Software Development and Testing

Approach and Challenges in a distributed HEP Collaboration

Doris Burckhart-Chromek
CERN PH
ATLAS TDAQ

ICALEPCS 2007 Knoxville, Tennessee

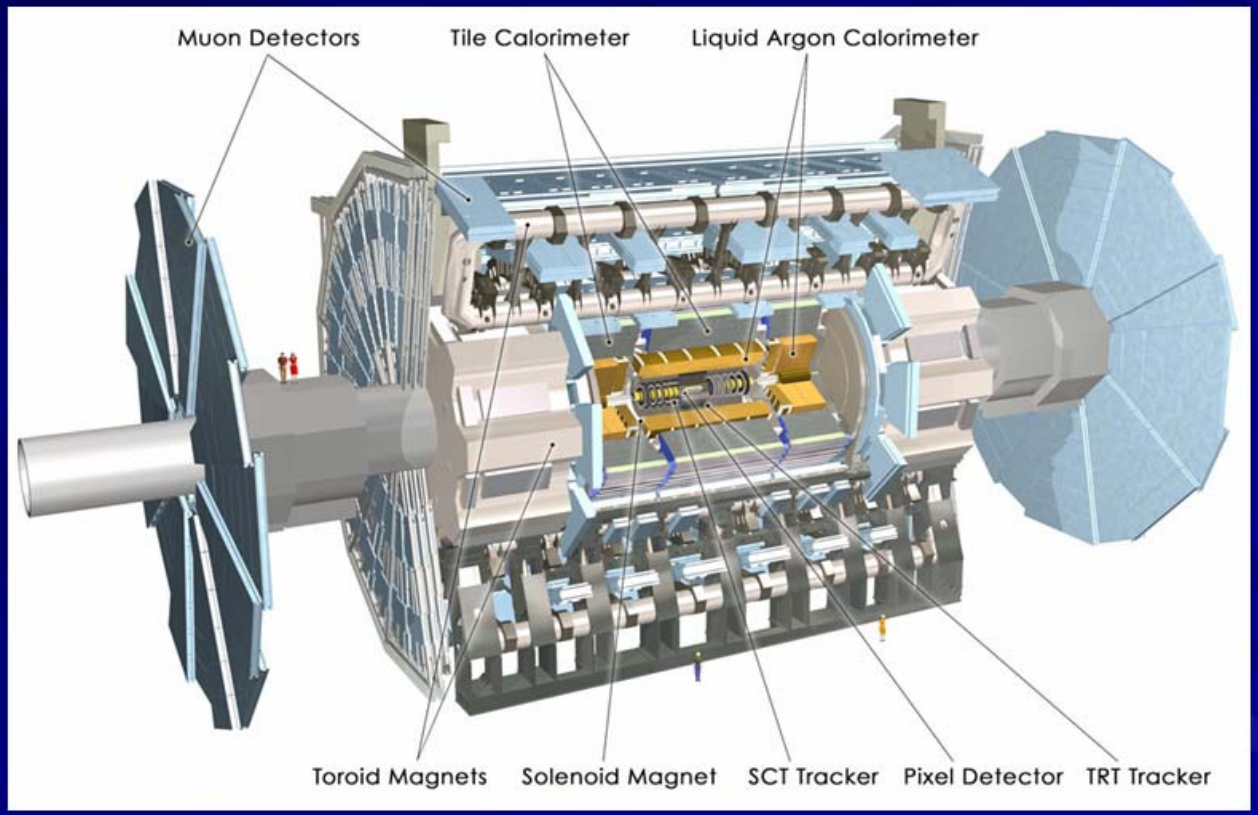
Content

- The ATLAS Experiment
- The Trigger and Data Acquisition Project
- The SDP challenge in ATLAS Online
- SDP Models
- Software Management
- Review and Inspection
- The ALICE Online SDP
- Testing
- Do's and Don'ts
- Conclusions

The ATLAS Detector

At the Large Hadron Collidor at CERN

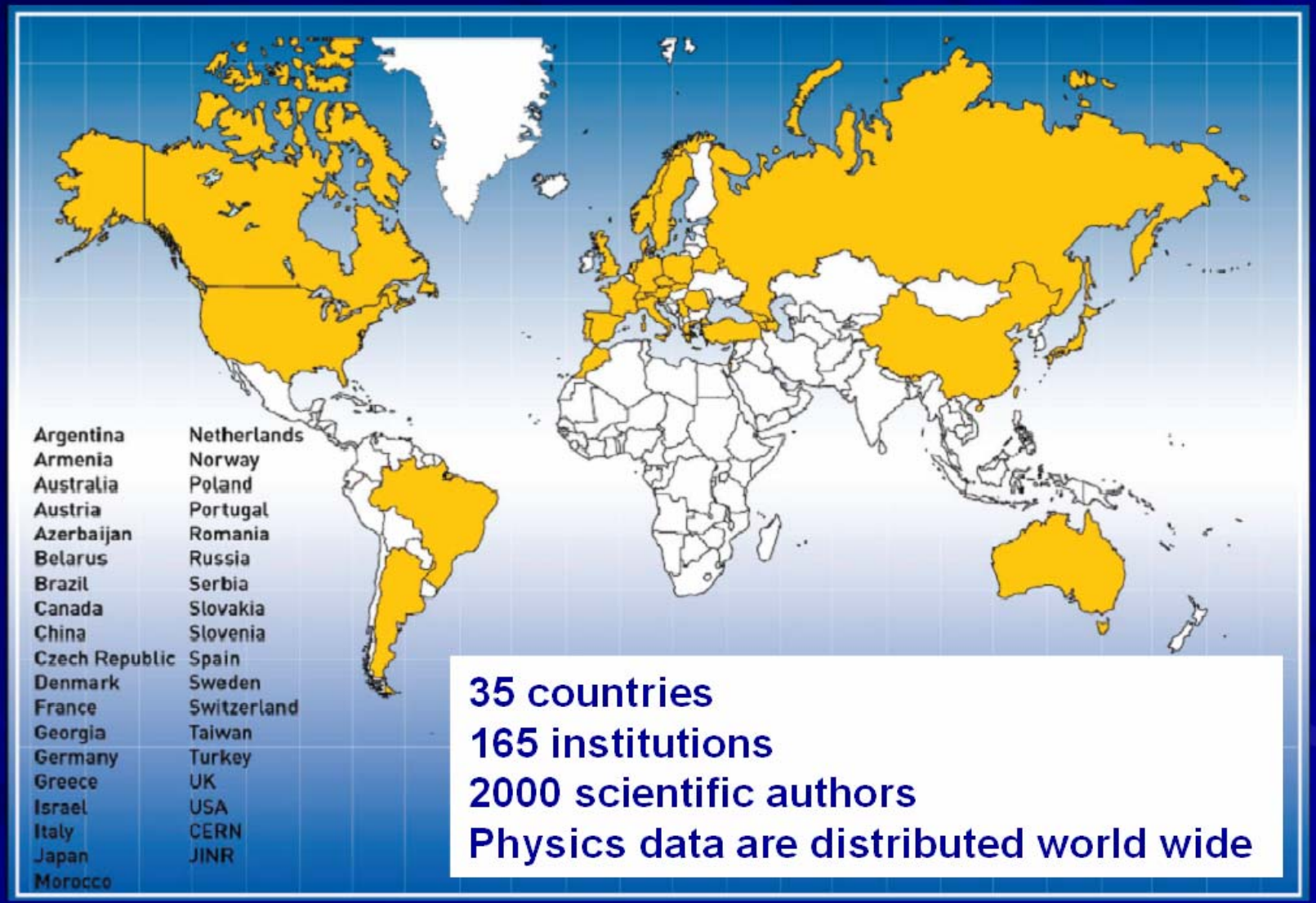
pp collisions at 14 TeV every 25 ns: Higgs, Supersymmetry and more



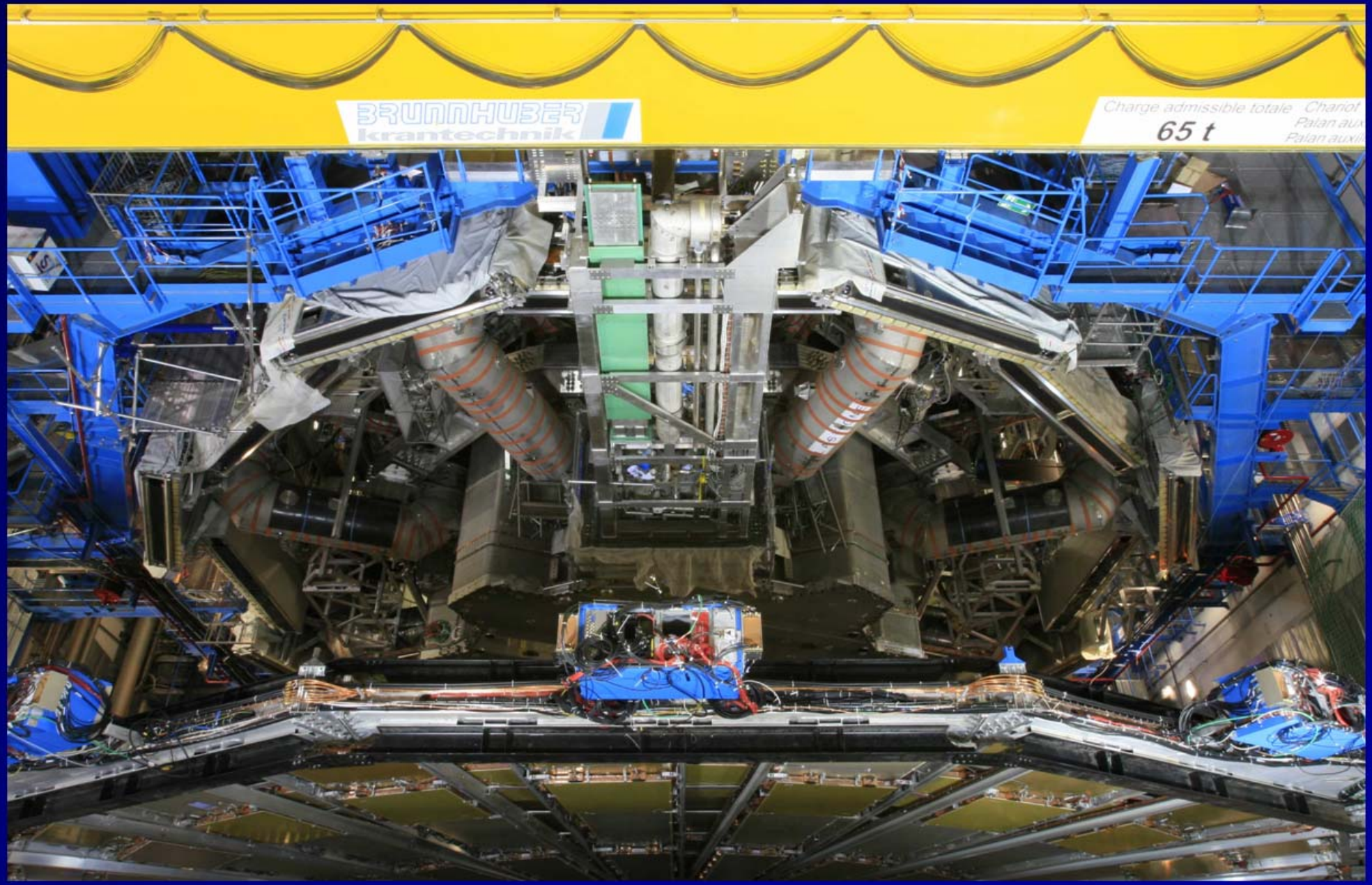
Weight: 7 000 tonnes
Diameter 25 m
Length 46 m

140 million electronic channels
3 000 km of cables

ATLAS Collaboration



The ATLAS Experiment

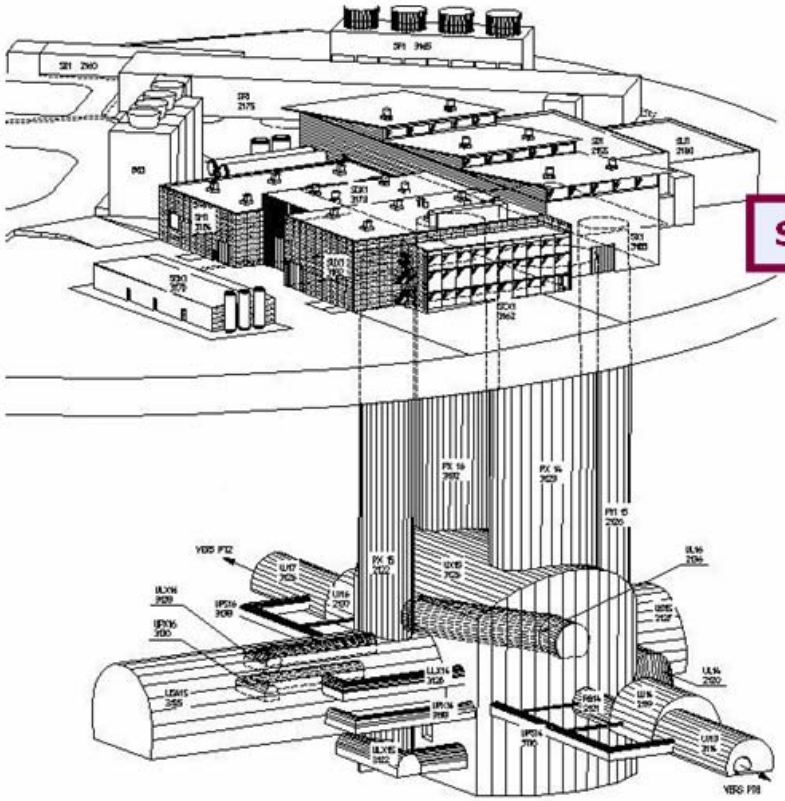


Current Status

Close to completion

Move of the Endcap Toroid (280 tonnes) into the ATLAS hall, July 2007

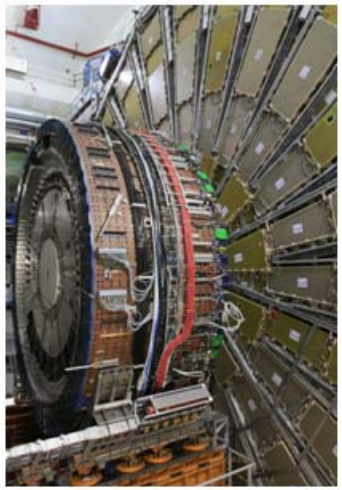
ATLAS Trigger / DAQ Data Flow



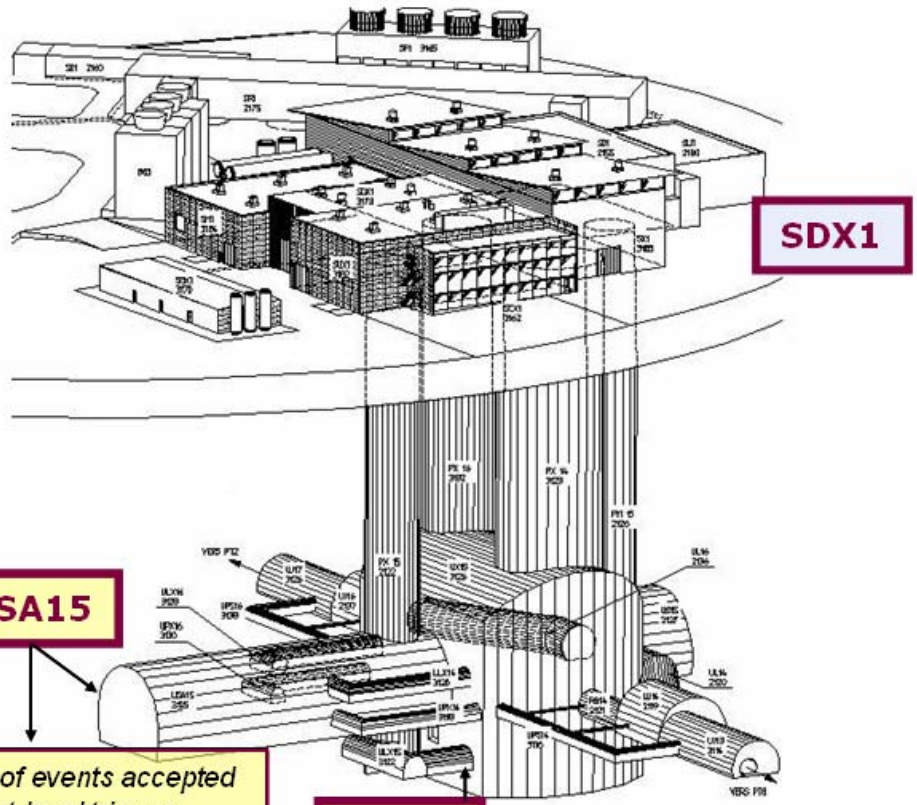
SDX1

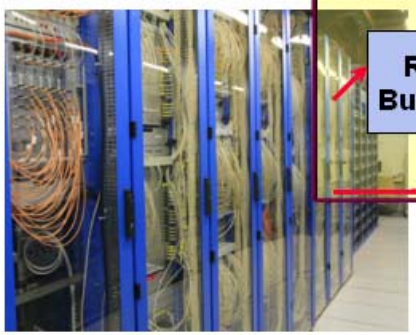
ATLAS
detector

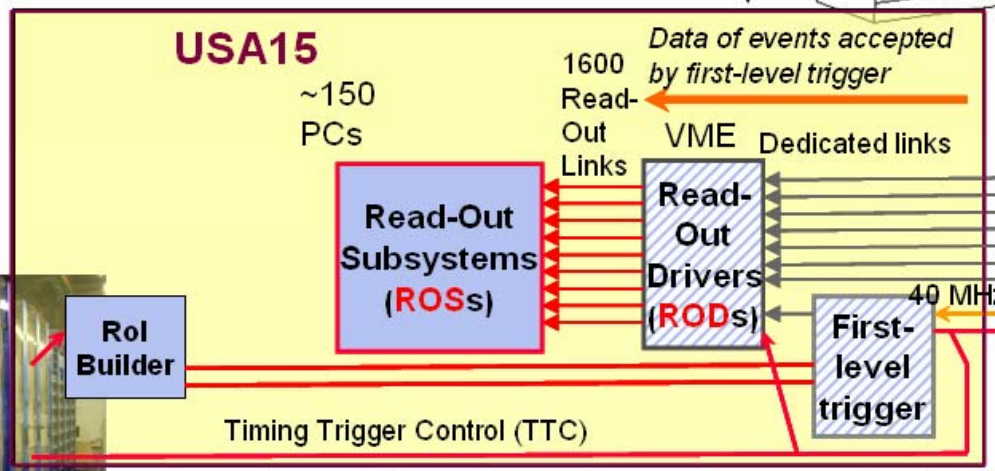
UX15

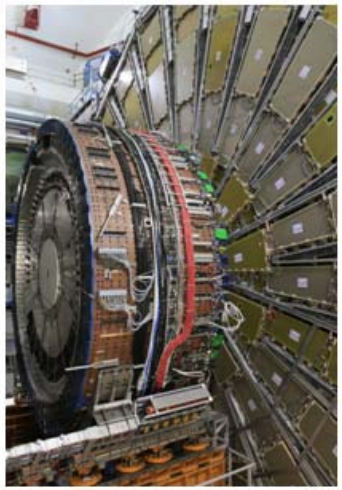


ATLAS Trigger / DAQ Data Flow



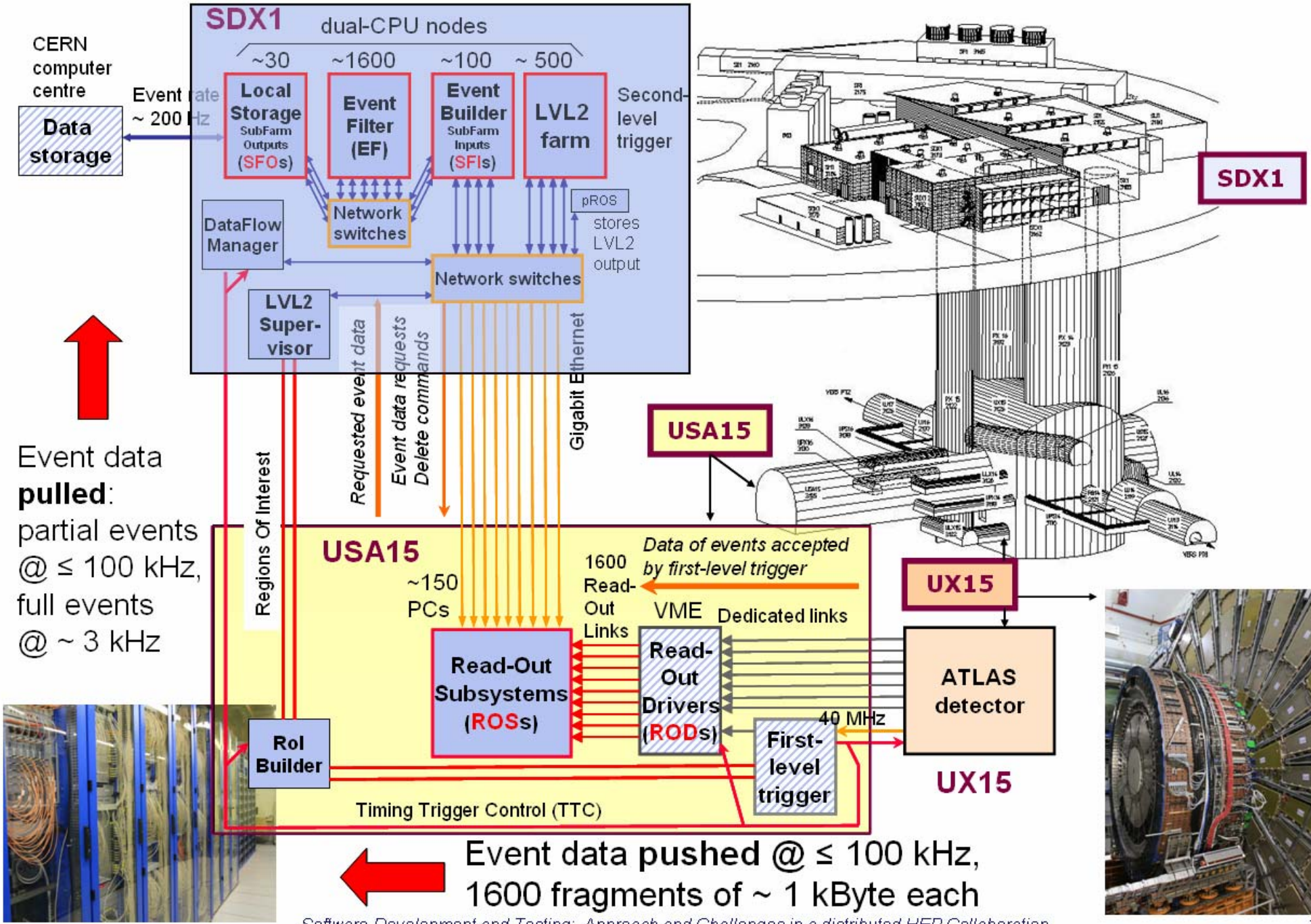




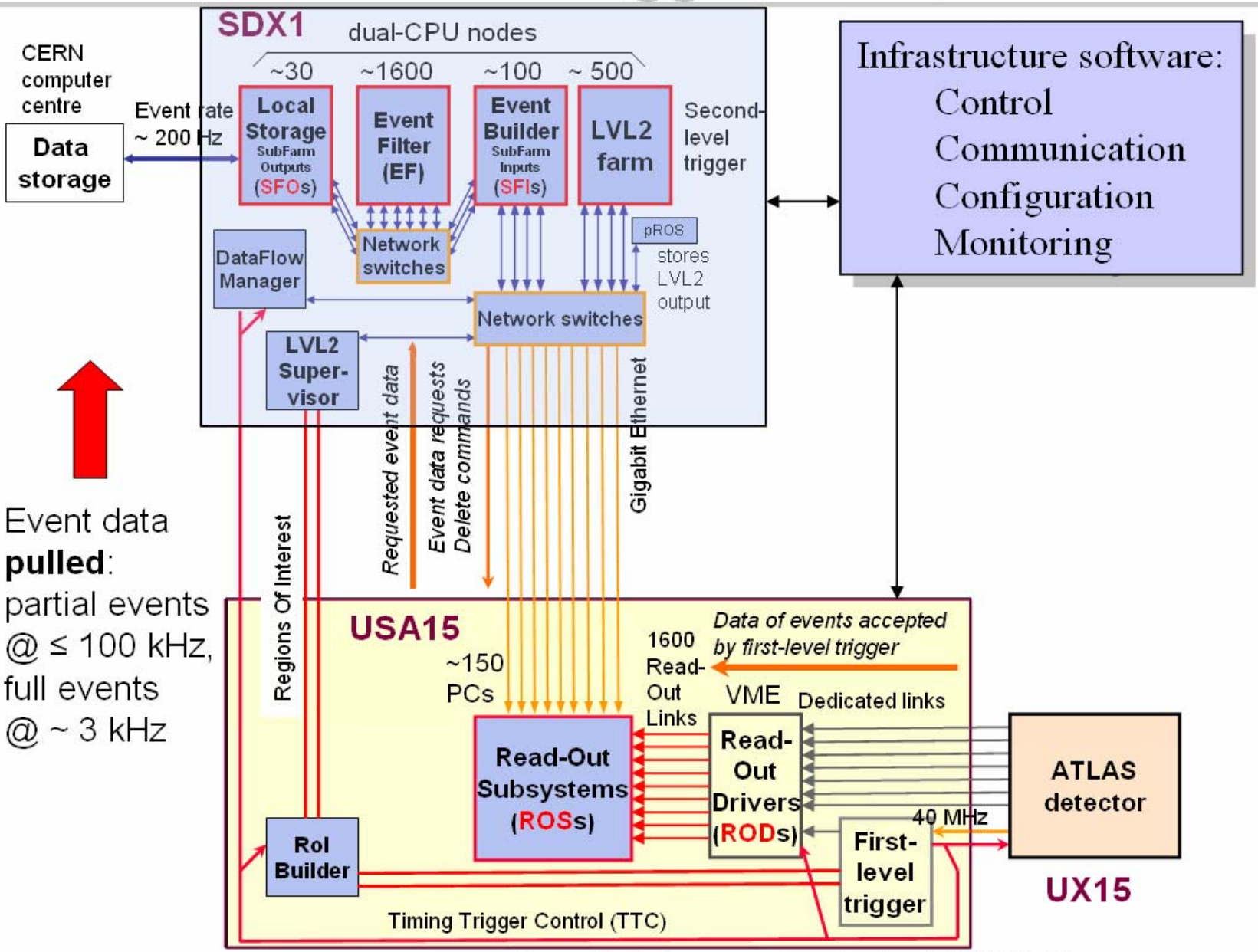


Event data **pushed** @ ≤ 100 kHz,
1600 fragments of ~ 1 kByte each

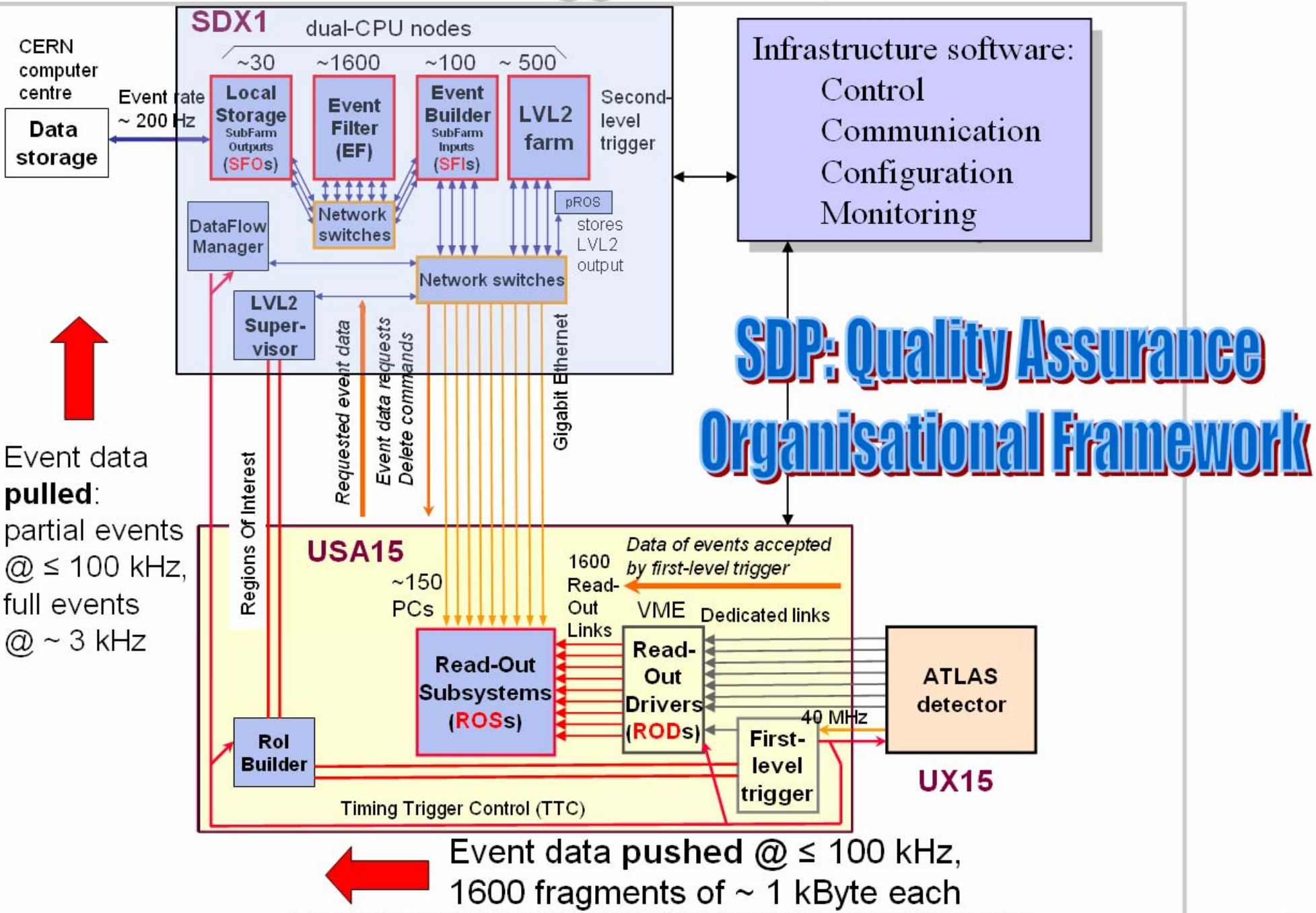
ATLAS Trigger / DAQ Data Flow



ATLAS Trigger / DAQ Data Flow



ATLAS Trigger / DAQ Data Flow



ATLAS TDAQ environment

No SDP model suitable 'as is'

- **The project**
  - **Long start-up time - 10 years**
    - Technical requirements not all knows when starting
    - Subset of the system needed early on in test beams
    - Advanced system for detector commissioning
  - **Long life time of the experiment – 15 years**
  - **60 participating institutes located all around the world**
    - Nominally 400 collaborators
  - **No strong hierarchical management power**
    - In a scientific collaboration we rely on participation by conviction
- **The team in the scientific environment**
  - **Large, geographically widespread team**
  - **High turnover of developers**
    - Short time contracts
    - Additional duties at universities and in labs
  - **Predominantly physicists with some participation of computer scientists**
    - Reluctant if faced with a working framework and rules to obey
    - Development habits from small projects
    - No specialized programming expertise

ATLAS TDAQ SDP

'Light' SDP

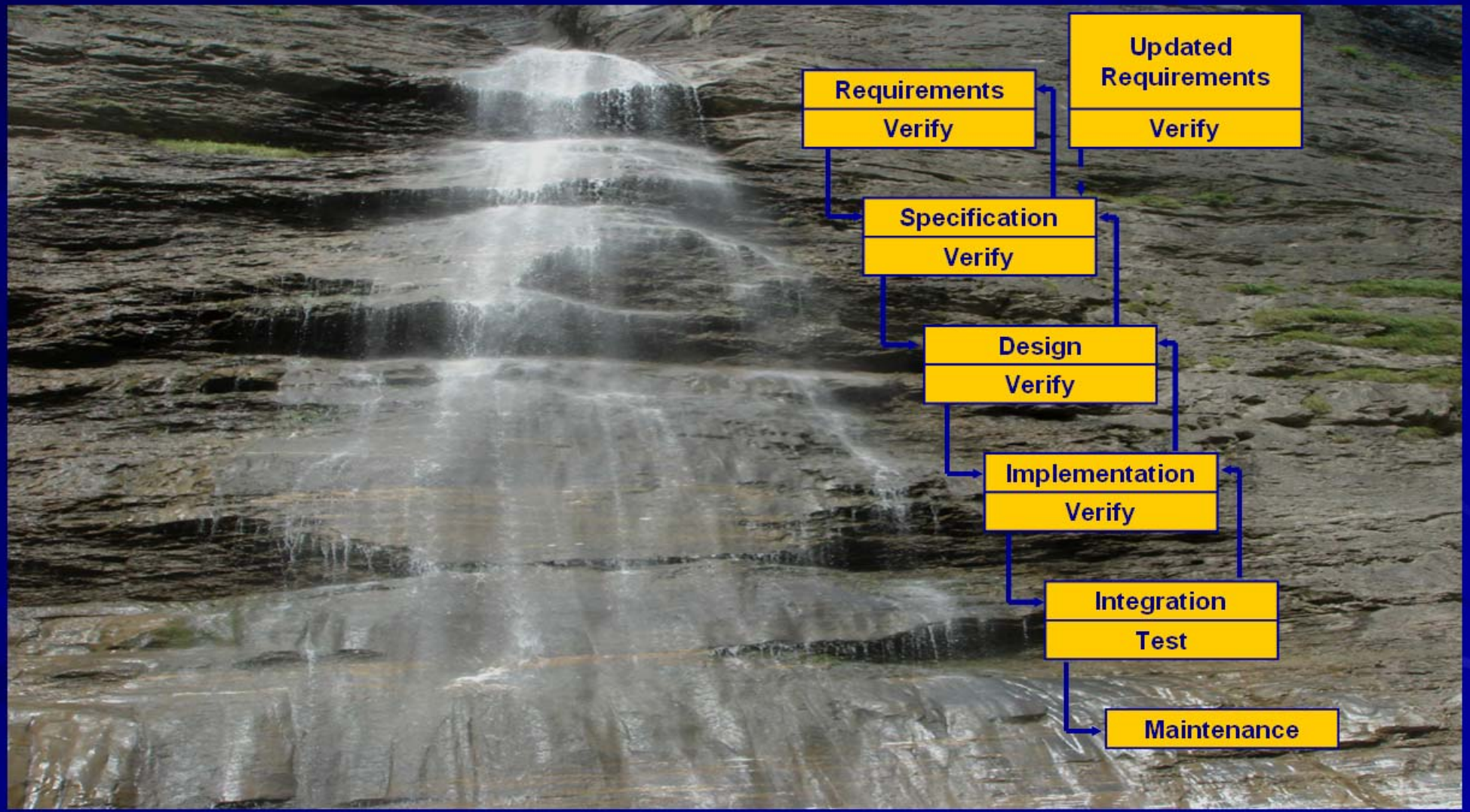
- Structured organization as a helpful framework
- Integration of team members
  - Introduce process gently
    - Initially introduced the SDP in a sub-group with 12 components and 10 part time developers
    - Later the other TDAQ sub-systems were gradually integrated: 160 packages, 30 core developers
    - Handle geographical distribution and turnover of team members
- Define and agree on common goals and priorities
  - -> community feels responsible for the results
  - -> increased communication
- Value and use incomplete contributions
  - Based on 'best effort' approach - not perfect
  - 'We are not gurus, just concerned developers'
- Allow for iterations

The Waterfall model

or 'linear sequential model'

The Waterfall model

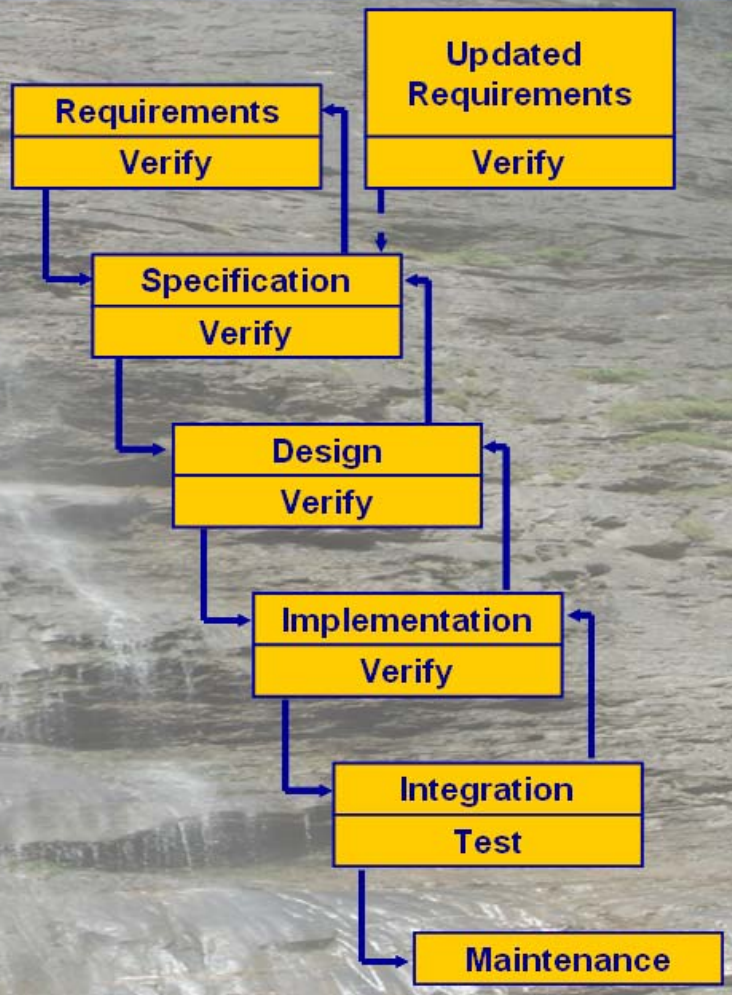
or 'linear sequential model'



The Waterfall model

or 'linear sequential model'

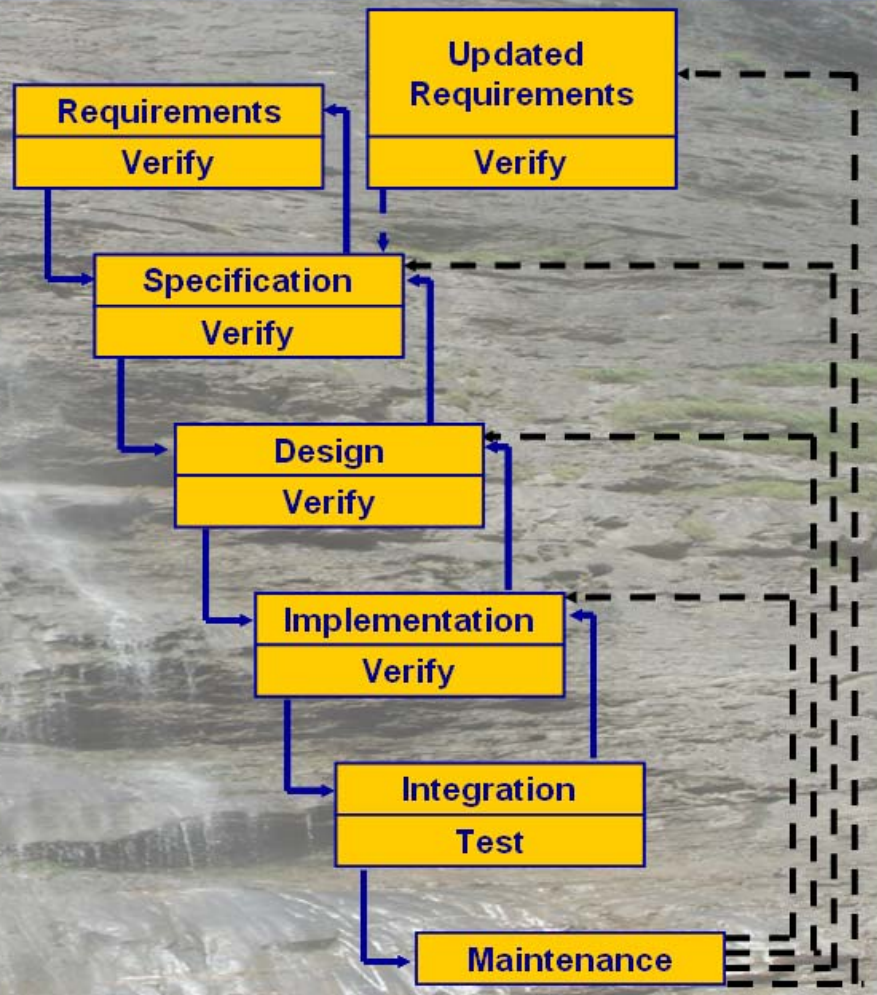
- The oldest and the most widely used paradigm
- Clear development phases
- Enforced disciplined approach
- Testing inherent to every phase
- Documentation driven
- Missing check points with the user during development
- Iterative cycles allowed only to each previous phase
- Maintenance paths to all phases



The Waterfall model

or 'linear sequential model'

- The oldest and the most widely used paradigm
- Clear development phases
- Enforced disciplined approach
- Testing inherent to every phase
- Documentation driven
- Missing check points with the user during development
- Iterative cycles allowed only to each previous phase
- Maintenance paths to all phases



Other models and methods

Spiral lifecycle model

Barry Boehm, 1988 "A Spiral Model of Software Development and Enhancement"

- Each iteration is carefully planned from the start
- Risk assessment
  - Prototyping, simulation, benchmarking
  - Estimates get more realistic as work progresses, because important issues are discovered earlier
- Ability to accommodate change
- Rigid approach

Agile methods

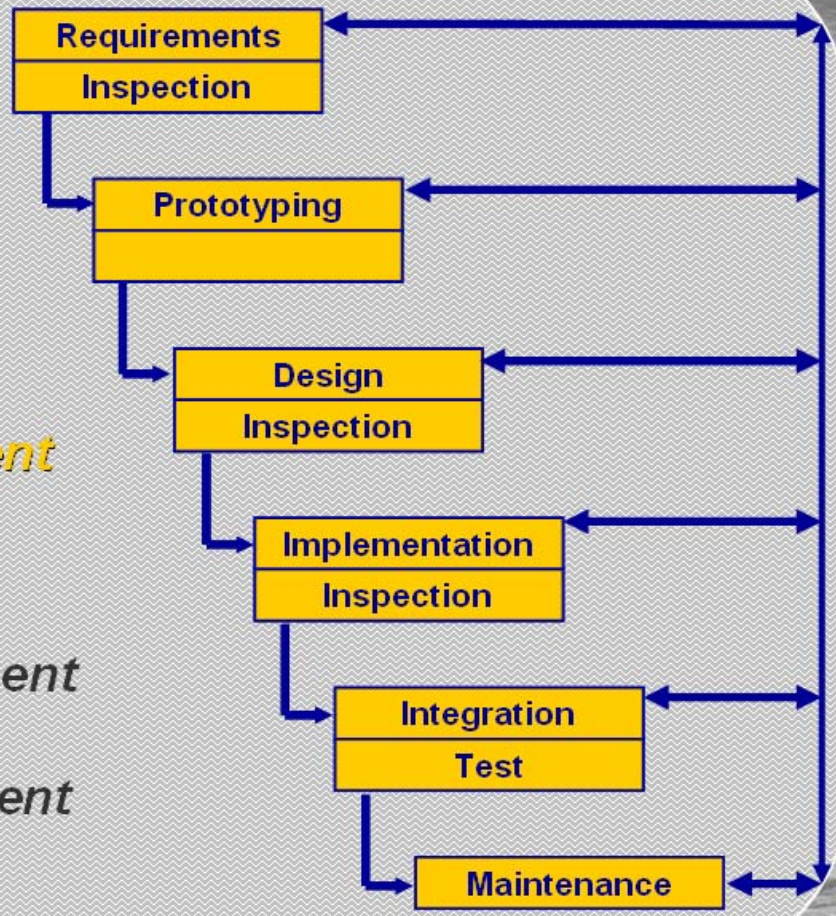
- Aimed for projects with rapidly changing requirements
  - Web oriented tasks
- Rapid evolutionary delivery throughout the project lifecycle
  - Days to weeks
- Emphasis on testing
  - tests are written before component code
- Pair programming is favored
- Used for small project teams located in one room or close by
- Little documentation
- No concept for collaboration with remote sites

ATLAS Online SDP

Based on iterative development
Iterative cycles are allowed between each of the phases

Embedded in a
Software Management Environment

Assisted by Tools for
Process management
Programming
Release management
Configuration



SDP Organization

- Work is organized around components
  - Small groups are dedicated to each component
  - Prefer one institute per component -> clear boundaries of responsibilities
  - Look for commonalties between components - don't duplicate functionality
  - Encourage to take / reuse as much as possible from colleagues: code, ideas, documentation, templates, examples
- Components are developed according to an agreed priority
- SDP documented on the Web allowing for change
  - Brief Check-lists and guidelines
    - For requirements, design and general documentation
    - Coding standards, automatic checking tools (C++)
    - Simple "how-to' instructions for most commonly used tools
  - Documentation Templates
    - From external sources or developed within the project
    - Provides type of content & style and includes examples
    - Generic technical note, test plan & test report, user requirements, design document, users guide
- Regular reviews

Structure
Flexibility
Support

Software Management Tools in ATLAS

Coherent releases allow to join the force of the project developers

- Nightly build for 2 compilers on linux, optimized and debug versions
- Now 3-4 major releases per year
- 160 packages
- 4000 source files
- 1 000 000 lines of code
- C++, Java, Python
- total 60Mbytes
- 2 GByte total size of nightly build
- Release nodes part of the build

Software Management Tools in ATLAS

Coherent releases allow to join the force of the project developers

- Nightly build for 2 compiler optimized and debug versio
- Now 3-4 major releases per
- 160 packages
- 4000 source files
- 1 000 000 lines of code
- C++, Java, Python
- total 60Mbytes
- 2 GByte total size of nightly
- Release nodes part of the bu

	i686-slc3-gcc323-opt		i686-slc4-gcc34-dbg		i686-slc4-gcc34-opt	
ac v4r2p1	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
AccessManager AccessManager-00-03-08	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
clips HEAD	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
ClipsServer HEAD	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
cmdl HEAD	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
cmem_rcc v2r0p23	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
coca HEAD	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
config HEAD	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
dal HEAD	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
DAQRelease DAQRelease-00-00-28	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
DataflowPolicy v1r6p0	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
dcmessages v1r1p15	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
ddc HEAD	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
DFConfiguration v6r3p2	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
DFDebug v2r0p16	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
DFDVSTests v4r3p2	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
DFExceptions v3r0p0	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
DFInfo v2r2p16	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
DFM v2r15p4	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
DFMessage v2r0p0	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
DFRelease pre-01-08-00	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK
DFSubSystemItem v6r3p3	MAKE	CHECK	MAKE	CHECK	MAKE	CHECK

Software Management Tools in ATLAS

Coherent releases allow to join the force of the project developers

- Nightly build for 2 compilers on linux, optimized and debug versions
- Now 3-4 major releases per year
- 160 packages
- 4000 source files
- 1 000 000 lines of code
- C++, Java, Python
- total 60Mbytes
- 2 GByte total size of nightly build
- Release nodes part of the build

Software Management Tools in ATLAS

Coherent releases allow to join the force of the project developers

- Nightly build
  optimized and
- Now 3-4 majo
- 160 packages
- 4000 source fi
- 1 000 000 line
- C++, Java, Py
- total 60Mbyte
- 2 GByte total
- Release nodes

Atlas Online Software

Development

SDP C++ CMT Results of build Doxygen Email lists

NB: if links does not work for you (e.g. from outside CERN), please remove 's' from https:// in URLs

Release (link to LXR)	Date	Comments	build logs	Download (patch level)	Release Notes	Doxy	Common releases	External s/w syncronization
tdaq-01-08-03	01.10.2007	M5 production release		0		C++ Java	tdaq-common-01-08-00 dqm-common-00-03-00	LCGCMT_53b, dqm-common-00-03-00
tdaq-01-08-02	14.09.2007	internal tdaq release, alias for tdaq-01-08-01 + LCGCMT_53b - used for Technical Runs only					tdaq-common-01-07-01 dqm-common-00-02-08	LCGCMT_53b, dqm-common-00-02-10
tdaq-01-08-01	24.07.2007	rebuild of tdaq-01-08-00 with patches and new offline/LCG syncronization; **M4 production release**		68		C++	tdaq-common-01-07-01 dqm-common-00-02-08	LCG_53a,b / offline-13.0.2x
tdaq-01-08-00	05.06.2007	production release; *NOT MAINTAINED - move to tdaq-01-08-01*		15		C++	tdaq-common-01-06-01 dqm-common-00-02-03	LCG_51 / offline-13.0.10
tdaq-01-07-00	24.01.2007	current production release for ATLAS comissioning		100		C++	tdaq-common-01-05-00	LCG_43b,c
tdaq-01-06-02	30.10.2006	LST-06 release tdaq-01-06-01 + fixes + SLC4 support. **NOT A PUBLIC RELEASE**		N/A		C++	tdaq-common-01-04-00	LCG_43b,c/COOL_1_3_2/offline-12.0.X
tdaq-01-06-01	26.08.2006	production release: patched tdaq-01-06-00 + new ROS s/w				C++	tdaq-common-01-04-00	LCG_43b,c/COOL_1_3_2/offline-12.0.X
tdaq-01-06-00	24.05.2006	obsolete		obsolete		C++	tdaq-common-01-04-00	LCG_43b,c/COOL_1_3_2/offline-12.0.X
nightly 270307	27.03.2007	saved nightly		N/A		C++	tdaq-common-01-06-00	LCG_50b
nightly 1603	16.03.2006	last stable nightly with tdaq-common-01-03-00 and P4 optimization		N/A			tdaq-common-01-03-00	LCG_41
nightly	today	build with HEAD or latest tags from CVS		N/A		C++ Java	tdaq-common-01-07-01 dqm-common-00-02-08	LCG_53
tdaq-01-05-00	20.01.2006	internal release for new ROBIN exploitation, based on tdaq-01-04-01 tags		obsolete	N/A		tdaq-common-01-01-01	LCG_36
tdaq-01-04-01	16.10.2005	tdaq-01-04-00 + patches + new externals				C++	tdaq-common-01-01-00	COOL_1_2_3/offline-11.0.x
tdaq-01-04-00	09.09.2005	"installation and comissioning" release					tdaq-common-01-01-00	COOL_1_2_0/offline-10.0.X
tdaq-01-03-00	08.06.2005	obsolete					tdaq-00-01-00	
tdaq-01-02-00	24.05.2005						tdaq-00-01-00	
tdaq-01-01-00	21.02.2005						not used	

Done

Software Management Tools in ATLAS

Coherent releases allow to join the force of the project developers

- Nightly build for 2 compilers on linux, optimized and debug versions
- Now 3-4 major releases per year
- 160 packages
- 4000 source files
- 1 000 000 lines of code
- C++, Java, Python
- total 60Mbytes
- 2 GByte total size of nightly build
- Release nodes part of the build

- Code versioning system CVS
- Configuration management system CMT
- Custom made scripts
- Build process takes 3-4 hours thanks to serialization and parallelism
- Presentation of build status on web pages
- Automatic information of component failures to the corresponding developer
- RPM for package building
- APT for package distribution
- VALGRIND as preferred memory debugger and performance profiler
- doxyGen for automatic documentation generation of the application interfaces

Informal Review

Informal Review: Clarification and accept/reject decision

- **Presentation of each component** to the group in each development phase
- Discussion and **coordination with other components**

- In ATLAS TDAQ SDP **applied from the start** of the project
  - document preparation and monthly open meetings
  - present status, results, proposals
  - inform colleagues - receive feedback
  - suggestions -> enhancements -> acceptance
- Results:
  - Coherent set of end-product components
  - Increased communication
  - Created the basis for a working culture
- Drawback:
  - **Lack of time of reviewers**
  - No code review

The Power of Software Inspection

Formal Inspection:

Quality Improvement Process to the software project

Defect Detection

- Documents are checked for cleanness and consistency against rules

Defect Prevention

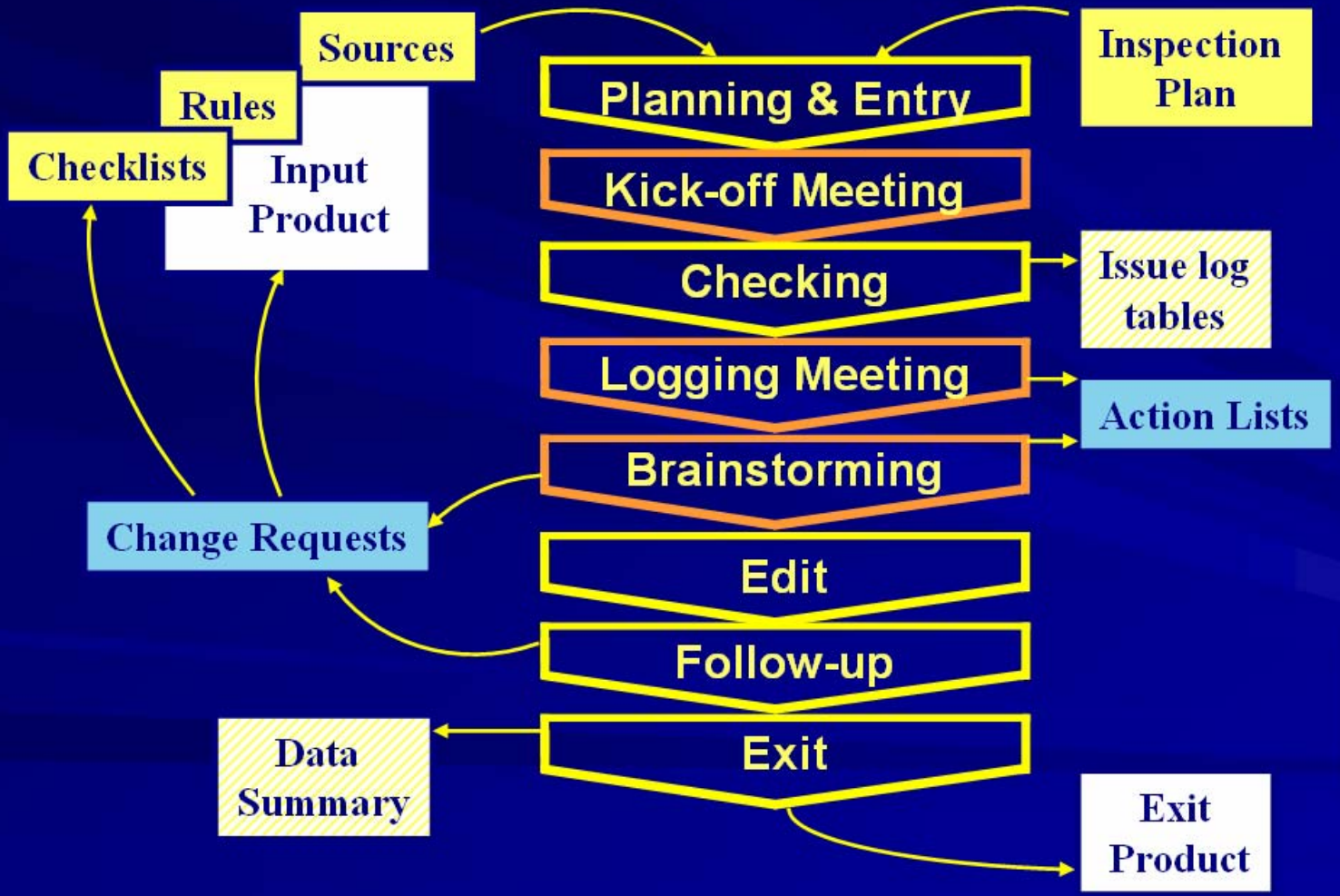
- Participants
  - learn from defects found
  - and suggest improvements

On the Job Training & Integration

- Peers
  - get familiar with standards and rules
  - apply creativity
  - learn to accept criticism
  - learn to trust colleagues
  - get integrated into the group & project

Inspection Process Map

Based on Tom Gilb's Inspection method



Inspection Team:
Inspection Leader
Authors
Inspectors

Vehicle to build a project culture

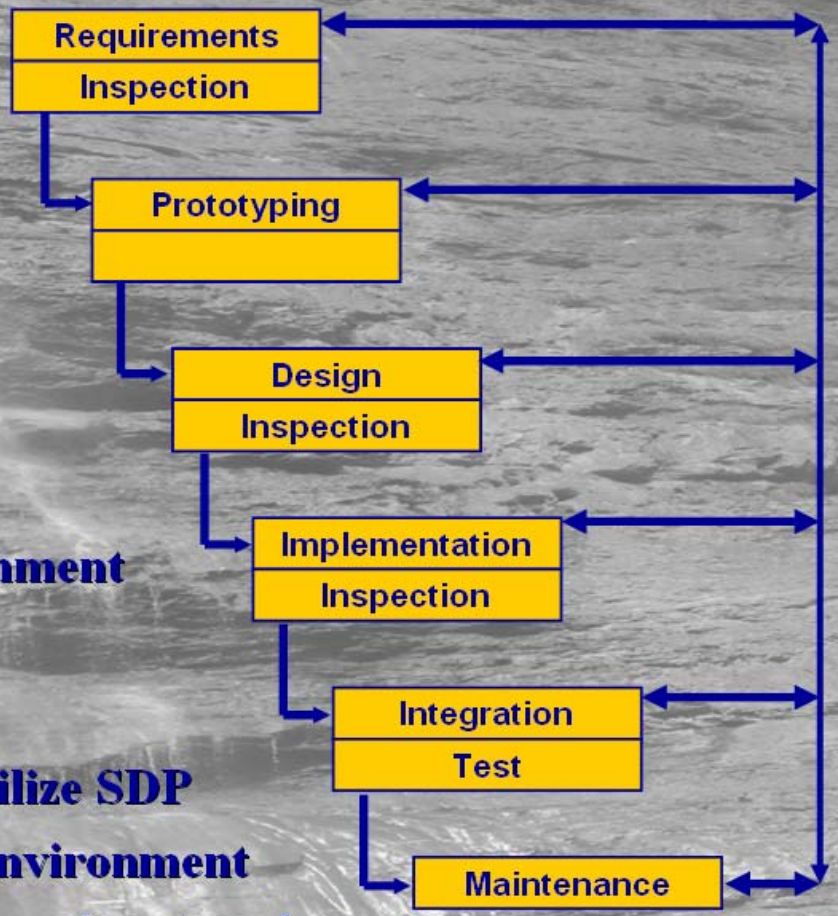
Inspection & Review: Experience

Requirements
- Highest payoff

Design
- Use walkthrough methos

Code
- Inspect samples
- Need a good set of rules
- Use automatic checking tools

General
- Method had to be adapted to the Environment
- Gain in quality and experience
- Appreciated by authors and peers
- Inspections prepare the ground and stabilize SDP
- Help for team building in a distributed environment
- Helped for breakthrough in politically prominent packages



Light form of inspection

- Can be applied when
  - working habits are assimilated by the team
  - criticism is accepted
- Retains primary benefits of inspection
- Meetings are replaced by the use of electronic communication tools
- Less time consuming for reviewers
- Less organizational overhead

- *Not with reviewers which are external to the project*
- *Not for politically prominent components*

The SDP in ALICE Online

- ALICE detector at the LHC is smaller than ATLAS
- TDAQ has high requirements on the performance of physics data storage
- Development team
  - Team started with 4 very experience developers
  - Familiarity of team members since a decade and a common history
  - Long term experience in development and support
  - Newcomers could be integrated stepwise (up to 10)
  - Based at CERN
  - Immediate task to provide a TDAQ system for related experiments
- Approach
  - Short software life cycle because of on going experiments
  - Common understanding, conventions, development techniques were a given
  - XP like programming
  - But unlike XP: Thorough user documentation from the start and maintained
    - -> allowed to discover discrepancies
  - Emphasis on Testing in test beds and as dedicated test in collaboration with the CERN computer center
  - Thorough requirements documents and reviews in fields which were new to the team

12000 lines of code
O(10) packages
CVS, RPM, YUM
5-10 releases per year

The SDP in ALICE Online

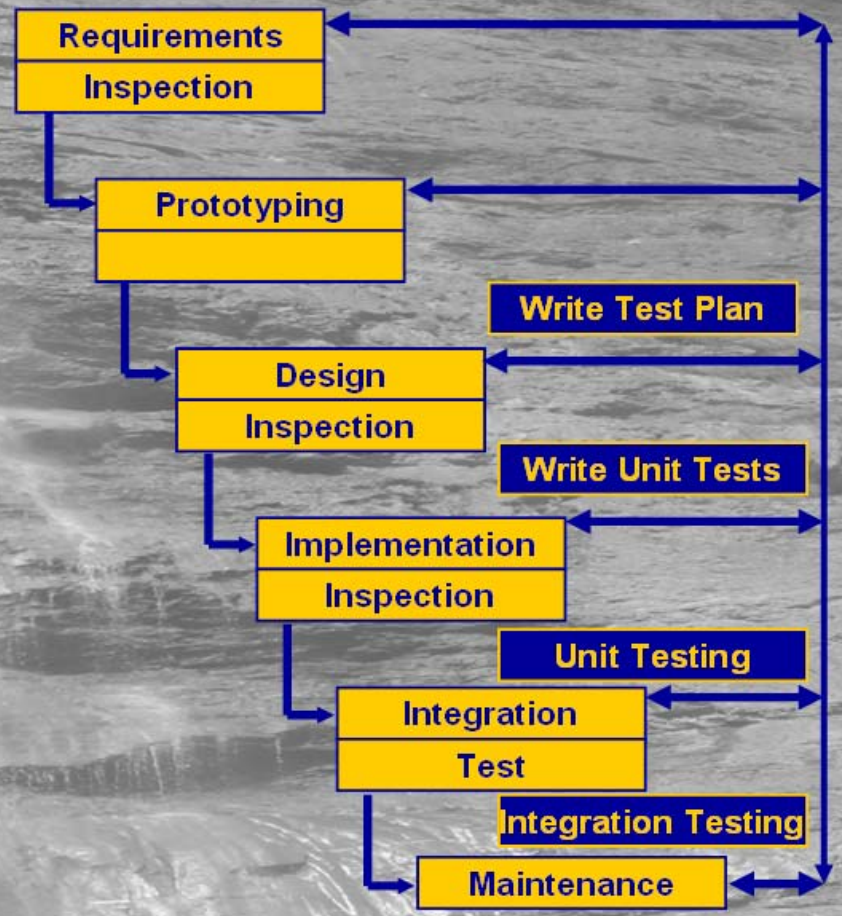
- ALICE detector at the LHC is smaller than ATLAS
- TDAQ has high requirements on the performance of physics data storage
- Development team
  - Team started with 4 very experience developers
  - Familiarity of team members since a decade and a common history
  - Long term experience in development and support
  - Newcomers could be integrated stepwise (up to 10)
  - Based at CERN
  - Immediate task to provide a TDAQ system for related experiments
- Approach
  - Short software life cycle because of on going experiments
  - Common understanding, conventions, development techniques were a given
  - XP like programming
  - But unlike XP: Thorough user documentation from the start and maintained
    - -> allowed to discover discrepancies
  - Emphasis on Testing in test beds and as dedicated test in collaboration with the CERN computer center
  - Thorough requirements documents and reviews in fields which were new to the team

12000 lines of code
O(10) packages
CVS, RPM, YUM
5-10 releases per year

Success: approach matched the team, size and conditions

Testing in the Atlas online SDP

- Component testing with test plan
- Nightly check targets
  - for each component
  - for the integrated system
- Release testing
  - performed by developers and testers
  - starts with infrastructure testing and then includes other sub-systems and application testing
  - on desktop PC, in labs
  - validated on a larger farm
  - patches are applied to the release if necessary
- Integration with external software
- Regular field tests on smaller scale (< 100PC's)
- Large scale & performance tests of the integrated system
- Deployment in commissioning tests



Test Organization

- Chop tests into small units
- Avoid personal test-ware of developer
- Provide test programs which can serve as examples for users
- Standardize on command, output and exit codes
- Make them part of the software repository
  -> follow the evolution of the component
- Allow for flexibility and test boundaries and critical areas
- Use testing tools for code coverage and Memory leak checking
- Use same basic test software for ALL testing aspects
- Document them
- Establish use cases
- Identify critical areas, test boundaries
- Prepare scripts or utilities to group and run them in a selective manner
- Involve non-authors for testing

Functionality
Scalability
Performance
Reliability
Recovery
Regression
Integration
Verification

In Industry: specialized department for Testing; 20% of manpower per project

Large Scale tests

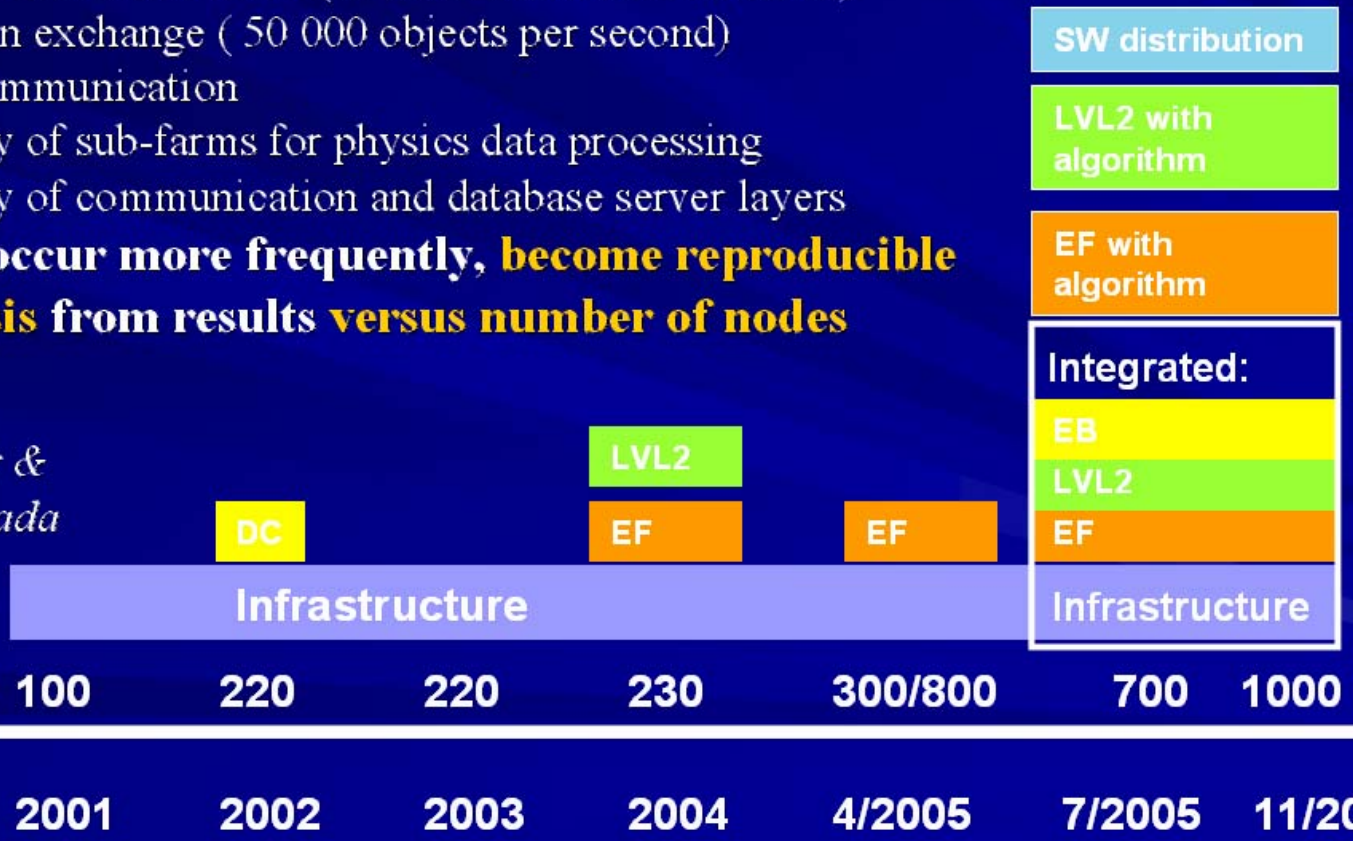
- **Study functionality on large computing farms**
- **Concentrate on operational aspects of the system**
  - Process control and state machine operation
  - Configuration
  - Concurrent database access (25 000 simultaneous clients)
  - Information exchange (50 000 objects per second)
  - Process communication
  - Granularity of sub-farms for physics data processing
  - Granularity of communication and database server layers
- **Rare events occur more frequently, become reproducible**
- **Trend analysis from results versus number of nodes**

CERN computing center &
Westgrid cluster in Canada
1-4 weeks

Large Scale tests

- **Study functionality on large computing farms**
- **Concentrate on operational aspects of the system**
  - Process control and state machine operation
  - Configuration
  - Concurrent database access (25 000 simultaneous clients)
  - Information exchange (50 000 objects per second)
  - Process communication
  - Granularity of sub-farms for physics data processing
  - Granularity of communication and database server layers
- **Rare events occur more frequently, become reproducible**
- **Trend analysis from results versus number of nodes**

CERN computing center & Westgrid cluster in Canada 1-4 weeks



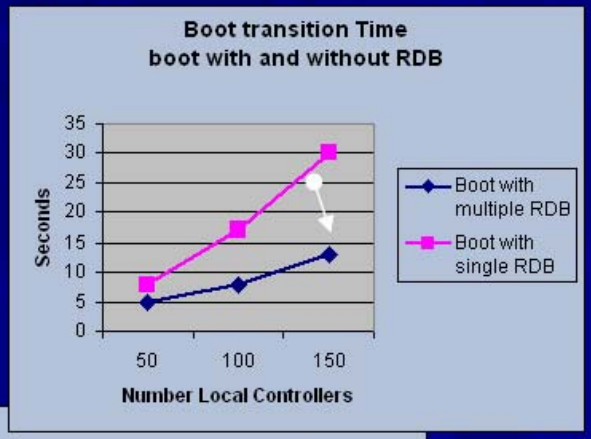
Experience and examples

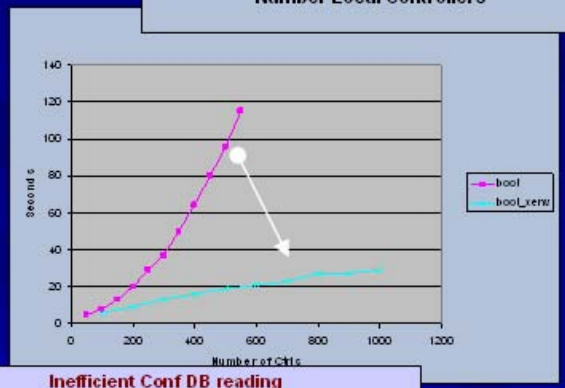
- **Learning process on how to conduct the tests**
- **Number of problems grows significantly with scale (> 500 nodes)**
- **Experienced the need for**
  - **Fault tolerance and stability**
    - Recovery from system failure is too costly in data taking time if having to stop and restart
  - **System monitoring and automatic repair**
  - **Farm management tools and testing tools**

- **Example:**
  Database access is sensitive to scale
  efficient access methods should be used
  - **Use of distributed remote configuration database servers**
  - **Optimize access mode at the application level**

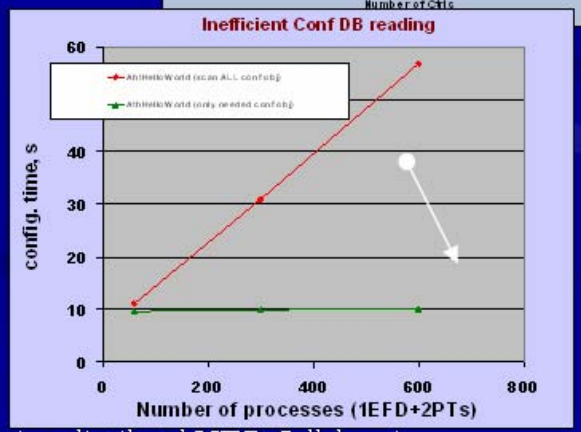
Experience and examples

- Learning process on how to conduct the tests
- Number of problems grows significantly with scale (> 500 nodes)
- Experienced the need for
  - Fault tolerance and stability
    - Recovery from system failure is too costly in data taking time if having to stop and restart
  - System monitoring and automatic repair
  - Farm management tools and testing tools





- Example:
  Database access is sensitive to scale
  efficient access methods should be used
  - Use of distributed remote configuration database servers
  - Optimize access mode at the application level



The Vertical Slice Test Bed

- 80 PCs in the experimental area
  - including final physics data readout hardware,
  - permanently available
- Test conditions close to final ATLAS
  - Network separated form the CERN network
  - Computing hardware
  - Logistics
- Concentrate on the data taking phase
  - Stability of a data taking run
  - Performance
  - Physics event data transport
- Validate simulation of components and subsequently model the full size ATLAS
- Validate technology and implementation choices

Deployment and Status

- **Technical runs and detector commissioning**
  - Organization towards final ATLAS
  - Conducted on the final system
  - Controlled from the ATLAS control room
  - Run for 1-2 weeks, every 1-2 months
  - Work plan, preparation
  - Run coordinator, shifts, e-log book
  - On the job training for shifters to become experts
  - Important activities for team building
  - Efficient occasion to get user feedback and suggestions for enhancements
  - Attractive activity for external collaborators to participate
  - Gives a momentum to people who have worked since many years on the project

Current Status

All TDAQ sub-systems integrated

Readout of all ATLAS detectors integrated

A lot still to be done

Deployment and Status

- Tech
  - dete
    - (
    - (
    - (
      - r
    - I
    - V
    - I
    - (
      - b
    - I
    - I
      - a
- Attractive activity for external collaborators to participate
- Gives a momentum to people who have worked since many years on the project

ATLOG ATLAS - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://pcatdwww.cern.ch/elog/ATLAS/ATLAS/?cmd=List

LST TDAQ RelTest SIMBA doris Monitoring Working G... Phone Dir's nightly myAFS ICALEPCS 2007 ITServices SysAdmin WorkBook

			del Busto						HIGH 3018 RawData; 32 samples, LArgEmecC
4628	15.10.07 17:21	valid	Murrough Landon	Information	LVL1	closed	ROS | LVL1	SW installation: failed attempt to try tdaq-01-08-03	Installed L1Calo SW 00-05-17 to try tdaq-01 ROS firmware upgraded earlier today.
4627	14.10.07 18:26	valid	Henric WILKENS	Information	LAr	closed	Tile | TDAQ | LArg | LVL1	Re: Cosmic muon run 13.10. - 15.10.2007	27060 is running, still some EvtId mismatch between half
4626	14.10.07 17:36	valid	Henric WILKENS	Information	LAr	closed	LArg | LVL1	clock signal not transmited between CTP & LTP	While investigating the corrupted data on EMBA & EMBC I went and had a look at the LTP modules. The 2 barrel LTPs had the BC
4625	14.10.07 17:05	valid	Henric	Fatal	LAr	open	LArg	Barrel Data completely corrupted	All the data in the Barrel is badly corrupt (see plot)
4624	14.10.07 16:20	valid	Henric	Information	LAr	closed	Tile | TDAQ | LArg | LVL1	Re: Re: Cosmic muon run 13.10. - 15.10.2007	Crate EMBC1-01L was missing (visible from monitoring plots)
4623	14.10.07 11:32	valid	Oleg	Information	LAr	closed	Tile | TDAQ | LArg | LVL1	Re: Cosmic muon run 13.10. - 15.10.2007	Run 27050 stopped due to DFM busy at 4060 recorded events. Starting a new run 27053...
4622	13.10.07 20:40	valid	Martin	Information	LAr	closed	Tile | TDAQ | LArg | LVL1	Cosmic muon run 13.10. - 15.10.2007	LAr run conditions: EMBA: Crates I03, I04, I05, I06, I07, I10,
4621	13.10.07 18:29	valid	Martin	Information	LAr	closed	Tile | TDAQ | LArg | LVL1	Re: Re: Cosmic muon run over the weekend 12.10.2007 - 15.10.2007	[quote="Martin Aleksa"][quote="Martin, Oleg Wainar, Murrough, Henric"]LAr run condition
4620	13.10.07 18:17	valid	Henric WILKENS	Information	LAr	closed	LArg	Missing ECRs on EMEC-A solved	ECR seen at ROD level (sent from emec-A to emec-C every second) 75kHz L1A
4619	13.10.07 15:15	valid	Henric WILKENS	Information	LAr	closed	LArg	Performance tests ongoing	We are profitting of the time without FE of last night cooling failure, to perform performance tes
4618	13.10.07 11:34	valid	Martin Aleksa	Information	LAr	closed	Tile | TDAQ | LArg | LVL1	Re: Cosmic muon run over the weekend 12.10.2007 - 15.10.2007	[quote="Martin, Oleg, Wainar, Murrough, Hen run conditions:
4617	13.10.07 01:22	valid	Martin R. Jäkel	Fatal	DSS	closed	DSS | LArg	LAr FE cooling alarm reset	> DSS Alarm Apparently we had a power cut in SE1, which
4616	12.10.07 23:06	valid	Teresa	Warning	LAr	open	LArg	HEC C HV locked after	Due to the cooling failure problem that hap tonight at 10:02 PM

Done pcatdwww.cern.ch

Deployment and Status

- **Technical runs and detector commissioning**
  - Organization towards final ATLAS
  - Conducted on the final system
  - Controlled from the ATLAS c… room
  - Run for 1-2 weeks, every 1-2 …
  - Work plan, preparation
  - Run coordinator, shifts, e-log book
  - On the job training for shifters to become experts
  - Important activities for team building
  - Efficient occasion to get user feedback and suggestions for enhancements
  - Attractive activity for external collaborators to participate
  - Gives a momentum to people who have worked since many years on the project

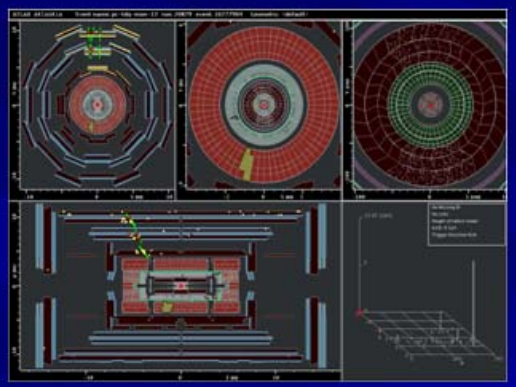
Do's and Don'ts

- **Choose an SDP model which matches most closely**
  - project size and life time
  - experience of team members and their geographical location
- **Build change management into the process**
- **Give importance to building up a project culture**
  - Invest in people
  - -> many benefits are implicit and come automatically
- **Write and review requirements thoroughly**
- **Make testing an integral part of the SDP as early as possible**
  - nightly builds and check targets
- **Use code management and configuration management system from the start**
- **Use project management tools and collaborative tools**
- **Invest in explicit and on the job training**
- **Seek management support and a driving person**

Conclusions

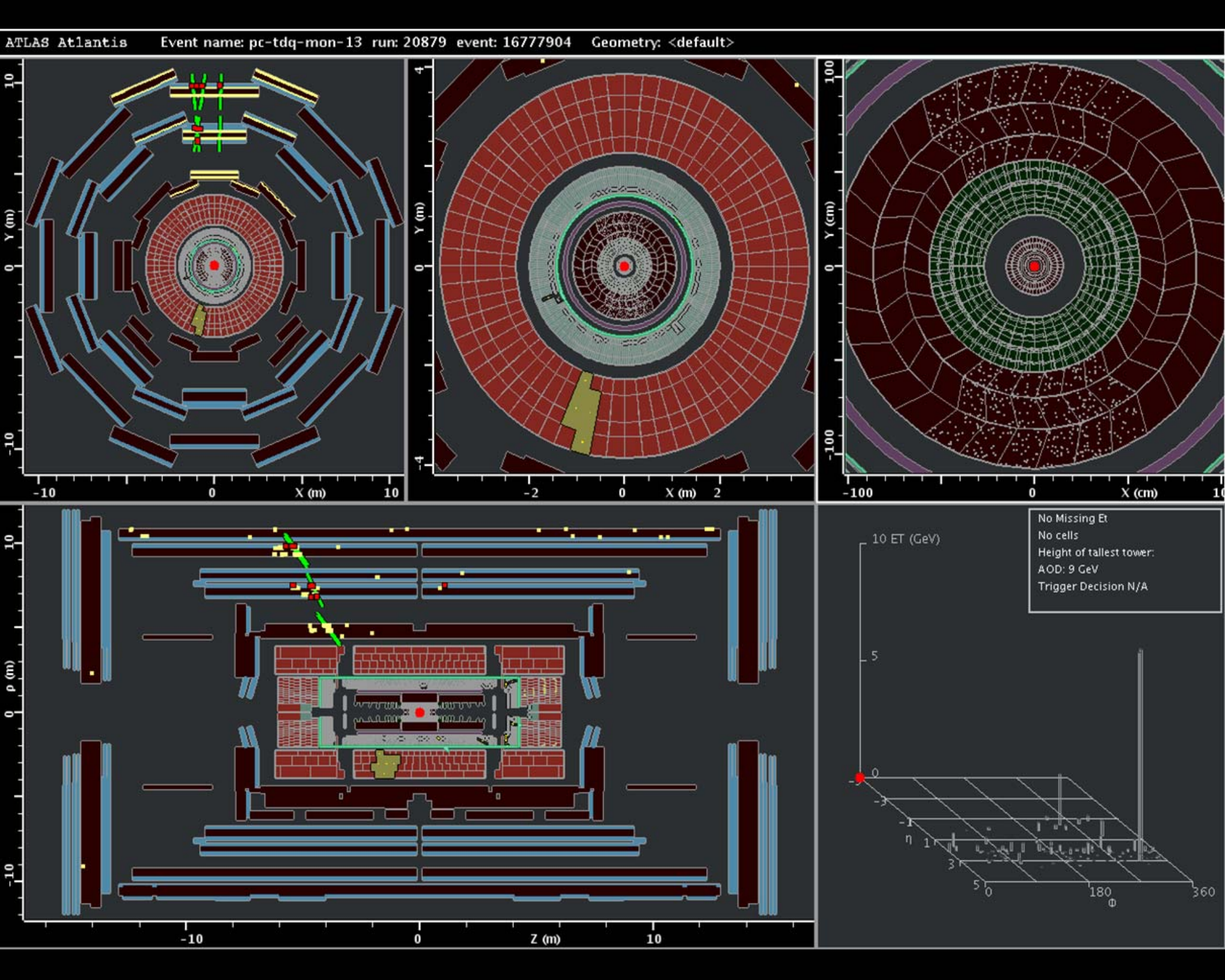
The ATLAS TDAQ project has benefited in many aspects from adapting a flexible SDP framework

Current deployment in detector commissioning activities demonstrates the success of the team and of the approach



Thanks to

- *the ATLAS TDAQ project members and its management*
- *Pierre Vande Vyvre from the ALICE collaboration*
- *Bob Jones who has started the SDP in ATLAS TDAQ*


ATLAS Atlantis Event name: pc-tdq-mon-13 run: 20879 event: 16777904 Geometry: <default>
Y (m)
X (m)
Y (cm)
X (cm)
ρ (m)
Z (m)
No Missing Et
No cells
Height of tallest tower:
AOD: 9 GeV
Trigger Decision N/A
10 ET (GeV)
η
Φ