

Unit Testing and Code Coverage Assessment with SASUnit

- Key Technologies for Development of reliable SAS Macros -

- **HMS Analytical Software is a specialist for Information Technology in the field of Data Analysis and Business Intelligence Systems**
- **Profile**
 - 40 employees in Heidelberg, Germany
 - SAS Institute Silver Consulting Partner for 14 years
 - Doing data oriented software projects for more than 20 years
- **Technologies**
 - Analytics and Data Management: SAS, JMP, R, Microsoft SQL Server
 - Application Development: Microsoft .NET, Java

Our IT Services for the Life Science Industry (SAS, JMP and R)



- Independent Consulting
- Programming
- Data Management
- Training and Individual Coaching
- Application Development and Integration
- Software Validation



Two everlasting questions..

- .. in software development:
- „Are we building the right product?“
(-> Validation)
- „Are we building the product right?“
(-> Verification)

Supporting tools for SAS
developers?

Overview

- **Unit Testing and SASUnit**
- **Test coverage assessment with SASUnit**
- **Conclusion**

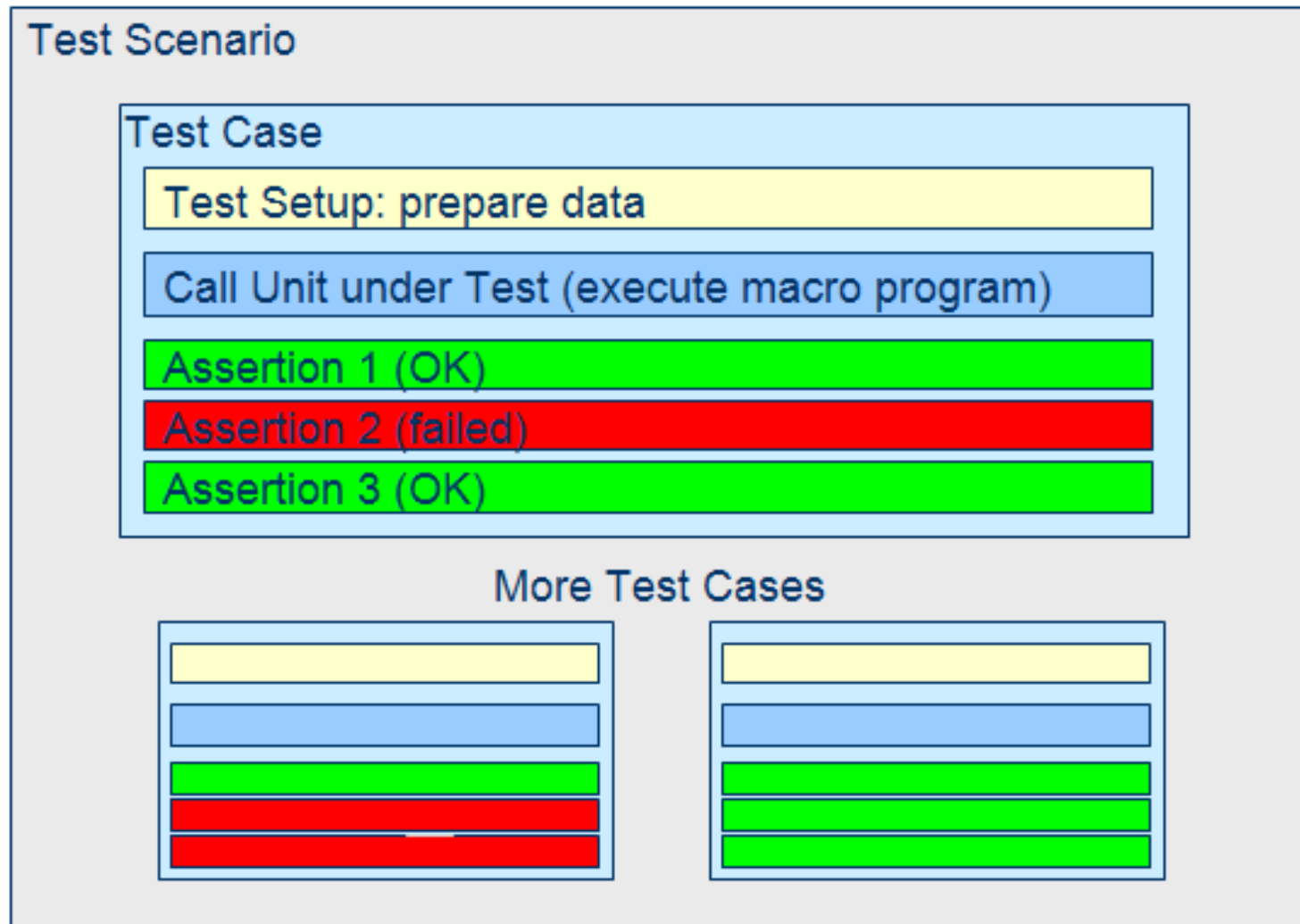
Unit Tests - Concept

- **Testing starts already during programming**
- **Tests are implemented as executable source code**
- **At least one test for every software unit:**
 - static test data
 - execution of the unit under test
 - asserts
- **Test protocol is generated automatically and summarizes test results**

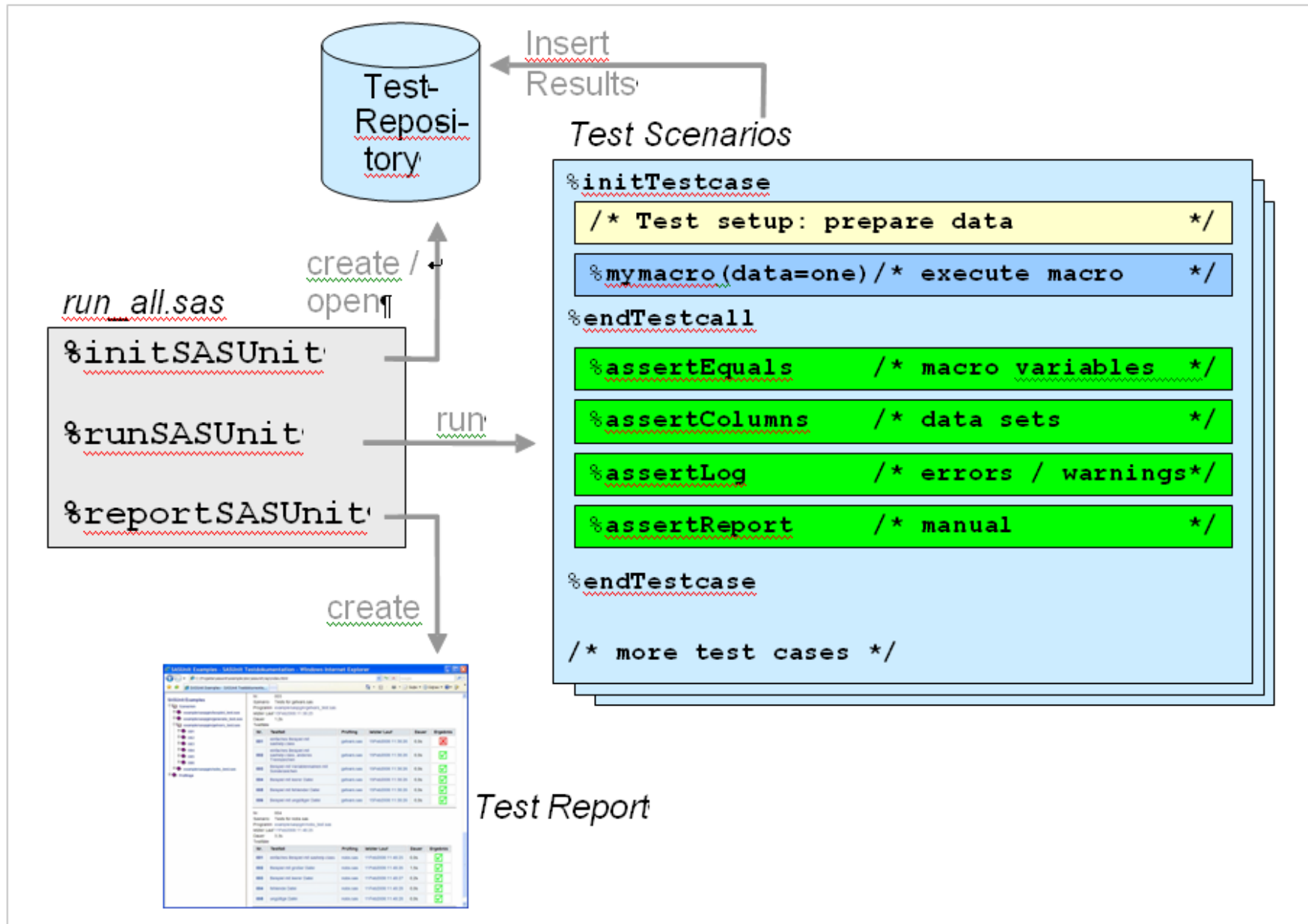
SASUnit – Unit testing for SAS

- Write test scenarios and test cases for SAS programs as SAS code
- Use assertions to test the values of macro variables, the contents of SAS data sets, the existence of external files or the presence or absence of log messages
- Run tests and generate test documentation in batch mode
- Get clearly arranged test documentation, integrated with program documentation
- Integrate non-automatic tests, for instance visual checks of report output
- Written purely on the basis of SAS macros and a few shell commands
- Available for SAS 8.2, 9.1 and 9.2 at sourceforge.net

Structure of unit tests



Usage of SASUnit - Overview



Test results report example

The screenshot displays the SASUnit Test Documentation web application in a Windows Internet Explorer browser. The browser's address bar shows the URL `C:\projekte\sasunit\example\doc\sasunit\rep\index.html`. The application has a navigation pane on the left with a tree view containing 'Scenarios' and 'Units under Test'. The 'Scenarios' list includes `example/saspgm/boxplot_test.sas`, `example/saspgm/generate_test.sas`, `example/saspgm/getvars_test.sas`, `example/saspgm/nobs_test.sas`, and `example/saspgm/regression_test.sas`. The 'Units under Test' list includes `example/saspgm`, `boxplot.sas`, `generate.sas`, `getvars.sas`, `nobs.sas`, and `regression.sas`. The main content area features a tabbed interface with 'Main Page', 'Test Scenarios', 'Test Cases', and 'Units under Test'. The 'Test Scenarios' tab is active, displaying a table of test results. The table has columns for 'No.', 'Test Scenario', 'Program', 'Last Run', 'Duration', and 'Result'. It lists five test scenarios, with results ranging from failed to passed. At the bottom of the table, a footer indicates the report was generated on Monday, 8 September 2008, at 16:45:01 by SASUnit Version 0.904 (38).

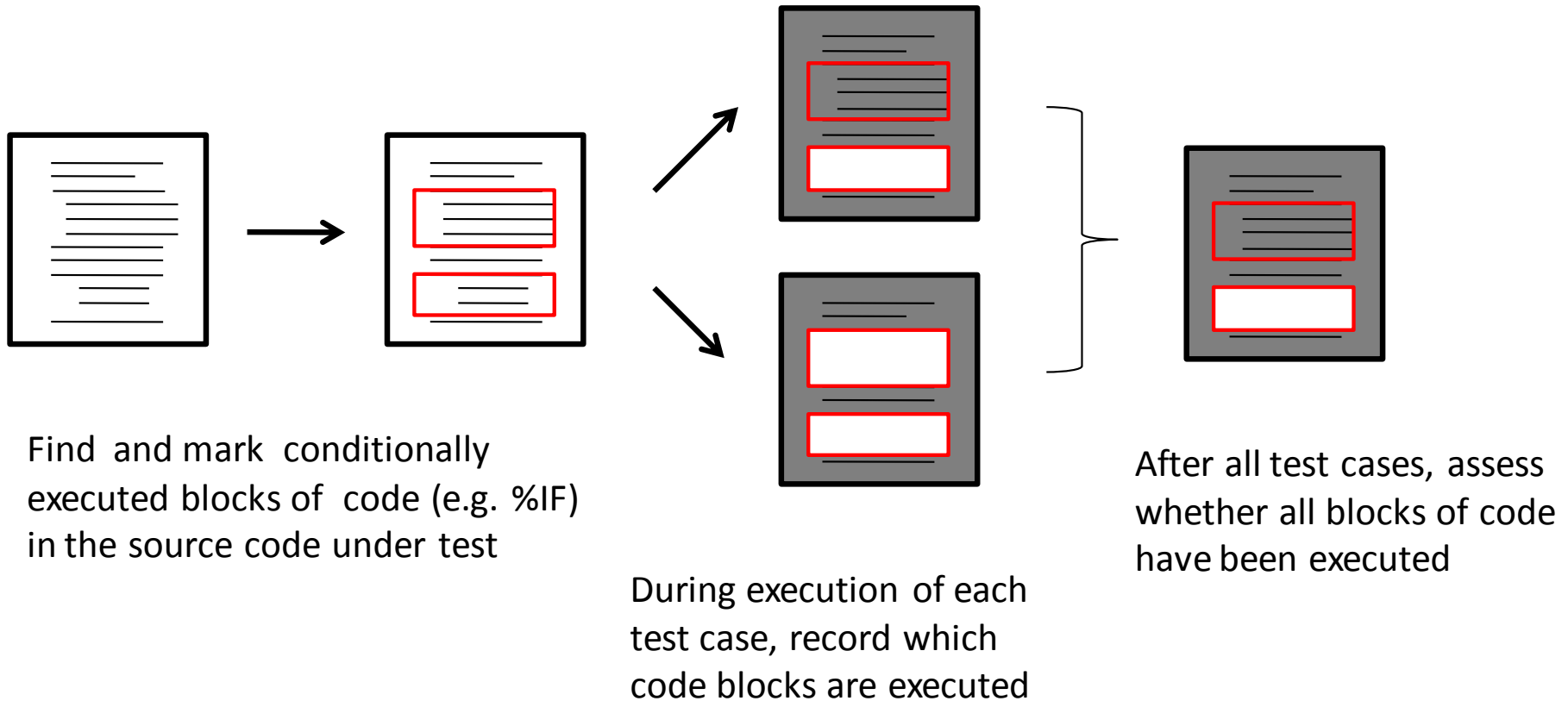
No.	Test Scenario	Program	Last Run	Duration	Result
001	Tests for boxplot.sas	example/saspgm/boxplot_test.sas	21AUG2008:15:13:23	7.0s	☐
002	Tests for generate.sas	example/saspgm/generate_test.sas	21AUG2008:13:58:49	1.8s	☒
003	Tests for getvars.sas	example/saspgm/getvars_test.sas	21AUG2008:13:58:51	1.0s	☒
004	Tests for nobs.sas	example/saspgm/nobs_test.sas	08SEP2008:16:44:58	2.0s	☒
005	Tests for regression.sas	example/saspgm/regression_test.sas	21AUG2008:14:44:29	1.7s	☐

Generated on Monday, 8 September 2008, 16:45:01 by **SASUnit** Version 0.904 (38)

Test Coverage

- Assessment of statement coverage with SASUnit
 - Experimental feature of SASUnit, work in progress
 - Enables determination of source code blocks within a SAS macro that are executed during unit tests
 - Tool for assessment how good the unit tests cover the source code of a SAS macro
- > Discover which parts of a unit under test are never executed during tests and thus are not tested

Implementation of statement coverage assessment



Simple Example

Assessment of Statement Coverage (here 75%):

```
/*Coverage Test Macro*/
```

```
%MACRO ccTestMacro1(binaryInput);
```

```
  %LOCAL printTxt;
```

```
  %LET printTxt = A value not equal to 1 was given.;
```

```
  %IF &binaryInput EQ 1 %THEN %DO;
```

```
    %LET printTxt = A value equal to 1 was given.;
```

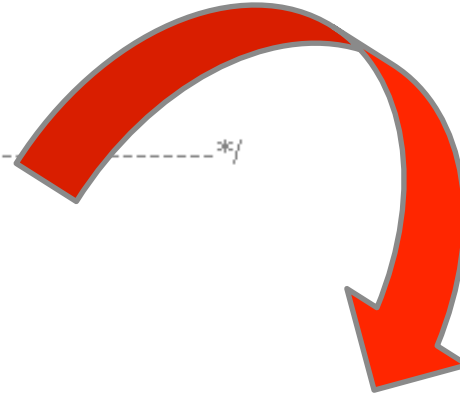
```
  %END;
```

```
  %PUT &printTxt;
```

```
%MEND ccTestMacro1;
```

Example: Assess whether all input parameter checks are tested

```
/*-- check input data set -----*/  
%let dsid=%sysfunc(open(&data));  
%if &dsid=0 %then %do;  
%put ERROR: boxplot: Data set &data does not exist;  
%return;  
%end;  
/*-- check whether x variable has been specified -----*/  
%if "&x"="" %then %do;  
%put ERROR: boxplot: X variable not specified;  
%let dsid=%sysfunc(close(&dsid));  
%return;  
%end;  
/*-- check for existence of x variable -----*/  
%if %sysfunc(varnum(&dsid,&x))=0 %t  
%put ERROR: boxplot: Variable &x do  
%let dsid=%sysfunc(close(&dsid));  
%return;  
%end;  
/*-- check whether x variable is numeric -----*/  
%if %sysfunc(vartype(&dsid,%sysfunc(varnum(&dsid,&x)))) NE N %then %do;  
%put ERROR: boxplot: Variable &x in data set &data must be numeric;  
%let dsid=%sysfunc(close(&dsid));  
%return;  
%end;
```



Program Library [example/saspgm](#)

Unit under Test	Test Scenario	Asserts	Test Coverage [%]	Result
boxplot.sas	002	9	63.50	☐
getvars.sas	001	4	72.22	☒

Conclusion

- **Using SASUnit ..**

- you may require minimal more effort during the initial development phase of SAS macros
- you will save a lot of time during implementing maintenance/change releases
- documentation of your tests is generated automatically
- in near future statement coverage assessment might help you to determine how good unit tests cover the source code of a SAS macro

Thank you for your attention

Dr. Patrick René Warnat

**HMS Analytical Software GmbH
Rohrbacher Str. 26
69115 Heidelberg
Germany**

www.analytical-software.de