

Improve ATLAS Software

Quality

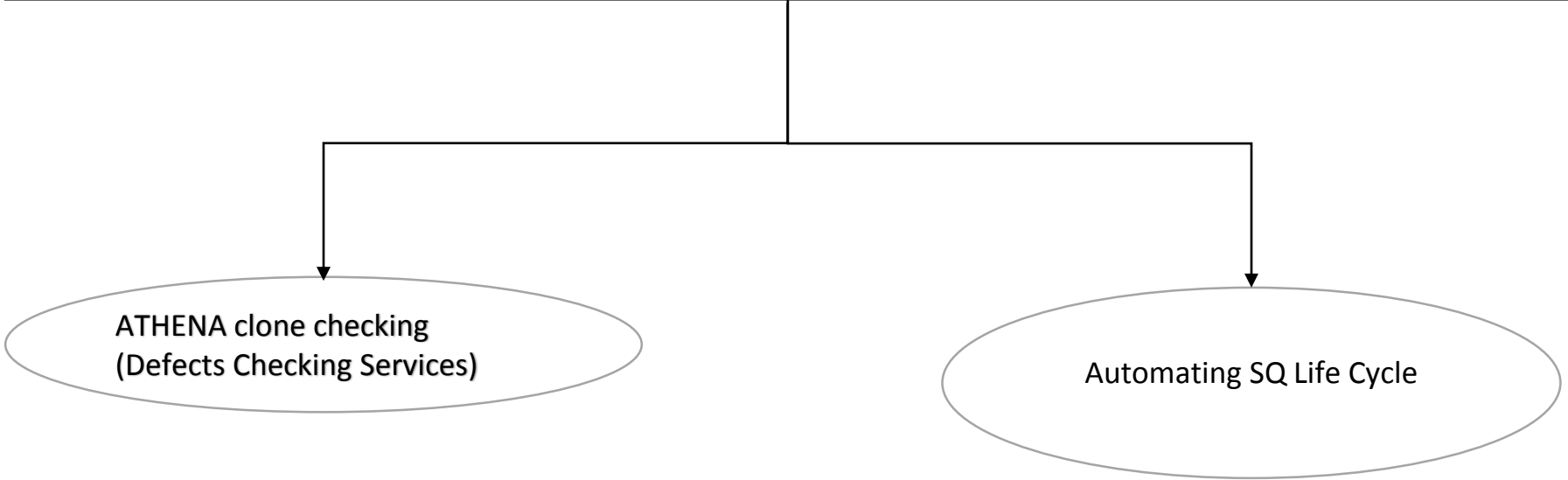


Pirtskhalava Mariami

Georgian Technical University




Software Quality evaluation requires our work in two directions



Defect Checking Services

Weekly activities of Georgian Team for COVERITY Scan:


1. Analyze status of ATHENA Scan on voatlas405
2. Preparation of Jira Ticket for scan
3. Getting defects from voatlas405
4. Finding authors on Git
5. Grouping defects
6. Preparation of Jira Tickets for defects

[ATLAS Software Quality](#) /  [ATLASSQ-98](#)
[Coverity Defects Report PhysicsAnalysis package](#)

Issue Type:  Bug

Assignee: [Andrew John Washbrook](#)

Created: 30/Nov/18 1:30 PM

Priority:  Minor

Reporter: [Mariam Pirtskhalava](#)

Coverity Defects Report

Package: Trigger

1. CID: 121382

Defect Type: Parse recovery warning

First Detected: 27/11/2018

File: /PhysicsAnalysis/D3PDMaker/TrackD3PDMaker/src/TrackIsolationFillerTool.h

Author: charles leggett <leggett@cern.ch>

Last Modified: 12/11/2018

1. CID: 121381

Defect Type: Parse recovery warning

First Detected: 27/11/2018

File: /PhysicsAnalysis/D3PDMaker/D3PDMakerUtils/D3PDMakerUtils/MultiAssociationToolMulti.icc

Author: Scott Snyder <scott.snyder@cern.ch>

- Example of Jira Ticket for Successful Scan

Coverity Scan Report	24/09/2018			
	Overall:			
	Name	Status	Size	Items
	Main folder	Complete	210,5 GB	417'81
	Coverity Build	Done		
	Coverity Analyze	Done		

Build History:					
Name	Instalation	Faults	File		
Status	Strings	Status	Strings		
Athena Build			Completed Successfully		/main/build_atheana.out
External Build			Completed Successfully		/main/buildexternals.out

Analyze history:

Status Done
 Files Analyzed 35788
 Defects Found 60084
 Analyze Time 13h00

- Example of Jira Ticket for Unsuccessful Scan

Name	Status	Size	Items
Main folder	Not Complete	7.0Gb	163'932
Coverity Build	Not Done		
Coverity Analyze	Not Done		

Coverity Scan Report 21/06/2018

Build History:					
Name	Instalation	Faults	File		
Status	Strings	Status	Strings		
Athena Build			Configuring incomplete, errors occurred!	#35	/main/build_athena.out
External Build			CMake Error at Coin3D-stamp/download-Coin3D.cmake:159 (message): make[2]: *** [src/Coin3D-stamp/Coin3D-download] Error 1 make[2]: Target `External/Coin3D/CMakeFiles/Coin3D.dir/build' not remade because of errors. make[1]: *** [External/Coin3D/CMakeFiles/Coin3D.dir/all] Error 2	#7'607	/main/build_externals.out
		[94%] Built target Package_Gdb make[1]: Target `all' not remade because of errors. make: *** [all] Error 2 make: Target `default_target' not remade because of errors.	#247'771		/main/build_externals.out

Analyze history:

Status	Not Done
Files Analyzed	0
Defects Found	0
Analyze Time	-

Since June 2018 Georgian team was providing Jira Tickets every week, but ATLAS SOFTWARE COMPUTING decided to stop the usage of Coverity, so we moved to CPPCHECK.

Cppcheck is a static analysis tool for C/C++ code. It provides unique code analysis to detect bugs and focuses on detecting undefined behaviour and dangerous coding constructs.

General Objectives:

- Run CPPCHECK for regular scanning process
- Get CPPCHECK defects
- Finding authors on Gitlab
- Grouping defects
- Preparation of Jira Ticket for defects

- We run application locally
- It takes 1h25 for full Athena repository scanning and 35min only for SIMULATION folder scan
- There are 519 defects for Simulation folder
- We have made defect reports for Simulation folder, sent reports to authors and got feedbacks from authors.

Hi Mariam,

Thanks for the report - we already have a ticket open on the new PixelFastDigitizationTool defects here:
<https://its.cern.ch/jira/browse/ATLASSIM-3826>

Automating the creation of JIRA tickets for new defects is a great improvement though! Thanks for doing this.

Cheers,

John

Hi Mariam,

Thanks for this report. Here are a few comments:

- Which branch of the software repository is this report about? I guess master?
- It seems like the defect messages are too brief. Perhaps including a print out of the problem line would be useful.
- Perhaps including a link to the problem line in gitlab would be most efficient?

Cheers,

John



```
root@coveritypc-VirtualBox:~/Desktop/cppcheck/athena2# git clone https://gitlab.cern.ch/Atlas/athena.git
Cloning into 'athena'...
remote: Enumerating objects: 742171, done.
remote: Counting objects: 80% (593737/742171)
```


Automating SQ Life Cycle

All the work, I described above, is massively done manually and we are developing automated procedures by using git commands and shell scripts. Here is a little part of the automatisisation script :

```
#!/bin/bash
```

```
echo 'type 1st file path'
read Fpath
echo 'type 2nd file path'
read Spath
echo 'type where to get output'
read Rpath
xmllint --c14n $Fpath > $Fpath+"1"

echo 'first file is canonical'
xmllint --c14n $Spath > $Spath+"1"

echo 'Second file is canonical'

diff $Fpath+"1" $Spath+"1" > $Rpath

echo ' difference is found '
```

```
#!/bin/bash
cppcheck --template="{file},{line},{message}" --xml /home/coveritypc/Desktop/cppcheck/Simulation 2> testauto.xml
k= grep -c <error testauto.xml

awk -F\" 'BEGIN {for (int i=0;i<k;i++) print "file:",$2,"line:",$4,"message:", $10}' testauto.xml > testauto2.txt
sed '1d' testauto2.txt
```