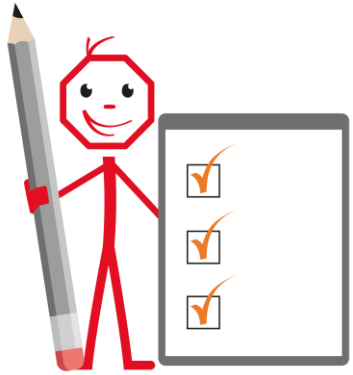

CWG2
Tools, Guidelines and Procedures

16 December 2014

ALICE O^2 Asian Workshop 2014@Pusan

Introduction

- Activities started in March 2013
- Contact: alice-o2-cwg2@cern.ch
- Mandate:
  - Bootstrap common tools for CWGs organization
  - Coordinate/conduct tools surveys
  - Propose common guidelines, policies and procedures

Common tools for CWGs organization

- Main motivation
  - use existing CERN infrastructure
- Proposed following tools:
  - Twiki for notes/documents sharing
    - https://twiki.cern.ch/twiki/bin/viewauth/ALICE/AliceO2
  - JIRA for issue tracking
    - https://alice.its.cern.ch
  - Indico for meetings and minutes
    - https://indico.cern.ch/category/4601/
- Created presentations and reports templates

Tools surveys

- Defined evaluation procedure
- Focused on collaborative/software development tools

Function	Tool
Issue Tracking System	JIRA
Version Control System	Git
Website creation tool	Drupal
Source code documentation	Doxygen
Software build system	Cmake (CWG11)

Tools surveys: JIRA

- Issue tracking system from Atlassian
  - Paid version, hosted by CERN IT

Mandatory Requirements	Details	JIRA
Users	# > 1500	Ok
	Cumulative addition	Ok
Access rights	NICE credentials	Ok
	Integration with e-groups	Ok
	Groups/Roles	Ok
Projects	Multiple	Ok
	Version	Ok
	Sub-projects/Components	Ok
API		Ok
VCS integration		Ok
Customization	Workflows	Ok
Prioritization		Ok
Notifications		Ok
Optional requirements		
Project planning		Ok
Customization	Issue type	Ok
	Issue fields	Ok
	Roles/groups	Ok
Reporting		Ok
Extendibility	Plugins/custom code	Ok
Reminders		Can be implemented

https://twiki.cern.ch/twiki/pub/ALICE/Tools/O2_CWG2_Eval_IssueTrackingSystems.pdf

Tools surveys: Git

- Git: a distributed version control and source code management
  - Ability to store big repositories
  - Client support for multiple operating systems, including SLC
  - Tags, atomic commits,
  - interactive commits, locks, mv/del, binary files, purge, international support etc
  - Hooks, date checkouts
  - Possibility to import from others VCS
  - Possibility to define workflows
  - Distributed version control system: all users act as backup repository, fast operations, easy branching/merging
  - Smaller size than SVN

https://twiki.cern.ch/twiki/pub/ALICE/Tools/ALICE_O2_CWG2_Version_Control_Systems_eval.pdf

Tools surveys: Drupal

- Website creation tool, open source
  - Hosted by CERN IT (currently Drupal 7)

Mandatory Requirements	Details	Drupal
Editing	WYSIWYG	OK
	Direct HTML/markup language	OK
	Dynamic content creation	OK
Structure	Hierarchy	OK
	Navigation	OK
Search	Efficient	OK
Revision Control	Change history	OK
	Diffs	OK
	Watch pages	OK
Look & Feel	Themes	OK
	Attractive by default	OK
Access Control	Role-based	OK
	Per page	OK
CERN Auth. integration	NICE credentials	OK
Import external content		OK
Print	Printer-friendly version	OK
Optional requirements		
Revision Control	Resolve conflicts	OK

https://twiki.cern.ch/twiki/pub/ALICE/Tools/O2_CWG2_Eval_Websites.pdf

Tools surveys: Doxygen

- Source code documentation tool
  - Documentation embedded in source code
  - Support for many output formats: HTML, LaTeX, Man pages, RTF, XML, Docbook, etc.
  - Attractive output
  - Flexibility in formatting, rich syntax
  - Simplicity

https://twiki.cern.ch/twiki/pub/ALICE/Tools/O2_Eval_Source_code_documentation.pdf

Tools surveys: CMake

- Software Build System

Mandatory requirements	Autotools	CMake	GYP	SCons
Incremental builds	Yes	Yes	Yes	Yes
Parallel support	Yes	Yes	Yes	Yes
Dependencies handling	Yes	Yes	Yes	Yes
Error tracking and reporting	Yes	Yes	Yes	Yes
Cross compilation	Yes	Yes	Yes	Yes
Multiple builds	No	Yes	No	Yes
Multiple platforms	Yes	Yes	Yes	Yes
Configurability	Yes	Yes	Yes	Yes
Optional requirements	**Autotools**	**CMake**	**GYP**	**SCons**
Generate build configuration for different IDEs, such as XCode or Visual Studio	No	Yes	Yes	No

https://twiki.cern.ch/twiki/pub/ALICE/Tools/ALICE_O2_C_Software_Build_System-12-11-2013.pdf

Guidelines, Policies and Procedures

- C++ Coding Guidelines
  - Based on Google Coding Guidelines
  - 3 documents
    - Coding Guidelines
      - e.g. #define guards, namespace usage, variables initialization, virtual functions, casting, Exceptions, etc.
    - Naming and Formatting
      - e.g. files, classes, functions and variables names, spaces vs tabs, line length, etc.
    - Comments
      - e.g. mandatory documentation, Doxygen

Guidelines, Policies and Procedures

- Software License
  - Work in progress
  - Leaning towards GPLv3
  - Looking at recommendations from CERN Task Force
    - https://legal.web.cern.ch/licensing/software

Future

- Finalize software license
- CWG2 will then enter in ‘hibernation’
- Nevertheless continuing to follow:
  - Tools evaluations
  - Feedback on published tools and guidelines, particularly the C++ Coding Guidelines