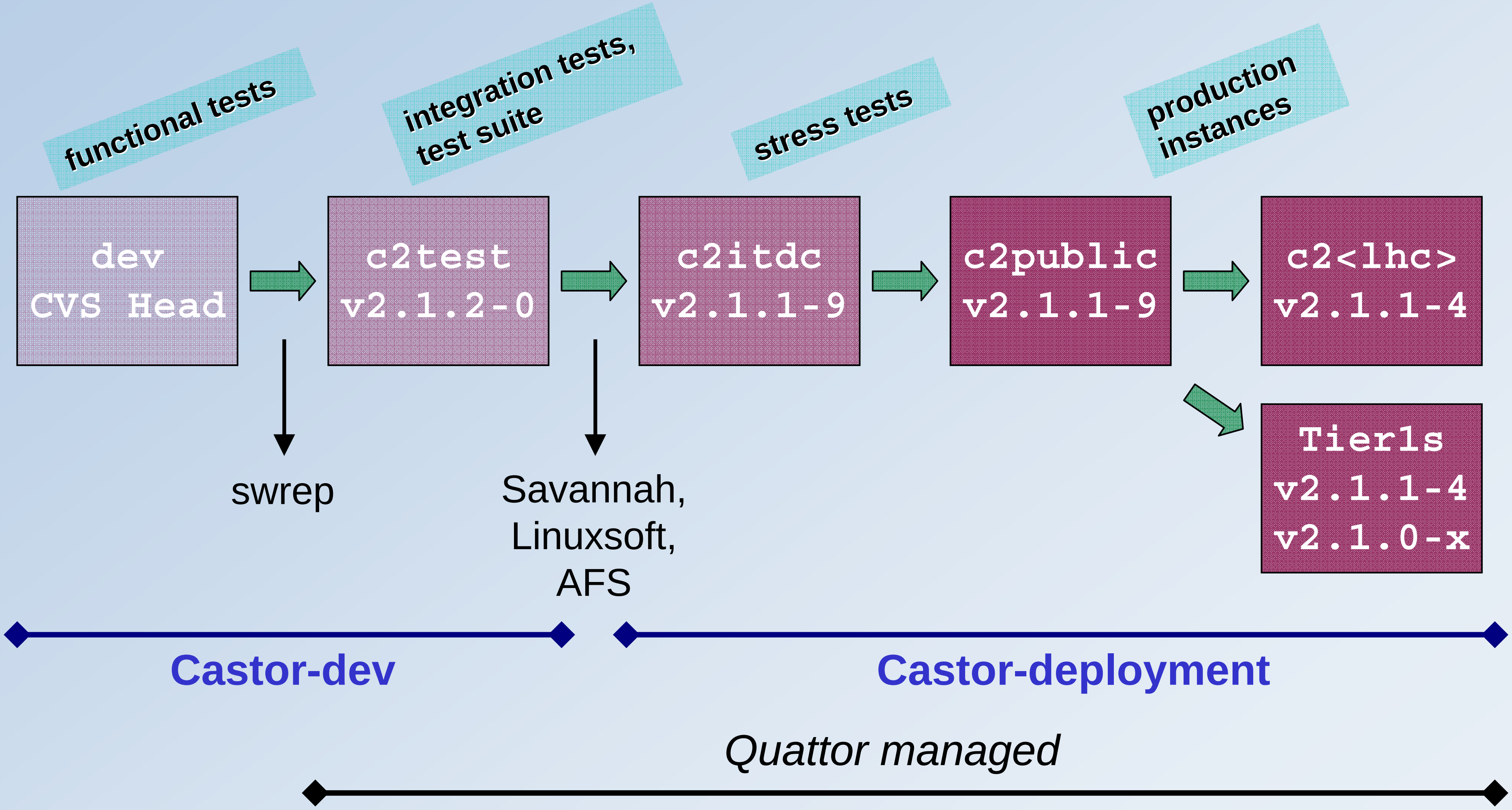
From Castor External Workshop, Nov 2006

- For each Component ≡ main directory in CVS
  - **`cmt`**
    - requirements file and any other package related files for CMT
  - **`src`**
    - source code, with subdirectories for subcomponents, plus compiled **`.o`** files
  - **`h`**
    - public headers, to be shared with other components; all include dirs go to -I flag at build time
  - **`doc`**
    - all available documentation, including UML diagrams for code generation
    - man pages
  - **`test`**
    - src code and makefiles for tests, with subdirectories for subcomponents
  - **`debian`** (for the time being, to be renamed **`config`** later on)
    - all scripts and files needed for packaging the component. New component-related **`changelog`** (debian style) and **`castor-<comp>.spec`** file for RPM build to be maintained here.
  - **`Linux-i686`** *(platform dependent)*
    - all executables and **`.so`** shared libraries. Static libraries not supported anymore
  - **`build`**
    - all packaged elements (**`.deb`**, **`.rpm`**)

R4 : Build process

- Evaluating build alternatives
  - CMT
  - autotools
- Showstoppers and status
  - We set a number of internal requirements, which make the whole task more complex
    - Keep CVS history
    - Adapt the code generator to fit the new layout
    - Make the changes without breaking other developments and releases
  - => Need for process automation
    - Aim to achieve a set of shell scripts to perform the refactoring in one go
    - DPM common project helped in gaining a better knowledge on the code structure and internal dependencies *(see later)*
  - Other priorities prevented this task from being further developed
    - But priority expected to raise (e.g. for support to other platforms)

R4 : From CVS to production



R4 : Tests

- Test suite: **238** tests
  - **25** castor non-rfio cases
  - **21** rfio non-castor cases
  - **12** rfio-castor cases x {**11** TURL variants | **5** env settings}
- Targeting integration tests
  - Aims to validate a release before handing it to the operation team
  - Already found and fixed new bugs
    - Some of them hard to reproduce by hand…
- Code-level unit testing to come
  - Especially for the new C++ code

R4 : Next steps

- Platform support – server side
  - SLC[3-4] / 32bit
    - Fully supported. Servers run (will keep running) on SLC3
  - SLC4 / 64bit (`x86_64`)
    - Ongoing. Will concentrate on mixed deployments.
  - Dropped builds for non-supportable platforms (e.g. `ia64`)
- Platform support – client side
  - MacOS X / Windows expected next year
    - After CVS refactoring
- ETICS
  - Take advantage from experience with Quattor

R 5 : AFS releases

“The new procedure of installing […] CASTOR [...] under the LCG External Library Service […] should continue”

- Part of the release process
- Includes backward compatibility libraries
- Note that it does not mean that old releases will work forever
  - Old clients may be discarded
  - But new ones can then be used by old code
    - Backward compatibility libraries are released for this purpose
- Experience shows a lack of agreement on what backward compatibility is between IT and experiments

R 5 : versioning and compatibility

- Compatibility can be at different levels :
  - API level (Avoids recompilation. May need relink)
  - Library level (Avoids relinking)
  - Protocol level (so that you can talk to new servers with old protocol)
- CASTOR behavior
  - API compatibility
    - When broken, new major release and library soname changes
    - In such a case, old API/library maintained for old clients
  - Dynamic library always backward compatible for a given soname
  - Protocol may change (not yet mature)
    - Libraries (including compatibility ones) are then updated
- Remaining issue
  - Experiments have to go to latest client version in case protocol changes so that they get the latest compatibility library
  - May be solved by a “default” link in the AFS distribution

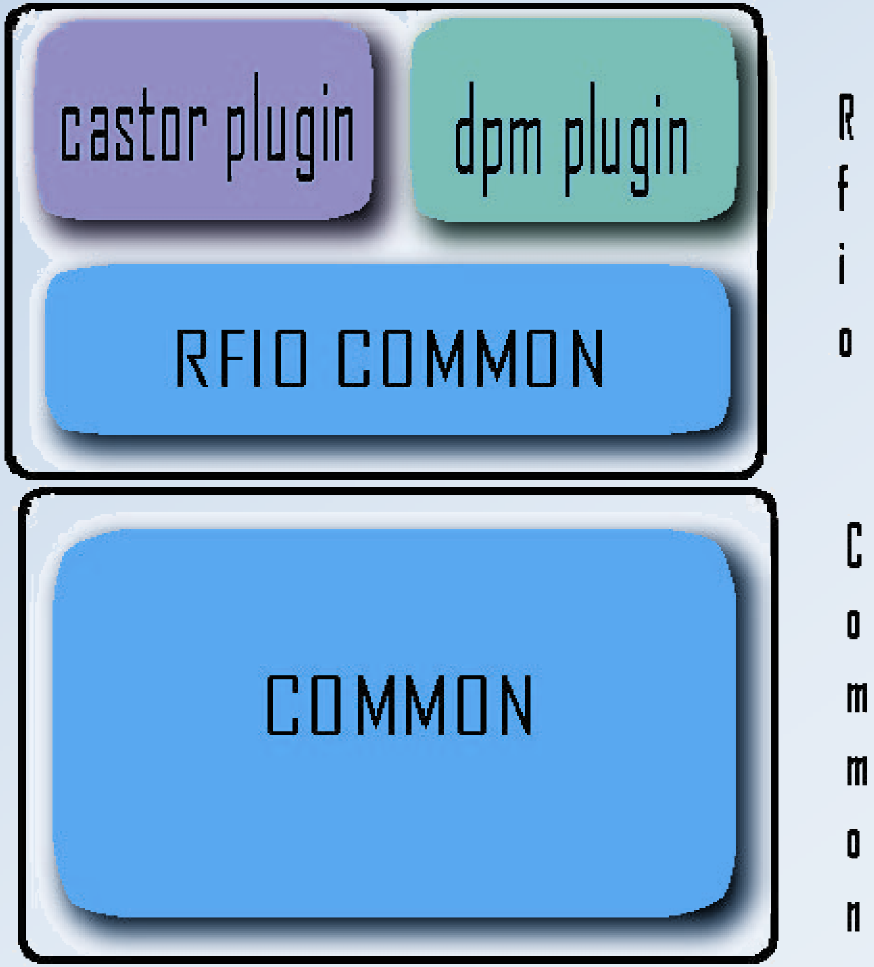
R6 : Castor-DPM collaboration

- *Pursuing common source code management for rfio and the nameserver with the LCG/DPM project seems to be highly desirable.*

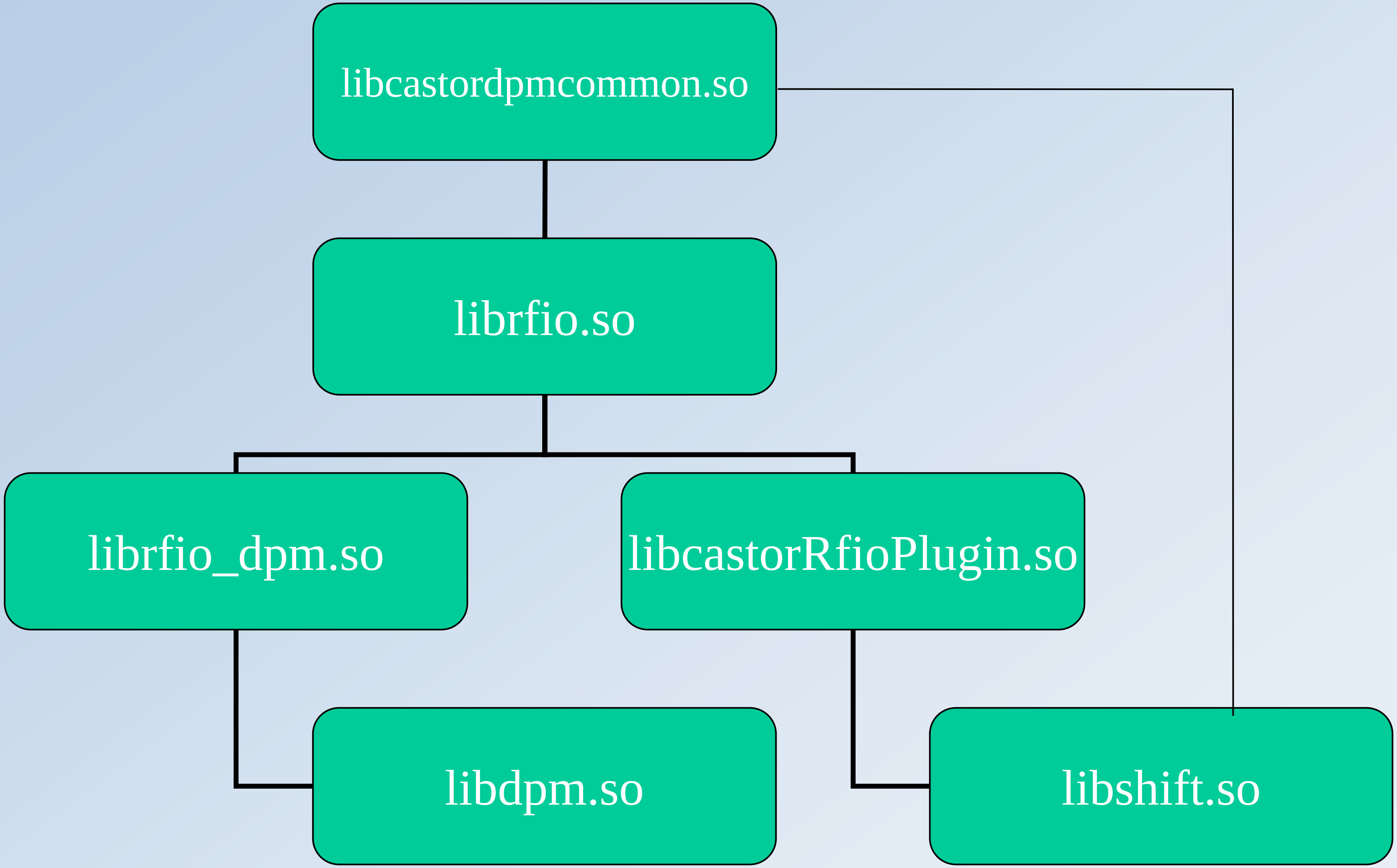
- Current status
  - Common CVS repository for RFIO in place
  - First tests successful (end Sept):
a common RFIO command line talks to both CASTOR and DPM
  - Packaging delayed because of lack of manpower on DPM
    - But now progressing

R6 : Architecture and CVS layout

- Layout chosen by DPM
  - Essentially following basic layout of Castor/DPM
  - RFIO implementation relies on a common infrastructure inherited from CASTOR-1
    - This **common** code is shared among all daemons, hence it’s not strictly RFIO and it’s kept in a separated directory
  - Implementation made dependent on a shared lib loading, according to the given RFIO TURL
- Plugins’ code resides on the relevant CVS
  - To be compiled against either CASTOR or DPM



R6 : Libraries for RFIO and Common



R6 : Packaging

- rfio.rpm
  - bin for the command line, librfio.so, man pages
- castor-dpm-common.rpm
  - libcastordpmcommon.so
  - required by all Castor/DPM modules
- castor-stager-client.rpm
  - will provide castorRfioPlugin.so
  - *the only one we will maintain*
- dpm-rfio-plugin.rpm
- Devel RPMs
  - Provided to build Castor and DPM

R6 : Next steps

- RFIO
  - Testing (new test suite ready)
  - Proper packaging and deployment of our plugin
  - rfio_copy() API

- Extend the collaboration to other modules
  - Csec, i.e. Castor security layer
    - Straightforward, minimal differences already merged in our code
    - Towards a secure interface to Castor
  - Aim for the NameServer

R 8 : ORACLE tuning

“consider that ORACLE tuning be decoupled from storage”

“seems to require significant investment”

- Resources and priorities did not make it possible yet
- ORACLE tuning is done essentially in 2 ways :
  - On the integration and test instances
    - Load is put manually or via new repack
  - By observation of the production instances
- Note that decoupled tuning may be disappointing
  - most of the improvements done were specific cases that would not have been highlighted
  - E.g. : stager_qry tuning for > 10K DiskCopies for a given file due to looping client script

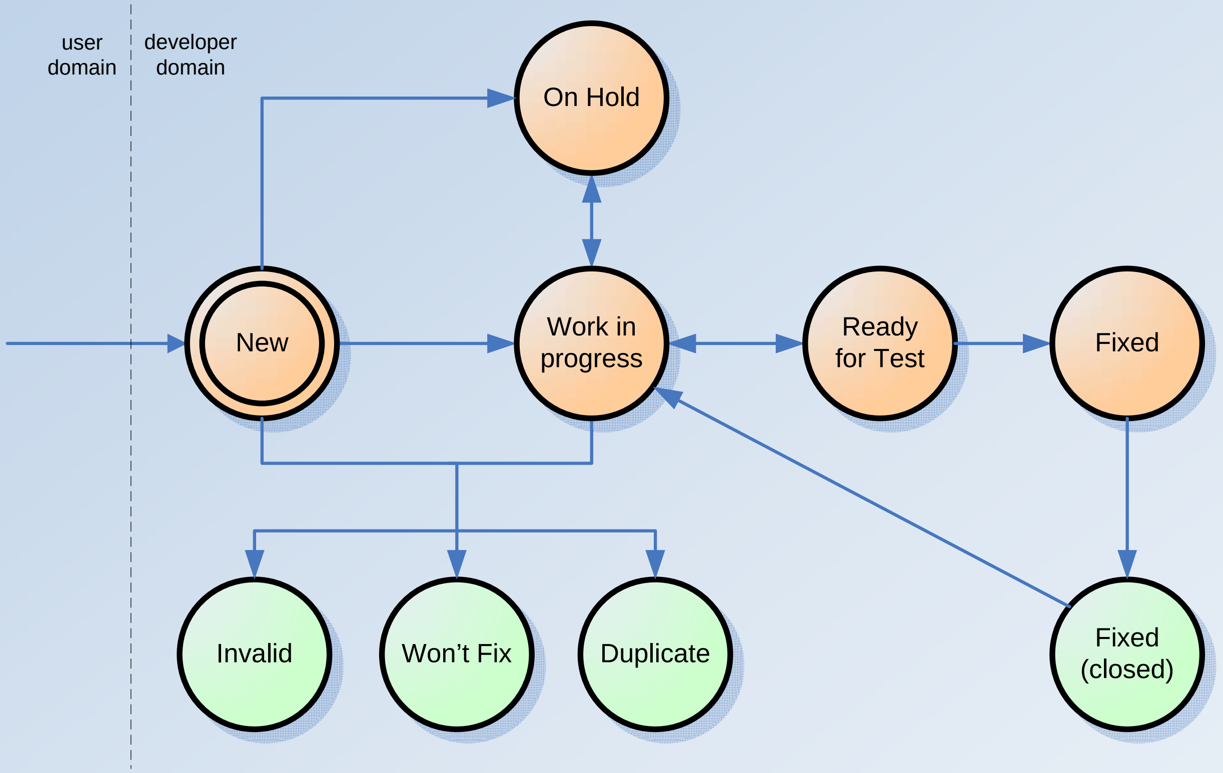
R10 : procedure for bugs tracking

- *The committee recommends that the procedures and tools used to register bugs/problems, and their follow-up, are reviewed and documented to ensure all problems are dealt with in a timely manner.*

- Status:
  A procedure for problems tracking is in place since July
  - Bug tracking and RFEs: savannah bug tracker
  - User problems: remedy flow, with proper escalation
    - Dev team deals with 3rd level support only
    - **Developer on call** for support (rota among the core developers)
  - Tier1 support: mailing list + savannah for RFEs
    - Mostly 3rd level

R10 : Bug tracking and RFEs

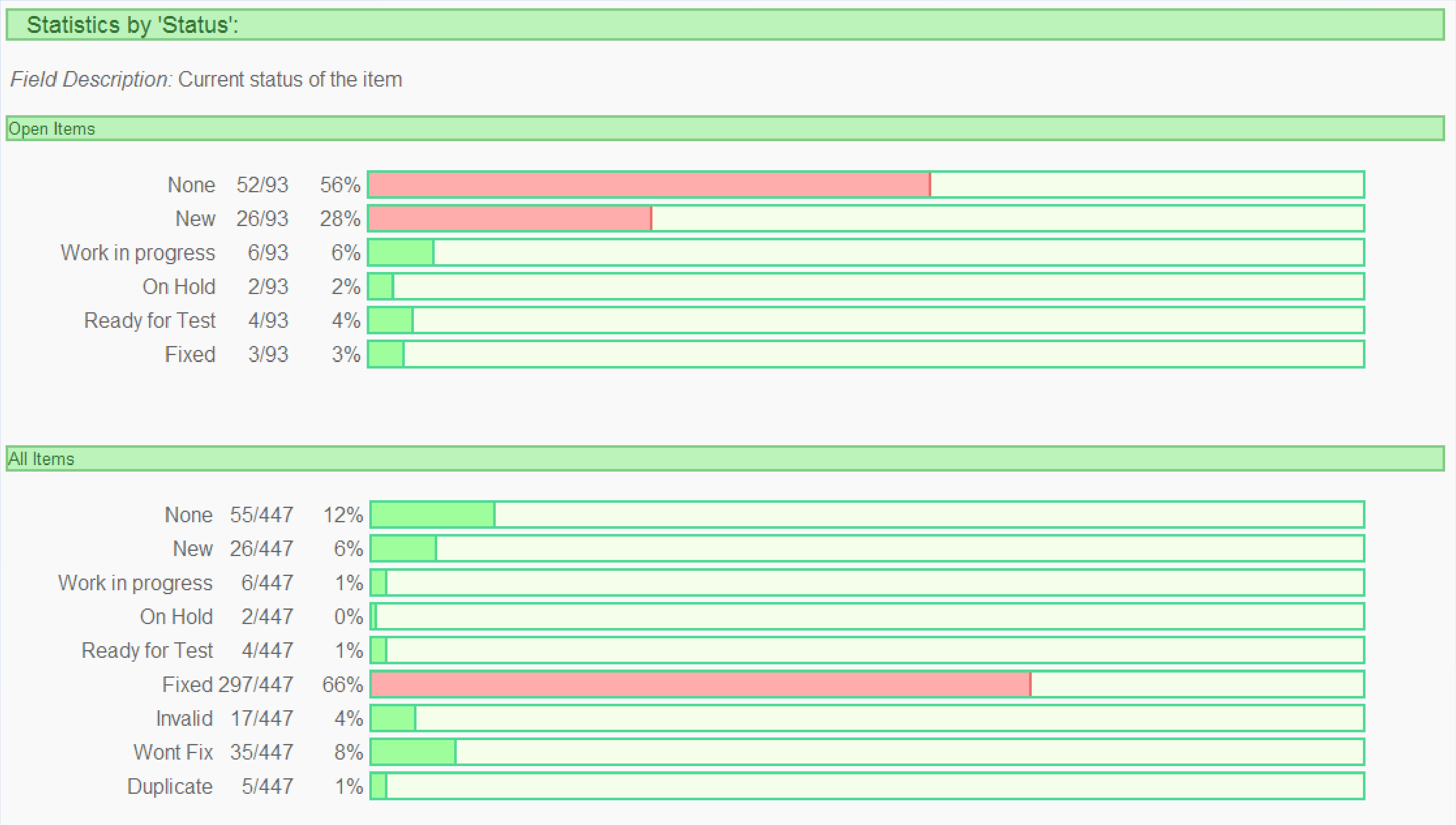
- Bugs are entered following a simple yet strict FSM
  - Allows for easier tracking
  - Automate transitions as much as possible – to achieve the minimal process overhead



From FD Section Meeting, Aug 11, 2006

R10 : Bug tracking and RFEs

- Current statistics



Statistics by 'Status':

Field Description: Current status of the item

Open Items

None	52/93	56%
New	26/93	28%
Work in progress	6/93	6%
On Hold	2/93	2%
Ready for Test	4/93	4%
Fixed	3/93	3%

All Items

None	55/447	12%
New	26/447	6%
Work in progress	6/447	1%
On Hold	2/447	0%
Ready for Test	4/447	1%
Fixed	297/447	66%
Invalid	17/447	4%
Wont Fix	35/447	8%
Duplicate	5/447	1%

R10 : About support

- Savannah for castor-deployment
- Remedy tickets
- The dev on call tries and attack one or more bugs in savannah
  - But the load of the past months made this task not sustainable by a single person
    - Current support load ~1.5 FTE *(see German’s talk)*
  - We’ll have to dedicate more and more effort on bug fixing
    - And many of them get fixed before going to savannah!

R16 : Credible WBS

"provide effort … maintenance of a credible WBS and schedule covering both development and operation tasks"

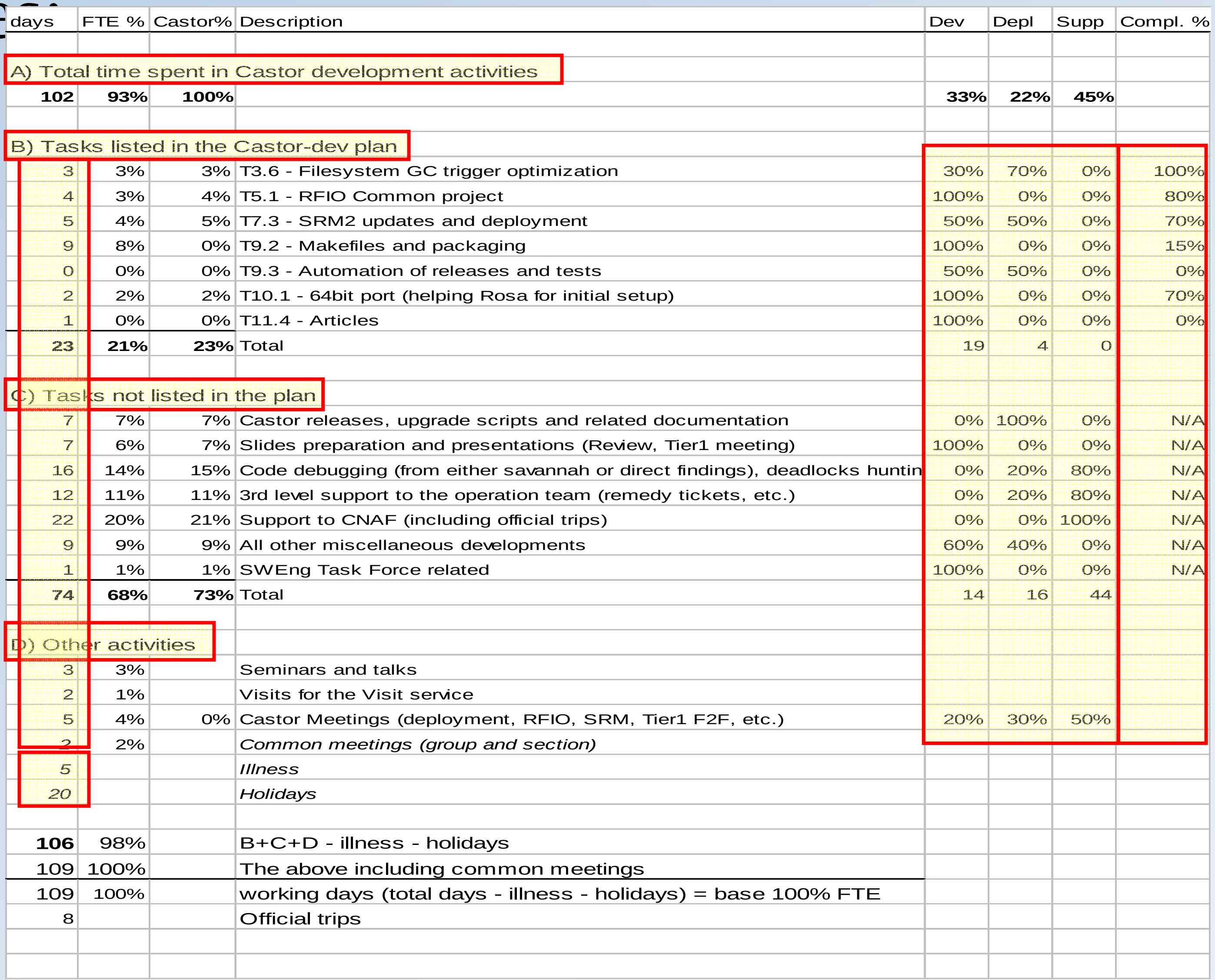
- Castor-dev situation in June 2006:
  - WBS-based planning document recently introduced
    - Task name, description, time estimation, manpower, progress, dependencies, risks+alternatives
  - Only development activities defined in plan
    - No allocation for ongoing or unforeseen activities
    - Manpower assignments made in % of available development time, but this % was not clear
  - Informal effort and progress tracking
    - Weekly reports discussed at section meetings, provided by Project Leader

R16 : Credible WBS (II)

- For better understanding where developers spend their time and in order to reduce uncertainty, tracking items have been identified
  - For each activity in the Castor-dev plan:
    - time spent in a) development b) deployment c) support
    - Progress in % of the task definition
  - Same for all other activities:
    - Tasks not known/defined in the last plan
    - Including ongoing activities like preparing/building releases, 3rd-level support, support to T1’s, meetings etc
    - Specify time spent on non Castor-dev activities (seminars/training, common meetings, illness/holidays)
- A questionnaire is sent to all Castor developers in the section and their feedback is inserted into an Excel spreadsheet

R16 : Credible WBS (III)

- Examples:
  - Senior



days	FTE %	Castor%	Description	Dev	Depl	Supp	Compl. %
A) Total time spent in Castor development activities							
102	**93%**	**100%**		**33%**	**22%**	**45%**	
B) Tasks listed in the Castor-dev plan							
3	3%	3%	T3.6 - Filesystem GC trigger optimization	30%	70%	0%	100%
4	3%	4%	T5.1 - RFIO Common project	100%	0%	0%	80%
5	4%	5%	T7.3 - SRM2 updates and deployment	50%	50%	0%	70%
9	8%	0%	T9.2 - Makefiles and packaging	100%	0%	0%	15%
0	0%	0%	T9.3 - Automation of releases and tests	50%	50%	0%	0%
2	2%	2%	T10.1 - 64bit port (helping Rosa for initial setup)	100%	0%	0%	70%
1	0%	0%	T11.4 - Articles	100%	0%	0%	0%
23	**21%**	**23%**	Total	19	4	0	
C) Tasks not listed in the plan							
7	7%	7%	Castor releases, upgrade scripts and related documentation	0%	100%	0%	N/A
7	6%	7%	Slides preparation and presentations (Review, Tier1 meeting)	100%	0%	0%	N/A
16	14%	15%	Code debugging (from either savannah or direct findings), deadlocks huntin	0%	20%	80%	N/A
12	11%	11%	3rd level support to the operation team (remedy tickets, etc.)	0%	20%	80%	N/A
22	20%	21%	Support to CNAF (including official trips)	0%	0%	100%	N/A
9	9%	9%	All other miscellaneous developments	60%	40%	0%	N/A
1	1%	1%	SWEng Task Force related	100%	0%	0%	N/A
74	**68%**	**73%**	Total	14	16	44	
D) Other activities							
3	3%		Seminars and talks				
2	1%		Visits for the Visit service				
5	4%	0%	Castor Meetings (deployment, RFIO, SRM, Tier1 F2F, etc.)	20%	30%	50%	
2	2%		*Common meetings (group and section)*				
5			*Illness*				
20			*Holidays*				
106	98%		B+C+D - illness - holidays				
109	100%		The above including common meetings				
109	100%		working days (total days - illness - holidays) = base 100% FTE				
8			Official trips				

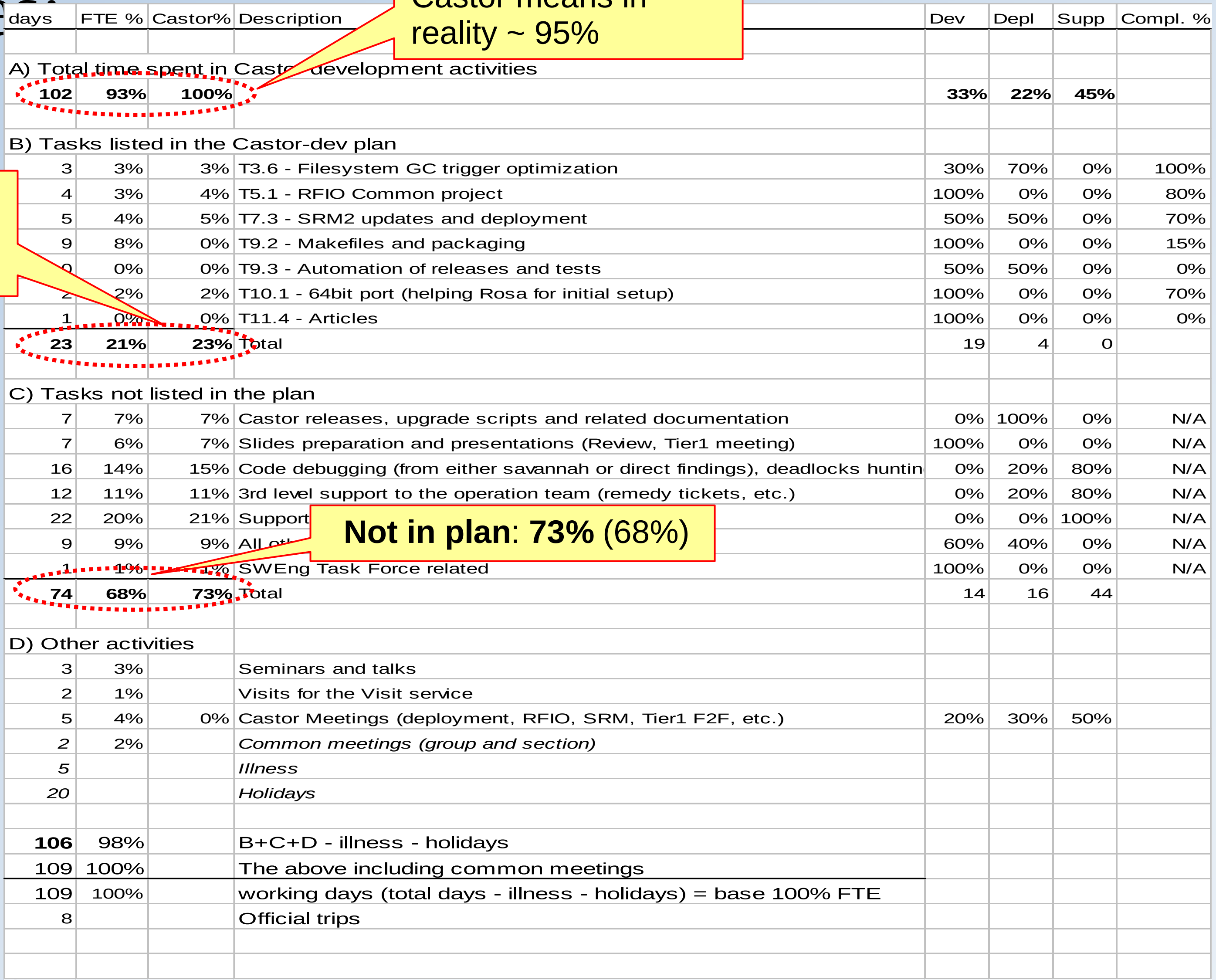
R16 : Credible WBS (III)

- Examples:
  - Senior

Allocation: 100% Castor means in reality ~ 95%

Planned tasks: Only **23%** of Castor-dev time (21% total time)

Not in plan: **73%** (68%)



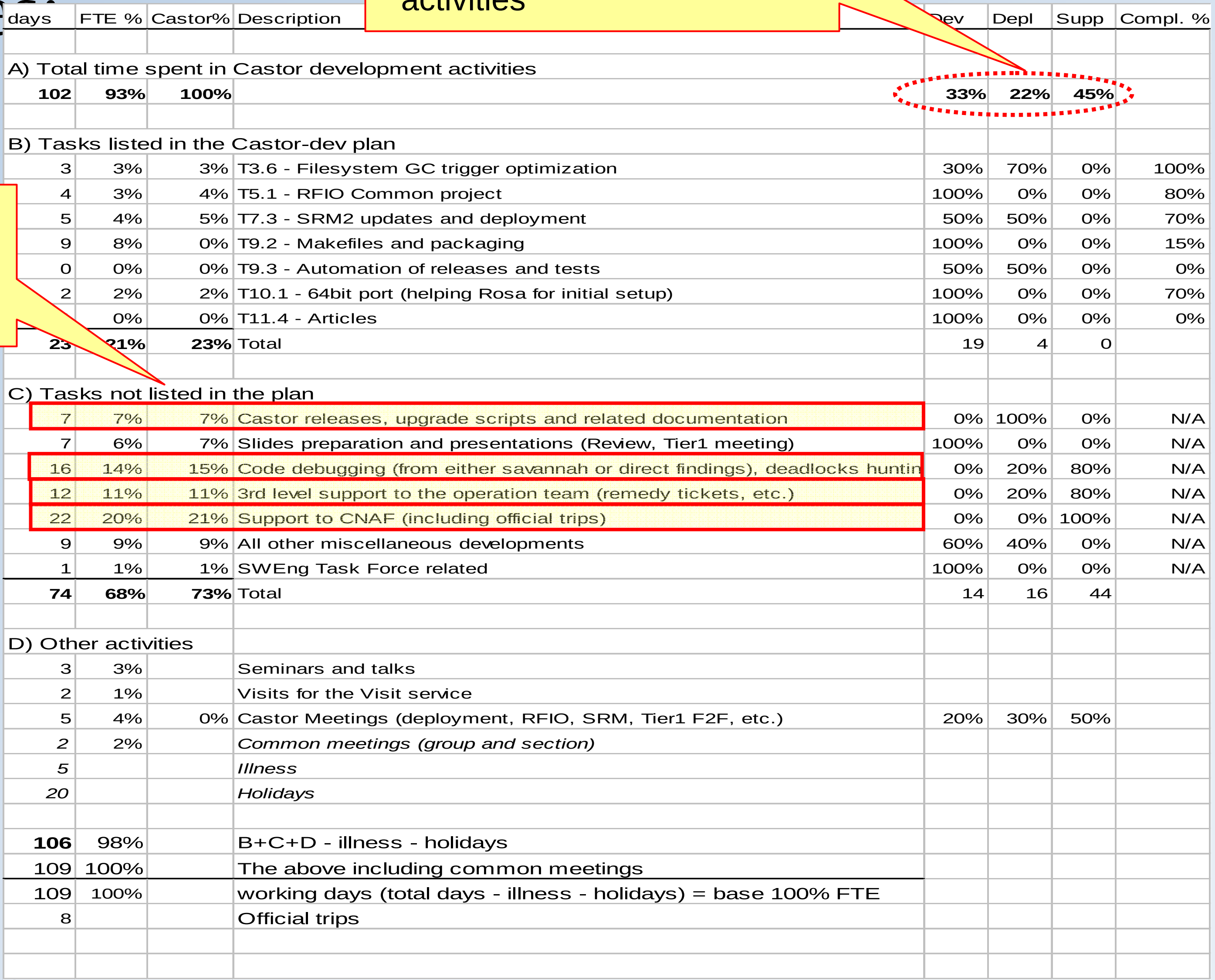
days	FTE %	Castor%	Description	Dev	Depl	Supp	Compl. %
A) Total time spent in Casto			development activities				
102	**93%**	**100%**		**33%**	**22%**	**45%**	
B) Tasks listed in the Castor-dev plan							
3	3%	3%	T3.6 - Filesystem GC trigger optimization	30%	70%	0%	100%
4	3%	4%	T5.1 - RFIO Common project	100%	0%	0%	80%
5	4%	5%	T7.3 - SRM2 updates and deployment	50%	50%	0%	70%
9	8%	0%	T9.2 - Makefiles and packaging	100%	0%	0%	15%
0	0%	0%	T9.3 - Automation of releases and tests	50%	50%	0%	0%
2	2%	2%	T10.1 - 64bit port (helping Rosa for initial setup)	100%	0%	0%	70%
1	0%	0%	T11.4 - Articles	100%	0%	0%	0%
23	**21%**	**23%**	Total	19	4	0	
C) Tasks not listed in the plan							
7	7%	7%	Castor releases, upgrade scripts and related documentation	0%	100%	0%	N/A
7	6%	7%	Slides preparation and presentations (Review, Tier1 meeting)	100%	0%	0%	N/A
16	14%	15%	Code debugging (from either savannah or direct findings), deadlocks huntin	0%	20%	80%	N/A
12	11%	11%	3rd level support to the operation team (remedy tickets, etc.)	0%	20%	80%	N/A
22	20%	21%	Support	0%	0%	100%	N/A
9	9%	9%	All ot	60%	40%	0%	N/A
1	1%	1%	SWEng Task Force related	100%	0%	0%	N/A
74	**68%**	**73%**	Total	14	16	44	
D) Other activities							
3	3%		Seminars and talks				
2	1%		Visits for the Visit service				
5	4%	0%	Castor Meetings (deployment, RFIO, SRM, Tier1 F2F, etc.)	20%	30%	50%	
2	2%		*Common meetings (group and section)*				
5			*Illness*				
20			*Holidays*				
106	98%		B+C+D - illness - holidays				
109	100%		The above including common meetings				
109	100%		working days (total days - illness - holidays) = base 100% FTE				
8			Official trips				

R16 : Credible WBS (III)

- Examples:
  - Senior

development+deployment vs. **support: 45%** of Castor-dev time spent on support activities

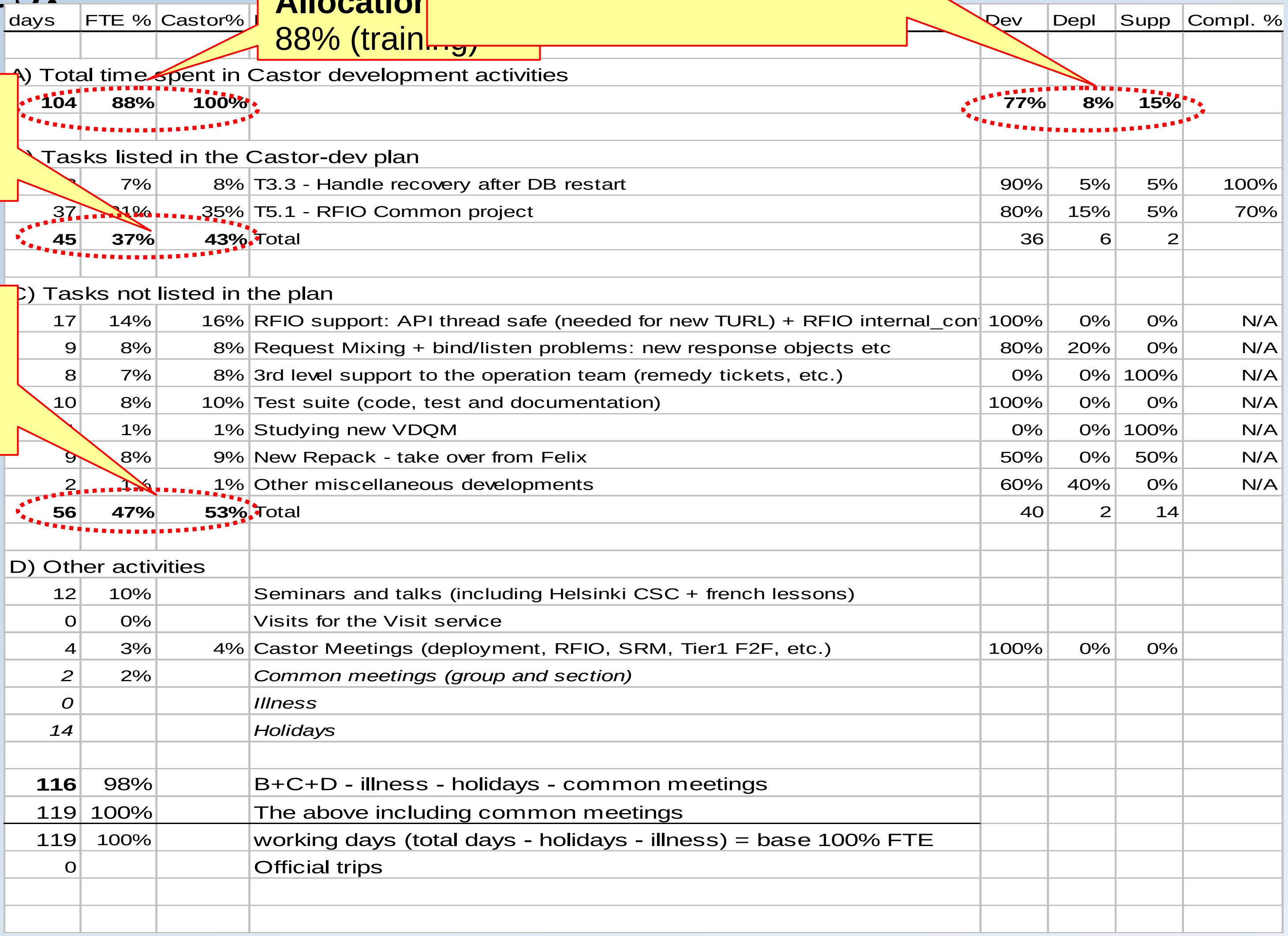
Not planned tasks: Most of them of **ongoing** nature (eg support, releases)



days	FTE %	Castor%	Description	Dev	Depl	Supp	Compl. %
A) Total time spent in Castor development activities							
102	**93%**	**100%**		**33%**	**22%**	**45%**	
B) Tasks listed in the Castor-dev plan							
3	3%	3%	T3.6 - Filesystem GC trigger optimization	30%	70%	0%	100%
4	3%	4%	T5.1 - RFIO Common project	100%	0%	0%	80%
5	4%	5%	T7.3 - SRM2 updates and deployment	50%	50%	0%	70%
9	8%	0%	T9.2 - Makefiles and packaging	100%	0%	0%	15%
0	0%	0%	T9.3 - Automation of releases and tests	50%	50%	0%	0%
2	2%	2%	T10.1 - 64bit port (helping Rosa for initial setup)	100%	0%	0%	70%
	0%	0%	T11.4 - Articles	100%	0%	0%	0%
23	**21%**	**23%**	Total	19	4	0	
C) Tasks not listed in the plan							
7	7%	7%	Castor releases, upgrade scripts and related documentation	0%	100%	0%	N/A
7	6%	7%	Slides preparation and presentations (Review, Tier1 meeting)	100%	0%	0%	N/A
16	14%	15%	Code debugging (from either savannah or direct findings), deadlocks huntin	0%	20%	80%	N/A
12	11%	11%	3rd level support to the operation team (remedy tickets, etc.)	0%	20%	80%	N/A
22	20%	21%	Support to CNAF (including official trips)	0%	0%	100%	N/A
9	9%	9%	All other miscellaneous developments	60%	40%	0%	N/A
1	1%	1%	SWEng Task Force related	100%	0%	0%	N/A
74	**68%**	**73%**	Total	14	16	44	
D) Other activities							
3	3%		Seminars and talks				
2	1%		Visits for the Visit service				
5	4%	0%	Castor Meetings (deployment, RFIO, SRM, Tier1 F2F, etc.)	20%	30%	50%	
2	2%		*Common meetings (group and section)*				
5			*Illness*				
20			*Holidays*				
106	98%		B+C+D - illness - holidays				
109	100%		The above including common meetings				
109	100%		working days (total days - illness - holidays) = base 100% FTE				
8			Official trips				

R16 : Credible WBS (IV)

- Examples:



days	FTE %	Castor%		Dev	Depl	Supp	Compl. %
A) Total time spent in Castor development activities							
104	**88%**	**100%**		**77%**	**8%**	**15%**	
B) Tasks listed in the Castor-dev plan							
8	7%	8%	T3.3 - Handle recovery after DB restart	90%	5%	5%	100%
37	21%	35%	T5.1 - RFIO Common project	80%	15%	5%	70%
45	**37%**	**43%**	Total	36	6	2	
C) Tasks not listed in the plan							
17	14%	16%	RFIO support: API thread safe (needed for new TURL) + RFIO internal_con	100%	0%	0%	N/A
9	8%	8%	Request Mixing + bind/listen problems: new response objects etc	80%	20%	0%	N/A
8	7%	8%	3rd level support to the operation team (remedy tickets, etc.)	0%	0%	100%	N/A
10	8%	10%	Test suite (code, test and documentation)	100%	0%	0%	N/A
1	1%	1%	Studying new VDQM	0%	0%	100%	N/A
9	8%	9%	New Repack - take over from Felix	50%	0%	50%	N/A
2	1%	1%	Other miscellaneous developments	60%	40%	0%	N/A
56	**47%**	**53%**	Total	40	2	14	
D) Other activities							
12	10%		Seminars and talks (including Helsinki CSC + french lessons)				
0	0%		Visits for the Visit service				
4	3%	4%	Castor Meetings (deployment, RFIO, SRM, Tier1 F2F, etc.)	100%	0%	0%	
2	2%		*Common meetings (group and section)*				
0			*Illness*				
14			*Holidays*				
116	98%		B+C+D - illness - holidays - common meetings				
119	100%		The above including common meetings				
119	100%		working days (total days - holidays - illness) = base 100% FTE				
0			Official trips				

Allocation
88% (training)

development+deployment vs. **support:** Only 15% goes to support

Planned tasks: 43% of Castor-dev time

Not planned : 53%. Out of which 45% development activities (mostly unforeseen) and 8% support

R16 : Credible WBS (V)

- Findings and conclusions for future planning:
  - Clear profile difference between ‘senior’ (Sebastien, Giuseppe) and ‘junior’ (the rest!) developers
    - dvp time vs. support/training/coaching/management time
    - try allocating development activities to those having the time for it!
  - Significant overall support load (3rd-level, Tier-1’s, bugfixes) reducing available time for developments
  - Required development tasks and their duration cannot be always reliably predicted, so a generic ‘development buffer’ is required
    - only a fraction of the Castor codebase is known to the development team
    - Continuous flow of new requirements and RFE’s streaming from end users, ops team at CERN, ops teams at T1’s
  - Changes to WBS:
    - Take into account only available manpower fractions
    - Reduce uncertainty: Add all known activities eg. ongoing support activities, and not identified development tasks
    - Avoid any development allocation below 10%
    - In addition to task duration estimation (now in FTE/days) and task completion (%), add consumed manpower so far (also FTE/days). Separate ‘consumption’ from ‘progress’; take ‘consumption’ into account for updated duration and completion estimations. Take into account bottom-up data for re-assessing progress.
    - (To Do) Add finer-grained deliverables and break down tasks exceeding 20 FTE/days

R16 : Credible WBS (VI)

- Pending/ongoing:
  - Define more formal (but lightweight..) mechanisms for reviewing development WBS and progress with Castor Ops team
  - Identify a light-weight effort tracking tool (for both Castor-dev and ops teams) which can be used at a (bi-)weekly basis
  - Include external collaborators in tracking (outside the section)
  - Move to more deliverable-based rather than activity-based planning e.g. GDPM

- Concerns:
  - High level of interruptions to Castor-development team
    - In particular for ‘senior’ team members
    - Significantly improved with knowledge build-up on Castor-Ops side, but still too high (e.g. Tier-1 sites)
    - A weekly support rota for developers has been put in place
  - Many external dependencies make planning ahead difficult
  - Management overhead not always appreciated
    - Keep it to the required minimum
    - Resources are limited and not easy to reallocate to this task
    - Not all developers see clear benefits