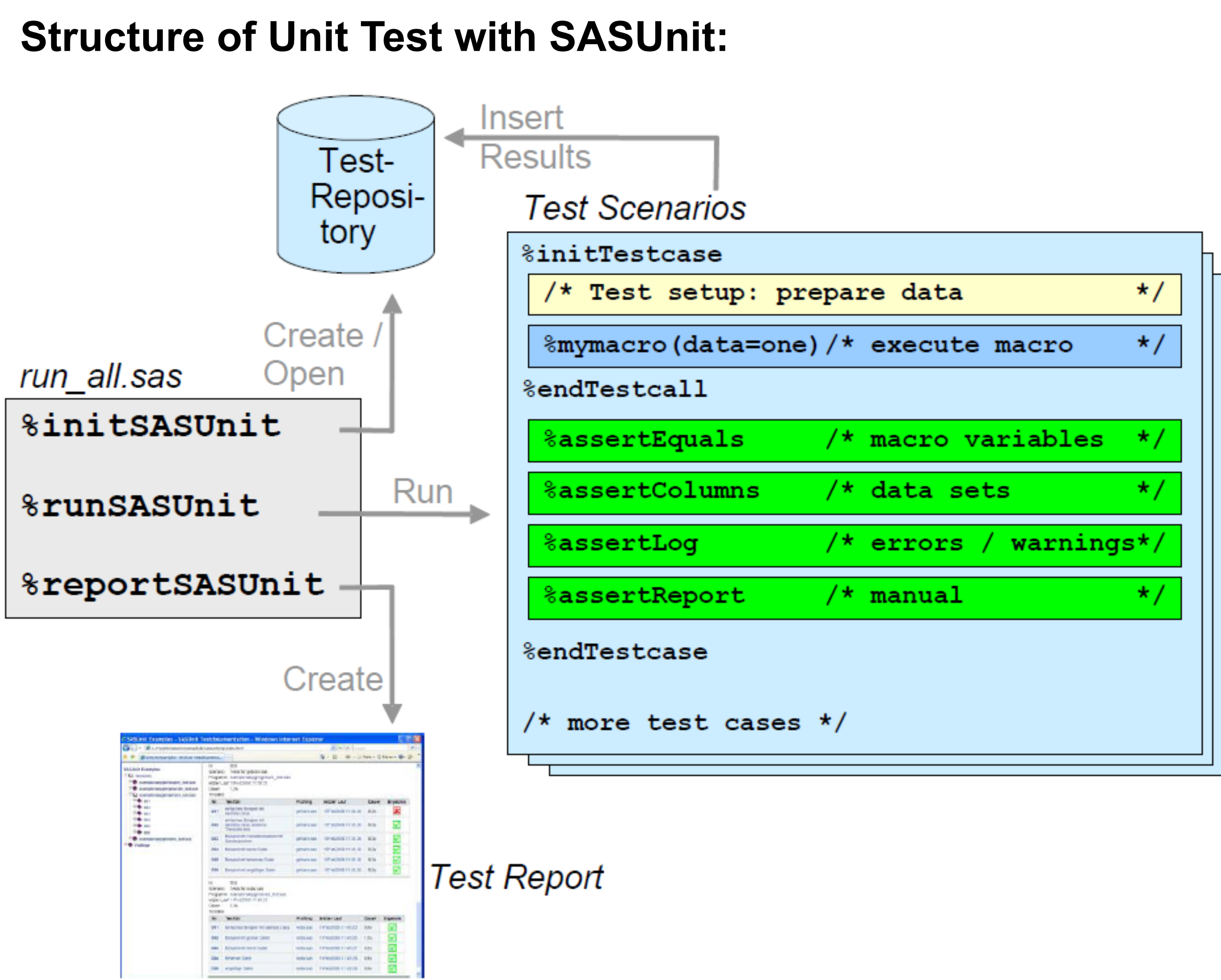


SASUnit

- SASUnit is a unit testing framework for the development, execution and documentation of tests for SAS programs (SAS 8.2, 9.1, 9.2, 9.3 and 9.4 on Windows and Linux).
- With SASUnit, tests are implemented as executable source code. These tests can be repeated with very little effort, therefore negative side effects of software changes are spotted easily.
- SASUnit consists of SAS macros and a few shell commands and is provided for free from HMS Analytical Software GmbH (Download on sourceforge: <http://sourceforge.net/projects/sasunit>).

Generell Overview



- Features of SASUnit:
- ✓ Write tests as SAS source code.
 - ✓ Use “Assertion” macros to check for expected test results.
 - ✓ Run tests in batch mode.
 - ✓ Measure macro run times.
 - ✓ Include messages in the test results documentation.

Test-Reporting

Example or report “Units under Test” :

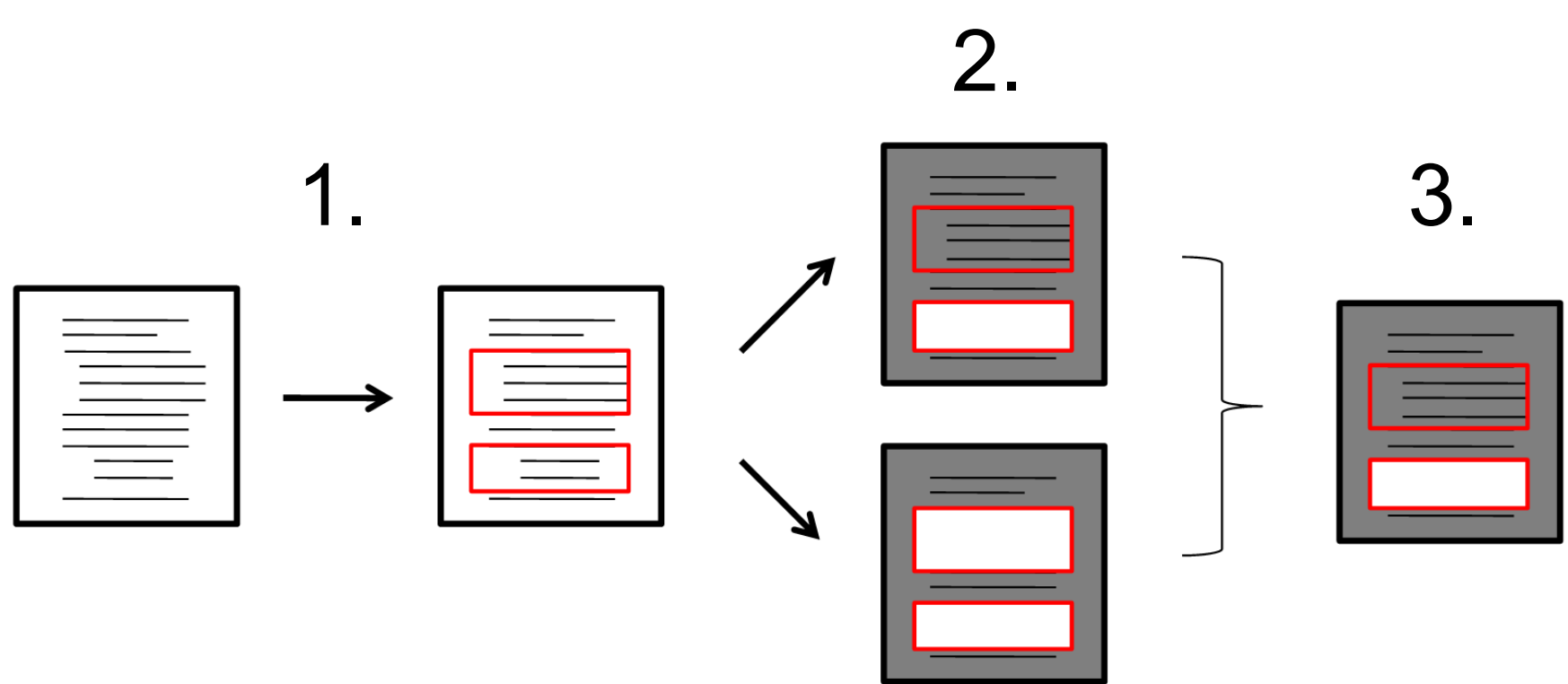
Unit under Test	Test Scenario	Test Cases	Assertions	Test Coverage [%]	Result
boxplot.sas	001	22	45	97	☐
generate.sas	002	6	9	100	☒
getvars.sas	003	6	12	100	☒
nobs.sas	004	6	12	100	☒
regression.sas	005	1	6	100	☐

- Automatically generated documentation of test results:
- ✓ Test result documentation is consistently structured and combined with the documentation of the SAS macros.
 - ✓ Summary Statistics:
 - ✓ number of Test Cases and Assertions
 - ✓ Test Coverage
 - ✓ Intuitively colored representation of test results

Recent Developments

- New Features in Versions 1.2 - 1.3:
- 1.2:
 - Test procedures for SASUnit itself were improved.
 - The new self-test environment executes a detailed test plan for SASUnit.
 - New assertion for performance testing (assertperformance.sas).
 - Test cases support redirection of log files.
 - 1.2.1:
 - Assessment of test coverage (SAS 9.3 or higher)
 - 1.3:
 - Write results of assertions to saslog.
 - New assertions asserttableexists, assertrowexpression, assertrecordexists, assertrecordcount, assertforeignkey.
 - Integrate SASUnit with continuous integration tools like Jenkins.

Assessment of test coverage



- Process of Test Coverage assessment:
- During execution of each test, it is recorded with SAS Option MCOVERAGE which code blocks are executed.
 - For each macro under test, coverage results of all tests are combined.
 - Finally, it is easy to assess which blocks of code have never been executed during tests.

```
000001 /*Code Coverage Test Macro*/
000002 %MACRO ccTestMacro1(binaryInput);
000003 %LOCAL printTxt;
000004 %IF &binaryInput EQ 1 %THEN %DO;
000005     %LET printTxt = value equal to 1;
000006 %END;
000007 %ELSE %DO;
000008     %LET printTxt = value not equal to 1;
000009 %END;
000010 %PUT &printTxt;
000011 %MEND ccTestMacro1;
```

Example: assessment of test coverage of a macro under test
Line 5 is detected as never executed during tests.