# Introduction

Since few years SAP has developed object orienting approach very much and this is recommended way of developing new code now. There is also framework provided for writing and executing automatic unit tests which validates functionality behaviour on code level (units, methods, blocks of code). In Statoil ERP DEV there is policy of developing ABAP programs according to clean code rules, including unit tests creation when it is possible. Unfortunately there is missing framework for automatic daily tests execution and results evaluation. We came to conclusion that such framework would be highly desired. Daily tests run and presenting results is today considered as mandatory thing in agile development. Continuous Integration idea is becoming nowadays a standard. That is why we want to develop framework that will use SAP automated tests execution, but will present results in a better way, through Hudson server which is dedicated for Continuous Integration and testing issues.

Enhancement is needed to standard code regarding ABAP unit tests executions. It will not affect system in quality or production where unit tests execution is disabled by default.

# Analyses

Framework for testing needs to be built in 4 steps approach:

1. Run unit tests

* Build report that will run unit tests for all defined classes, like custom development that starts with Z or Y.
* It must be possible to run it at night as the batch process.
* Set of programs/classes to test should be configurable by input parameters.

1. Save tests results to files

* Standard SAP unit tests framework will be enhanced to export data to file on local SAP disc.
* New dedicated directory may be created for that purpose.
* Data needs to be formatted as XML according to JUNIT XML file standard so Hudson server can later import and present results.
* Enhancement cannot impact normal unit tests execution for developers.
* Tests result (number of fails, success, total tests) and code coverage will be exported to files.

1. Export files to Hudson server.

* After files are created locally in SAP they must be exported to Hudson.
* Hudson server address must be configurable.
* FTP / shared folder are options for files transfer.

1. Notify Hudson and present results.

* There will be jobs build in Hudson framework that will discover new tests result files in specified directories and based on that process files and present them in graphical form.
* It is possible to configure Hudson job by checking directory and run project if proper file is found.
* Hudson can present results from single and many builds for tests fail/pass and code coverage.

Code updates are needed only for points 1-3. Point 4 is separate Hudson configuration task that does not require changes in SAP.

# Solution details

Development is created in package ZCAGS\_CI (CI is the shortcut from Continuous Integration).

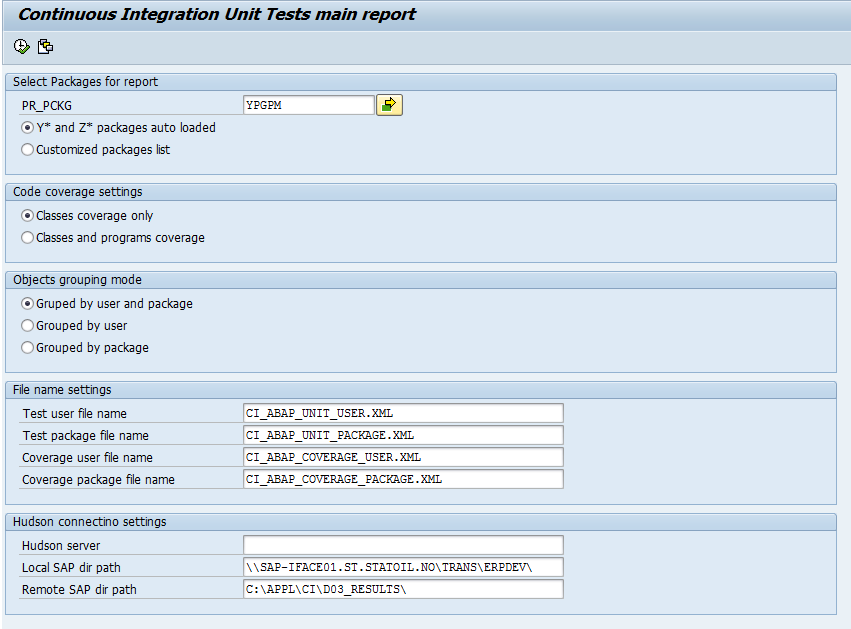
Chapters below describe details of final framework implementation.

1. zcags\_ci\_report

Report allows user to run Unit Tests for programs/classes and export results to XML file. It runs all Unit Tests from given packages, measure code coverage for found classes and generate files in JUNIT XML format, so result can be imported to Hudson server.

Program is generating XML files in JUNIT format as output. These files can be used to visualize Unit Tests results and code coverage values. Hudson Continuous Integration server is used to parse files and visualize results. This program can be scheduled as batch job. There is an option of running report from HTTP request through zcl\_cags\_ci\_http\_req\_handler.

Selection parameters:



* Default selections finds all Y\* and Z\* packages for which report will be run and unit tests measured.
* If we want to specify packages manually we need to change option to “Customized packages list”, delete already existing packages loaded into selection dialog and put own packages.
* By default code coverage will be measured only for classes and ignored for programs (“Classes coverage only” option). The fact is that we do not often need to have unit tests on program level if logic is built into classes. In addition, code coverage with programs is very low (about 0.2 % for lines). With coverage on classes level only we can focus on improvement and see results (current code coverage about 6 % for lines).
* By default both result variants will be generated – for user and package level
* File names for different results must be specified
* Files will be generated in directory specified as “Local SAP dir path” parameter, default:

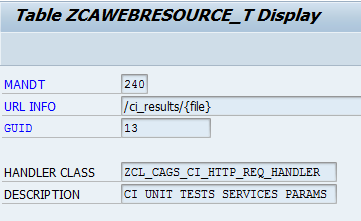
\\sap-iface01.st.statoil.no\trans\ERPDEV\

* Remote SAP dir path - a path on local PC where files will be downloaded if report is run in dialog SAP logon and not background mode.
* Parameter “Hudson server” is currently not used.

1. Hudson integration

To be able to trigger test run from Hudson server, HTTP request handler has been established on SAP side. There was already interface ZIF\_RESOURCE\_HANDLER created before that allows to publish HTTP request listener whcih can react on HTTP queries and run actions on SAP. This has been used for final integration as final effective solution.

zcl\_cags\_ci\_http\_req\_handler is class responsible for integration between Hudson and SAP serer. Table ZCAWEBRESOURCE\_T has been extended with new entry:



It is enough to trigger URL now to run unit tests:

<http://sapq152.statoil.no:8014/sap/bc/resources/ci_results/JOBRUN>

or

<http://sapq152.statoil.no:8014/sap/bc/resources/ci_results/JOBRUN_D87>

D83 server is used for both systems trigger, because D87 line has no ZIF\_RESOURCE\_HANDLER feature implemented.

When JOBRUN is called, actually the method

zcl\_cags\_ci\_http\_req\_handler=>zif\_resource\_handler~handle\_get is run.

Inside handle\_get, there is function ZCAGS\_CI\_RUN\_TESTS\_BATCH responsible for running unit tests in background mode in current server or D87 ('D87CLI275' RFC destination).

Job is run through method:

lo\_ci\_report->run\_unit\_tests\_default\_variant( ).

There are also other supportive functions:

* DELLOCK which deletes lock file used to block subsequent test runs if there is already test running in background.
* DELFILES – deletes XML files with results

If command is not recognized, system will try to read file from default SAP location and append name that is given as parameter. Example reading of XML file result is done through query:

<http://sapq152.statoil.no:8014/sap/bc/resources/ci_results/D87_CI_ABAP_UNIT_USER.XML>

All such HTTP requests are built into Hudson project which:

* Runs job in development system
* Every 30 seconds checks if CI\_ABAP\_UNIT\_USER.XML file exists (with prefix D87\_ or D83\_).
* If file is found then DELLOCK request is called
* Other 3 XML files are downloaded.

To be able to read file content, wget has been used. It has given two parameters:

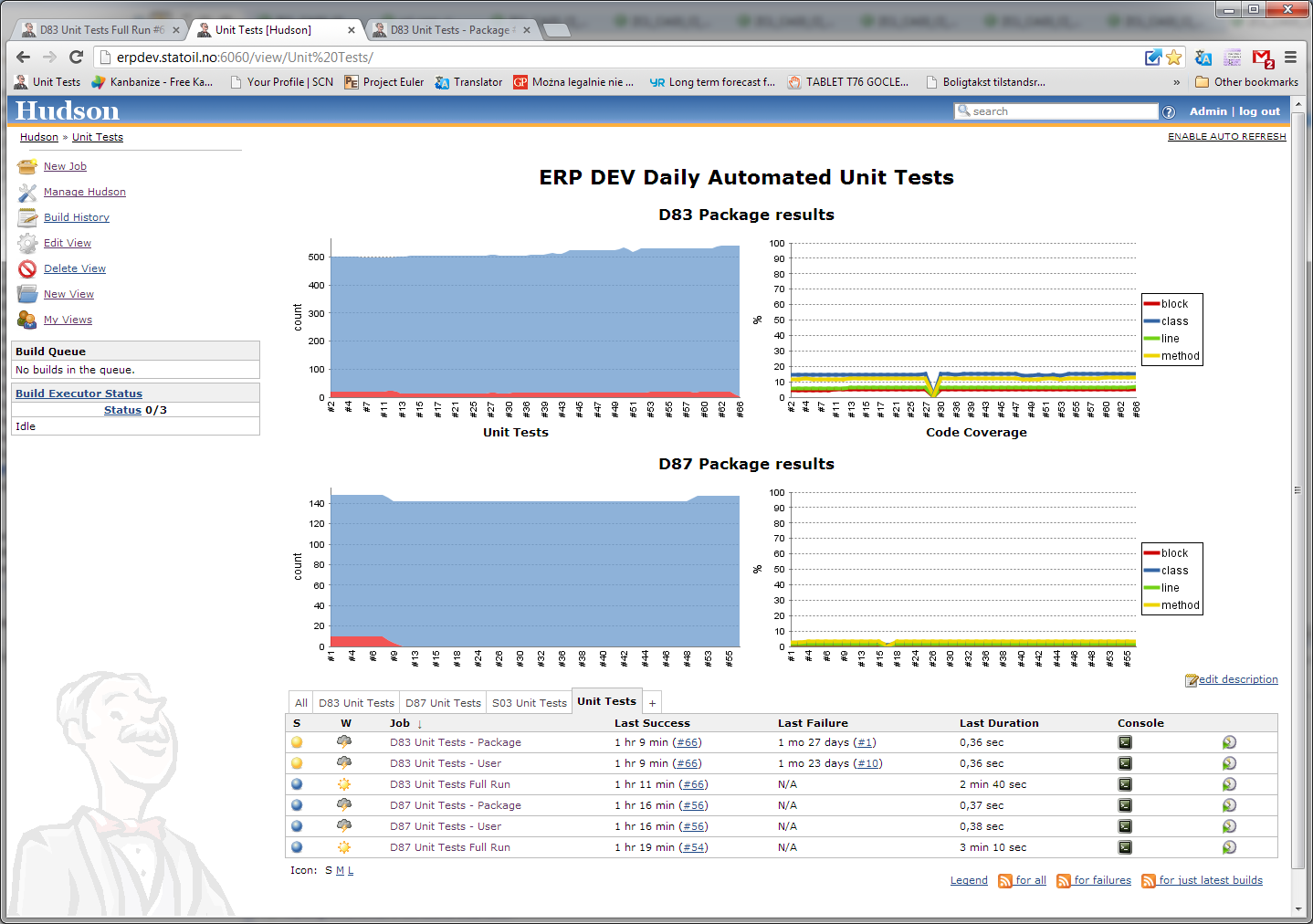
--http-user = SAP user id

--http-password – SAP password.

Thanks to this wget can display file content and save it to local disc by this long command:

C:\appl\WorkTools\wget\wget --secure-protocol=auto --http-user=sap\_user --http-password=xxxxx http://sapq152.statoil.no:8014/sap/bc/resources/ci\_results/D83\_ CI\_ABAP\_UNIT\_USER.XML -O C:\appl\ci\d83\_results\CI\_ABAP\_UNIT\_USER.XML

Hudson view presents tests results:



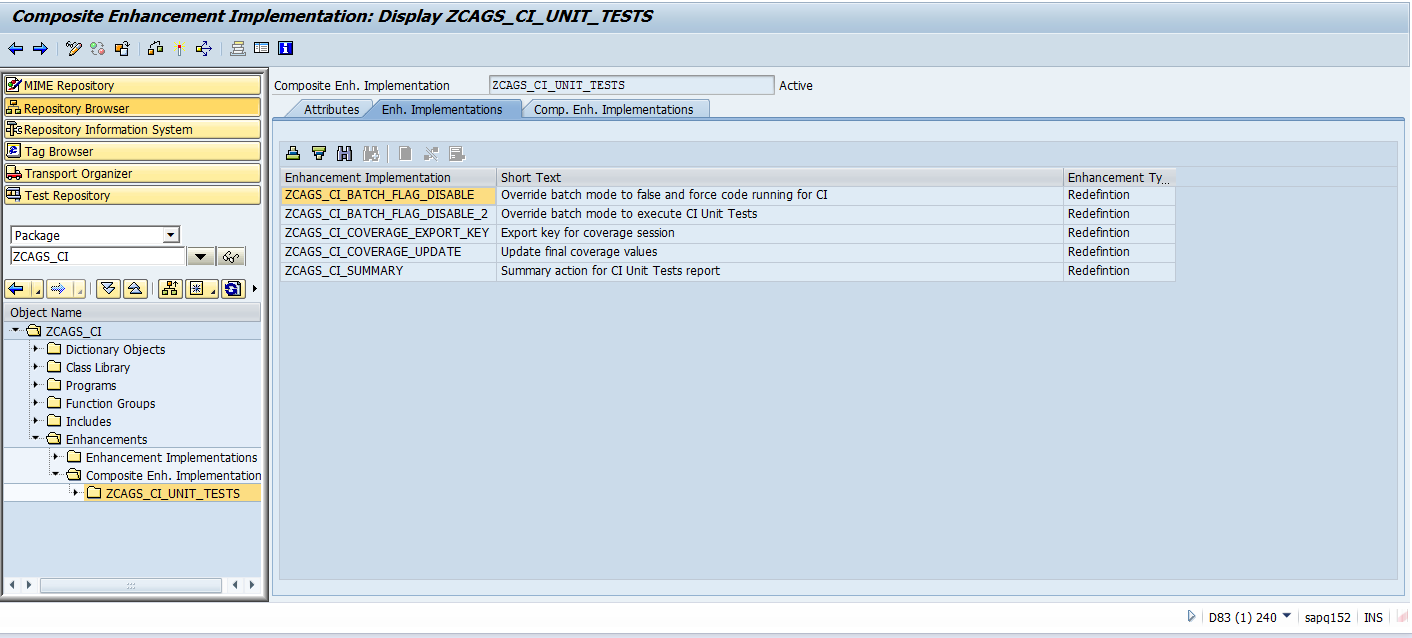
Jobs explanation:

* D83 Unit Tests Full Run – job which triggers unit tests in D83 system for HTTP\_DEFAULT variant of job ZCAGS\_CI\_REPORT. Then files are downloaded to local Hudson directory.
* After previous job is finished, “D83 Unit Tests – Package” and “D83 Unit Tests – User” are run. These jobs process downloaded previously XML files and visualize results into graphs and statistics.
* Analogue jobs are created fir Unit D87 system.

1. Unit tests and code coverage measurements

Main report ZCAGS\_CI\_REPORT is based on standard SAP program rs\_aucv\_runner which allows user to run unit tests for defined list of packages. ZCAGS\_CI\_REPORT is program build on top of standard SAP that extends select options with directories and default listed Y\* znd Z\* packages list. Class that handles report logic is ZCL\_CAGS\_CI\_REPORT.

To be able to retrieve results from standard SAP program RS\_AUCV\_RUNNER, few implicit enhancements have been defined. They are all put into composite enhancement implementation:

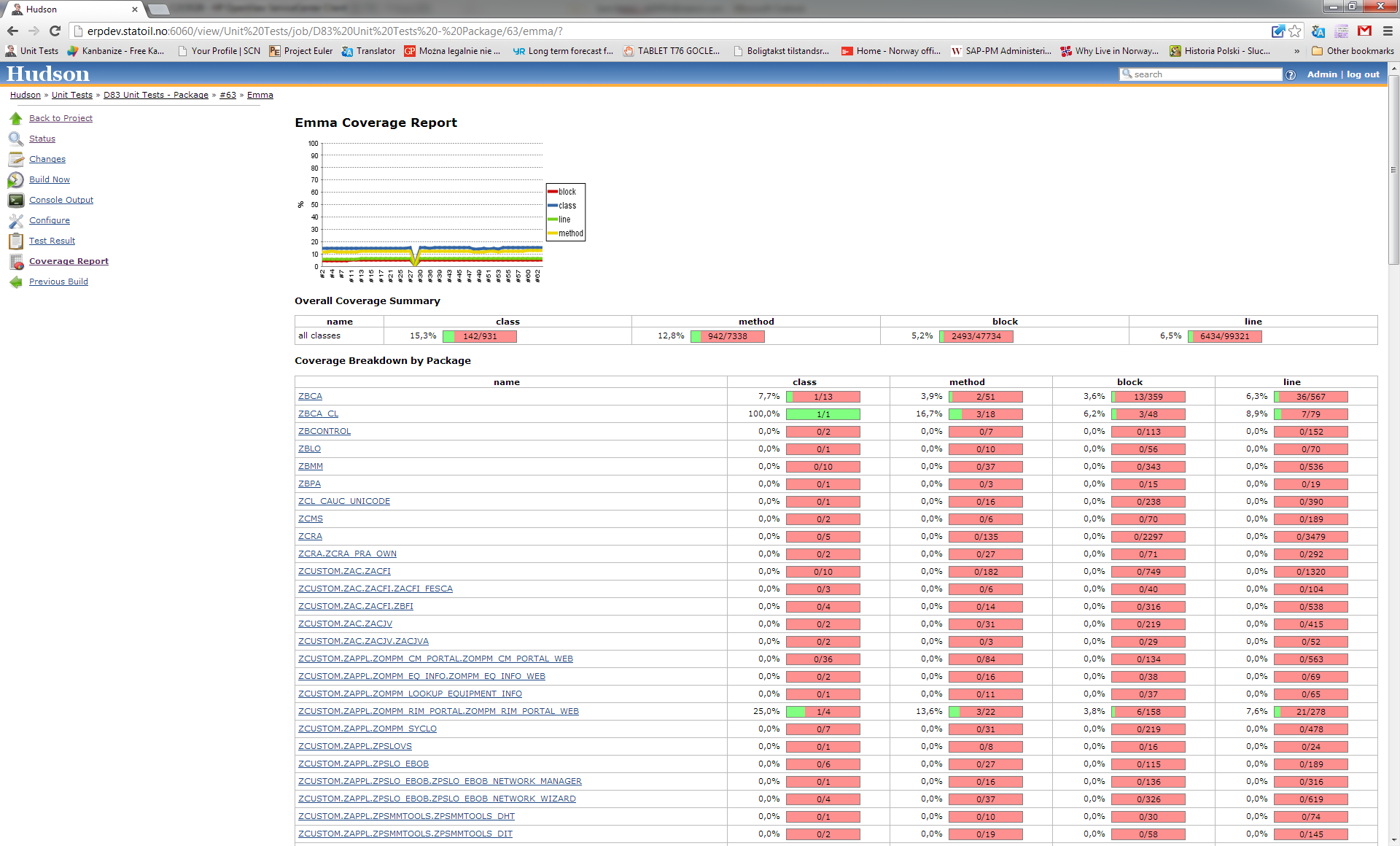


Some of implicit enhancements calculate logic, some of them switch of sy-batch flag so report can be also run in background and code coverage results are exported.

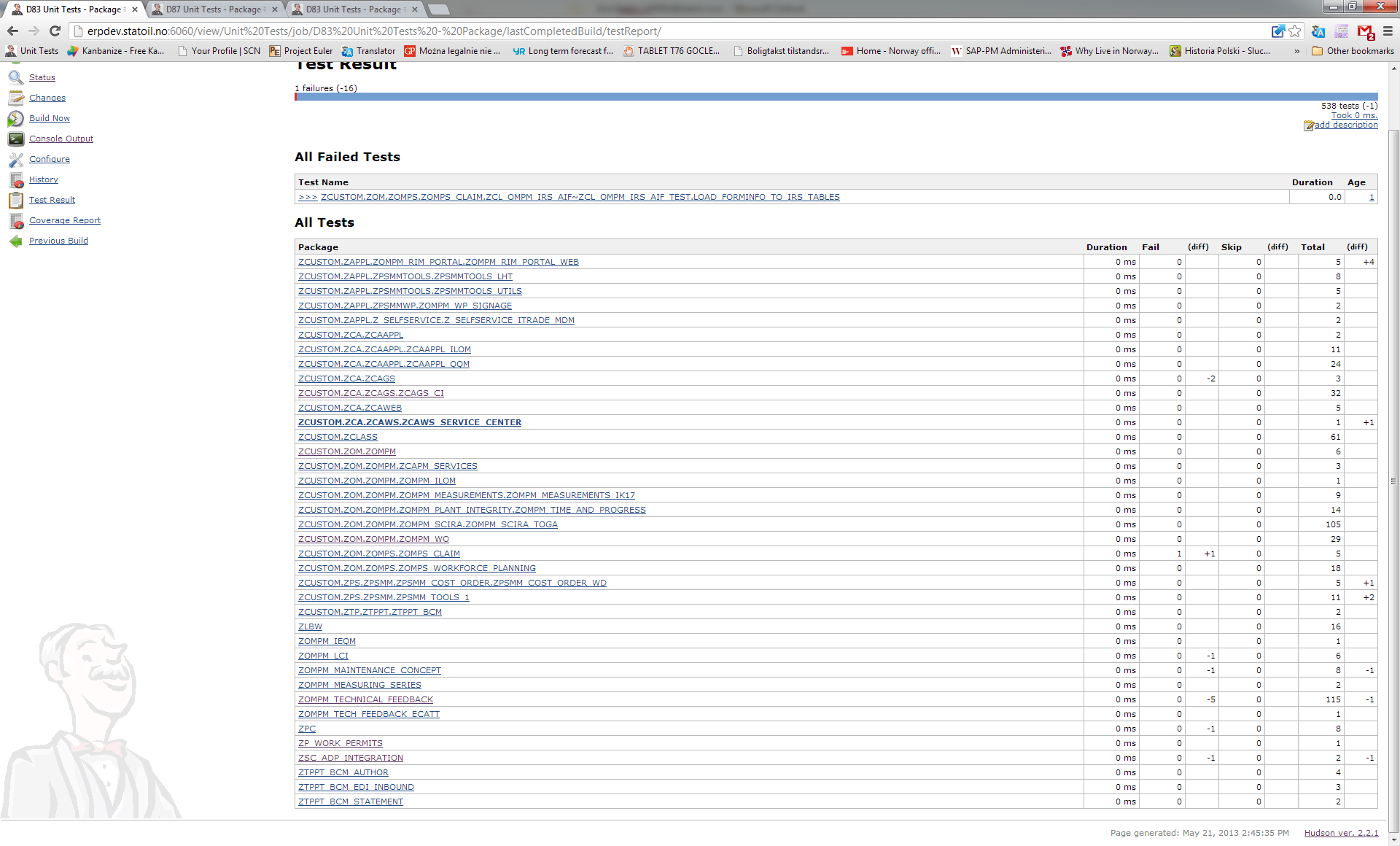
1. Hudson results details

We can see code coverage and unit tests results from Hudson server, on different filter level – from package or user perspective.

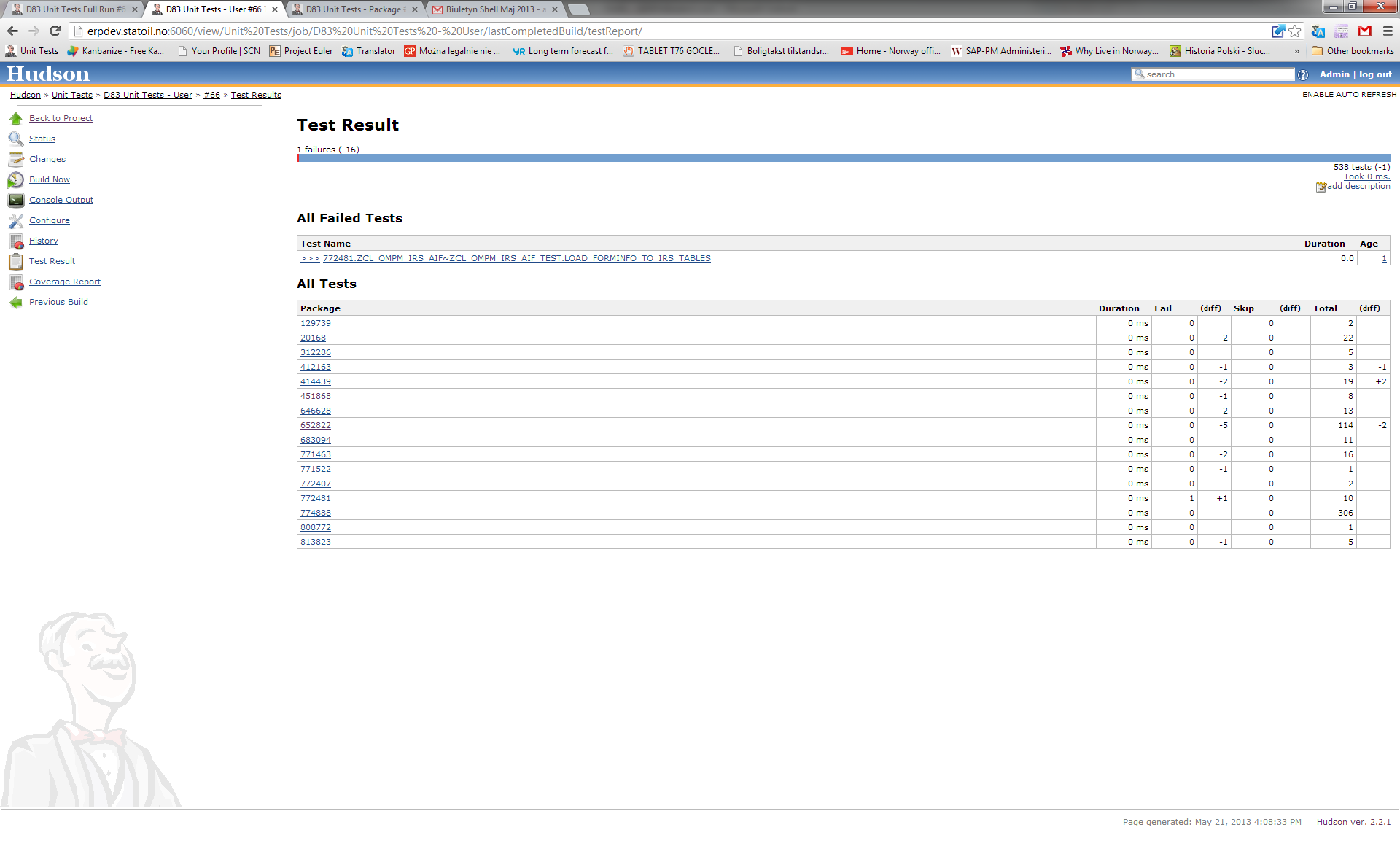
Example code coverage for D83:



Example unit tests on package level D83:



Example unit tests on user level D83:



User level is just approximate view. We cannot differentiate who has created which test method that is why results are approximated – the user who has created class is considered as the owner of all testcases from that class, even if someone else has changed test class contents. That is why it is just overview. It may be used for personal tracking of improvements.

# Objects description

Classes:

ZCL\_CAGS\_CI\_ABAP\_UNIT\_ASSERT – additional utility class for common unit tests methods

ZCL\_CAGS\_CI\_CODE\_COVERAGE – class that collects information about code coverage

ZCL\_CAGS\_CI\_HTTP\_REQ\_HANDLER – HTTP request handler (listener) that is integrating Hudson server with SAP server

ZCL\_CAGS\_CI\_METADATA\_DAO – data access object for searching metadata like user of class, package hieararchy etc.

ZCL\_CAGS\_CI\_METADATA\_DAO – mocked version of DAO used for tests

ZCL\_CAGS\_CI\_REPORT – main class handling report

ZCL\_CAGS\_CI\_TEST\_DATA – class collecting information about tests statistics – successful / failure cases

ZCL\_CAGS\_CI\_TEST\_XML\_CREATOR – class that can build XML in JUNIT format for unit tests results.

ZCX\_CAGS\_CI\_REPORT\_EXCEPTION – exception used for some error handling

Programs and includes:

ZCAGS\_CI\_SCREEN\_HANDLER – local class that handles main program logic

ZCAGS\_CI\_SEL\_SCREEN – screen input parameters definition

ZCAGS\_CI\_TOP – declaration of global parameters used in program

ZCAGS\_CI\_REPORT – main report

ZCAGS\_SAUNIT\_TEST\_FROM\_IDE – copied and modified subprogram used by SAP standard while unit tests execution. This copy is created to avoid batch job mode force, as we need to be able to run unit tests in background and retrieve tests and code coverage results.

Database table updated:

ZCAWEBRESOURCE\_T – new entry added