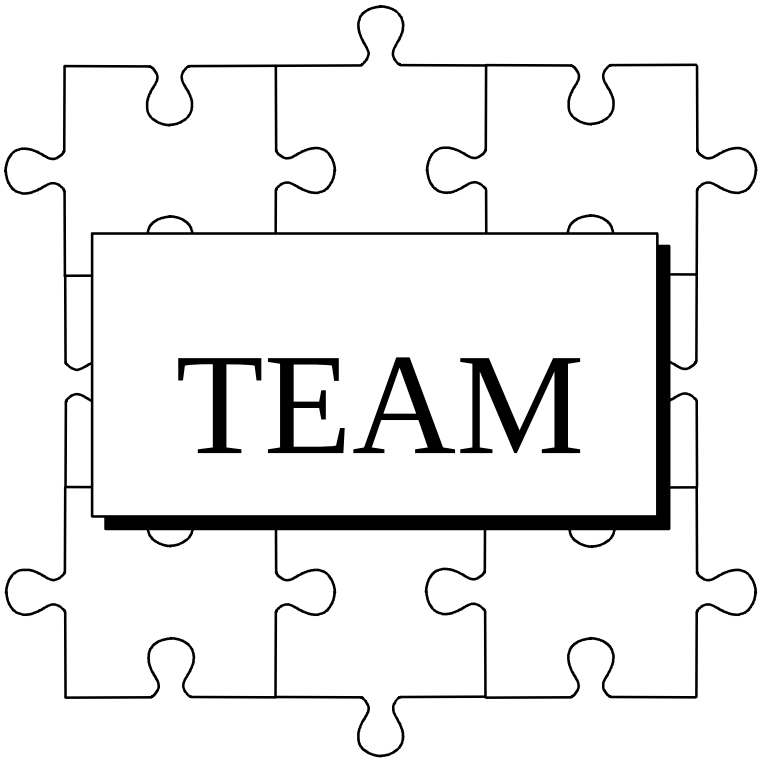

Test Execution and Test Management for Numerical Control Software

Best Practice Action IST-1999-20333

Deliverable D-3.1

Author(s):	Joachim Mayer, Andreas Grosse, Thomas Bürger
Type:	Deliverable
Activity:	WP 4.1; Set up of constructive QA activities
Date:	10.09.01
Status:	Released
Name of document:	D3_1_TEAM20333_V1.doc
Availability:	IST

Table of Contents

1 Abstract 3

2 Objectives of WP-4 3

2.1 Objectives of WP-4.1 3

2.2 Objectives of WP-4.2, WP-4.3 and WP-4.4 3

3 Set up of constructive QA activities (WP 4) 4

3.1 Specification of constructive QA activities (WP 4.1) 4

3.1.1 Tailored V-Model Requirements 4

3.1.2 Constructive QA activities to be set up at ISG 5

3.1.2.1 Classification of software development process 5

3.1.2.2 Methods and tools to support the software development process 7

4 Summary 10

5 Glossary 11

6 References 11

1 Abstract

The submitted paper contains the results of the Workpackage 4.1 (WP-4.1) of the IST project IST-1999-20333. The work performed in WP-4.1 is described in the present “Specification of constructive QA activities” as planed in the Description of work (DOW) /1/ (WP-4.1).

2 Objectives of WP-4

WP-4 contains all activities related to the set up of constructive QA activities. Generally, WP-4 is subdivided into the four subtasks

- WP-4.1 Specification of constructive QA activities
- WP-4.2 Preparation of standard forms for test execution
- WP-4.3 Realisation of spec. constructive QA activities
- WP-4.4 Validation of constructive QA activities

2.1 Objectives of WP-4.1

The objective of WP-4.1 in particular is to specify the constructive QA activities based on the results of WP2 and WP3. During the specification the way how to integrate the analytical QA activities to the software development process was defined. All further steps to put this constructive QA activities into practice will be described within the subsequent work packages.

2.2 Objectives of WP-4.2, WP-4.3 and WP-4.4

... to be continued in deliverable D-3.2

3 Set up of constructive QA activities (WP 4)

ISG will focus within the TEAM-project on the introduction of analytical and constructive QA activities to improve the NC software development process and test management. Both, the analytical QA activities as well as the constructive QA activities, shall be introduced according to the regulations of the QA submodel of the V-Model /2/. This regulations were adjusted to ISG's needs within the so-called Tailoring phase of the TEAM-project. More over this an investigation of the current state of the test process at ISG was executed within the Tailoring phase. The results of the Tailoring phase are the determination of the QA activities particularly required for ISG's needs.

Within work package WP-3 the main focus was on the introduction of suitable analytical QA activities (such as static or dynamic testing, code review, etc.), which are necessary to detect and localise weaknesses and defects within the software components /4/. In work package WP-4 the establishment of constructive QA activities, which can be considered to be preventive QA activities to improve NC test management, shall be achieved. This constructive QA activities comprise the specification, planning and reporting of the respective analytical QA activity.

3.1 Specification of constructive QA activities (WP 4.1)

3.1.1 Tailored V-Model Requirements

Quality can not be implemented afterwards into a product. That's why it is absolutely necessary, to promote the generation of quality by constructive QA activities. The stated objective is to avoid preventively quality relevant risks, to counteract quality faults and to prepare the product to make it testable.

Constructive QA activities comprise for example:

- The classification of the software development process by use of a SW-development standard.
- The support of the software development process by methods and tools.

3.1.2 Constructive QA activities to be set up at ISG

With respect to the Tailored V-Model requirements mentioned above ISG intends to improve the software development process and the quality of the CNC software.

3.1.2.1 Classification of software development process

As described in /3/, ISG's NC-Kernel software (baseline project) is a continuously growing project. The development process can be described as a further incremental development of an existing Version (Fig. 1).

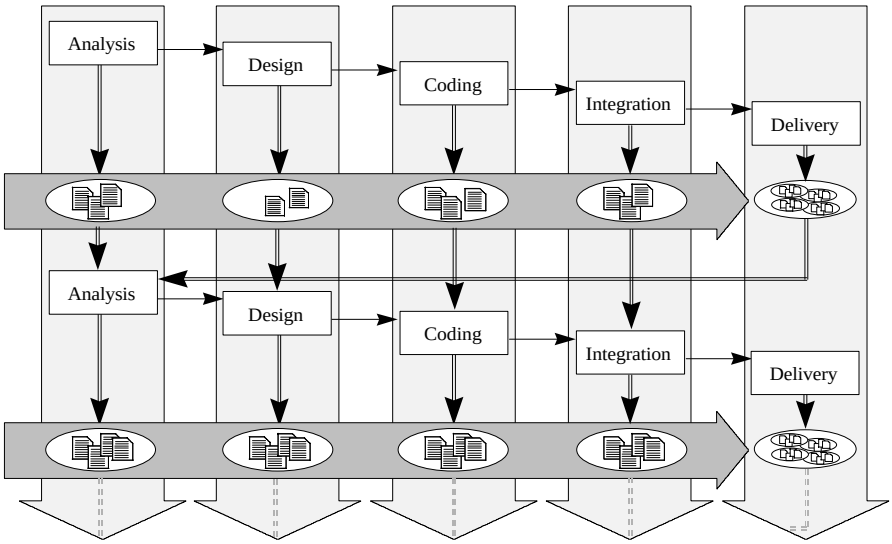

Fig. 1: Further incremental development of NC software

In this case the following three types of software change activities (Change Process) at ISG can be named:

1) Functional extensions

This change process (CP) comprises all stages of software development starting from analysis to integration. Generally this process is completely executed for comprehensive functional extensions or new developments. In close cooperation to the customer during the specification phase (Analysis and Design in Fig. 1) the requirements are defined and the realisation is described. After that the software development (Coding) starts. After conclusion of this phase the newly created software is integrated or merged into the mainstream of the related NC-software version (Integration). Then this new version can be delivered to the customer

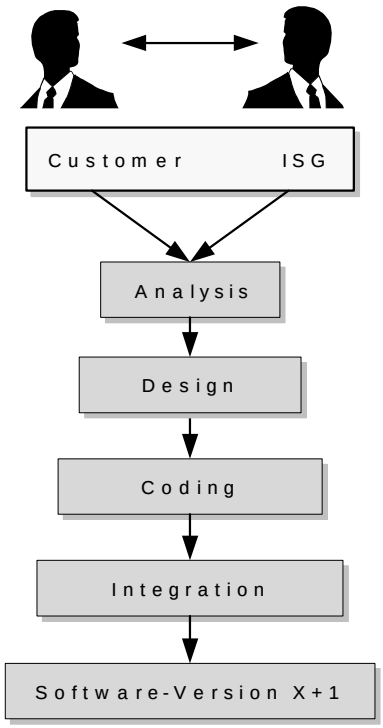

Fig. 2: Change process of a functional extension

2) Bug removal or change request

This change process is initiated by error messages or change requests, which are collected in a data base /3/. The bug is removed or a requested change is realised as far as possible. After that the software change is integrated in the related NC-software versions. (Fig. 3).

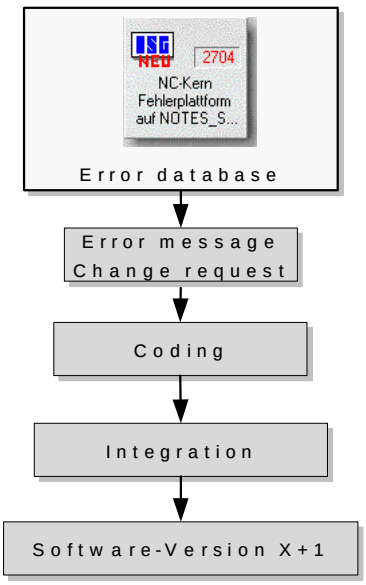

Fig. 3: Change process of a bug removal or change request

3) Functional improvement

The functional improvement is an exception, because it is not based on a specification. It means often the improvement and optimisation of already existing software regarding algorithms and source design and may not have any critical effects. But this software changes also are an evolution of a version and must be considered as a separate change process (Fig. 4)

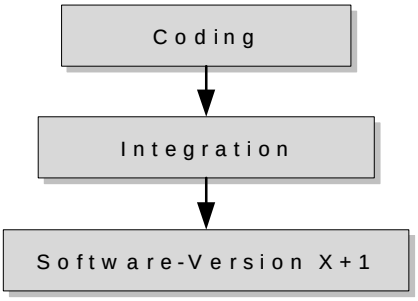

Fig. 4: Change process of a functional improvement

3.1.2.2 Methods and tools to support the software development process

3.1.2.2.1 Analysing methods

A.) Review

The review methods belong to the manual methods and can be distinguished between specification review and code review (Fig. 5). The specification review is used in the software development process after conclusion of the analysis and design phase and is done by other persons out of the project team (e.g. team manager, other developers). Necessary changes are considered and are integrated. Only after the release of the specification the coding phase may start.

The code review supports the phases of coding and integration and shall ensure the correct realisation and implementation of the specified software change.

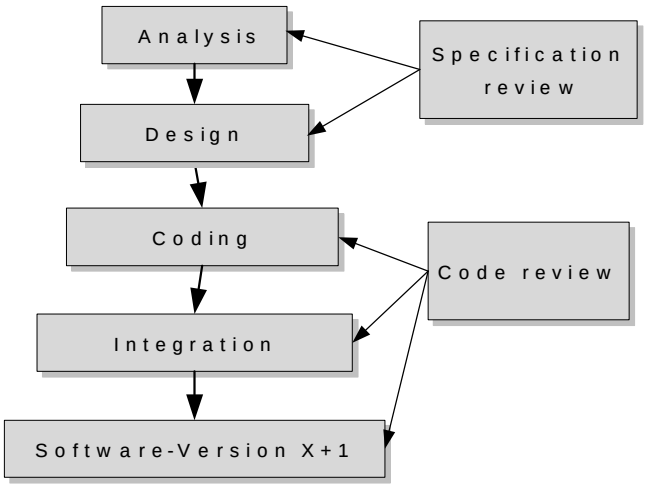

Fig. 5: Review methods during the software development process

B.) Commercial tool

To detect errors in a very early stage of software development process, the commercial tool INSURE++ from Parasoft for the additional code analysing must be used. Integrated in the ISG standard development environment this tool supports static and dynamic code analysing. So it can also be used during runtime of the NC-software. The tool is usable during coding and integration (Fig. 6). The analysing results are stored by the tool itself and detected errors must be judged by the developer /4/. Errors which are caused by the latest software change must be eliminated at once, because they belong to the current development. All other detected errors must be reported to the error data base described in /3/ because they also can concern other software versions

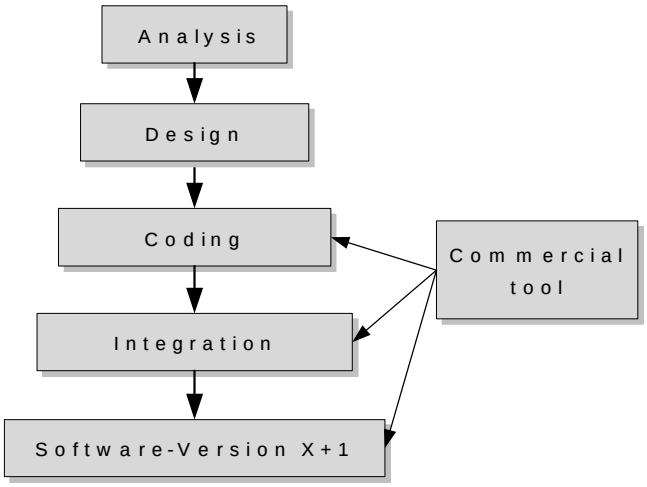

Fig. 6: Stages of the software development process supported by the commercial tool

3.1.2.2.2 Functional tests

A.) Regression test

The automatic test environment (regression test) makes it possible to execute a large number of test cases in a reduced time effort to detect and localise weaknesses and defects within the complete software components. So bugs, e.g. based on new developments, side effects, cross dependencies and also other bug fixes can be detected in an automatic way. The regression test process represents an important method of detailed inspection activities. It will be used for the verification of the existing basic functionality of the NC system after a change within the system and helps to detect and eliminate bugs and malfunctions before a customer delivery.

While running the regression test also the commercial tool INSURE++ for code coverage analysis can be used to examine the effectiveness of the automatic functional test.

The regression test is executed after the integration of a new development and before the delivery of a new version (Fig. 7) /4/.

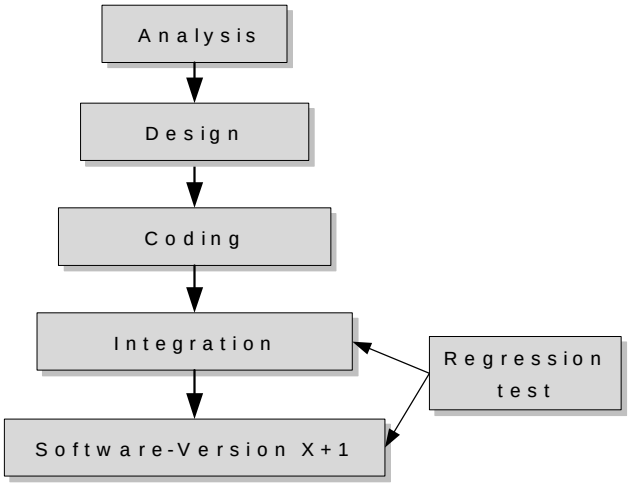

Fig. 7: Use of the regression test during the software development process.

B.) Special test tools

As described in /3/ ISG has created some special test tools. These test tools are mainly necessary e.g. for the test of interfaces to external control components (PLC-Simulation, HMI-Simulation) or the simulation of hardware elements (hand wheels) or axis movements. They are used in addition to the regression test (Fig. 8).

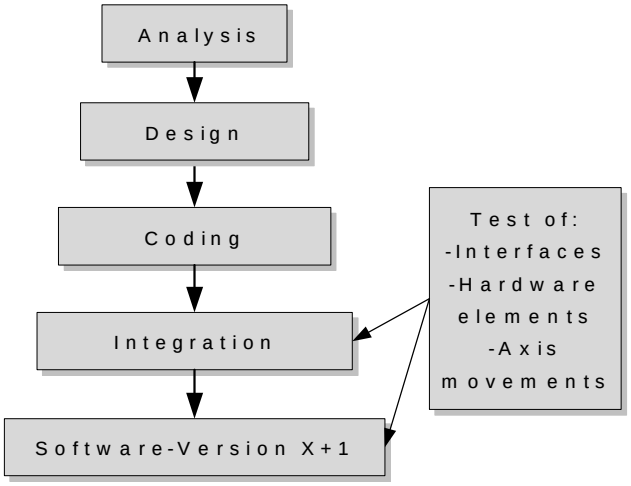

Fig. 8: Further functional tests

C.) Machine tests

After the conclusion of a complex change process a validation of the changed software is done with simulation systems or real machine tools. The tests are done after delivery by the customer himself or if possible by ISG, if an appropriate machine system is available. For the execution of the validation the complete NC-software “X+1”must be integrated and installed (Fig. 9).

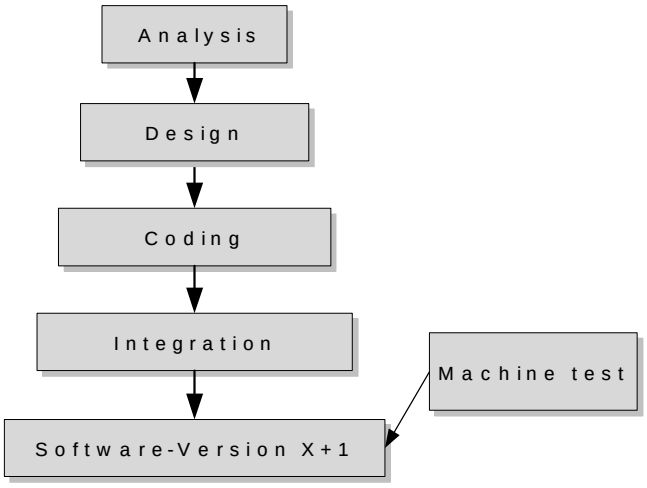

Fig. 9: Machine test of a new software version

4 Summary

At ISG the software development process can be described as a further incremental development of an existing version. This change activities occur not only due to functional extensions but also to functional improvements and bug removal activities. This three change processes can be executed parallel at one software version.

ISG introduced a commercial tool (INSURE++) to improve the software development process and developed a regression test to detect the effects of the software changes on the whole NC-kernel software. Within the scope of constructive QA activities ISG defined, that for a all change processes the new QA methods must be used in addition to the already existing. The commercial tool can be used during coding, integration and testing. The regression test checks the functional completeness of the NC-kernel software after conclusion of the change processes. The following figure (Fig. 7) gives an overview about the constructive QA activities at ISG.

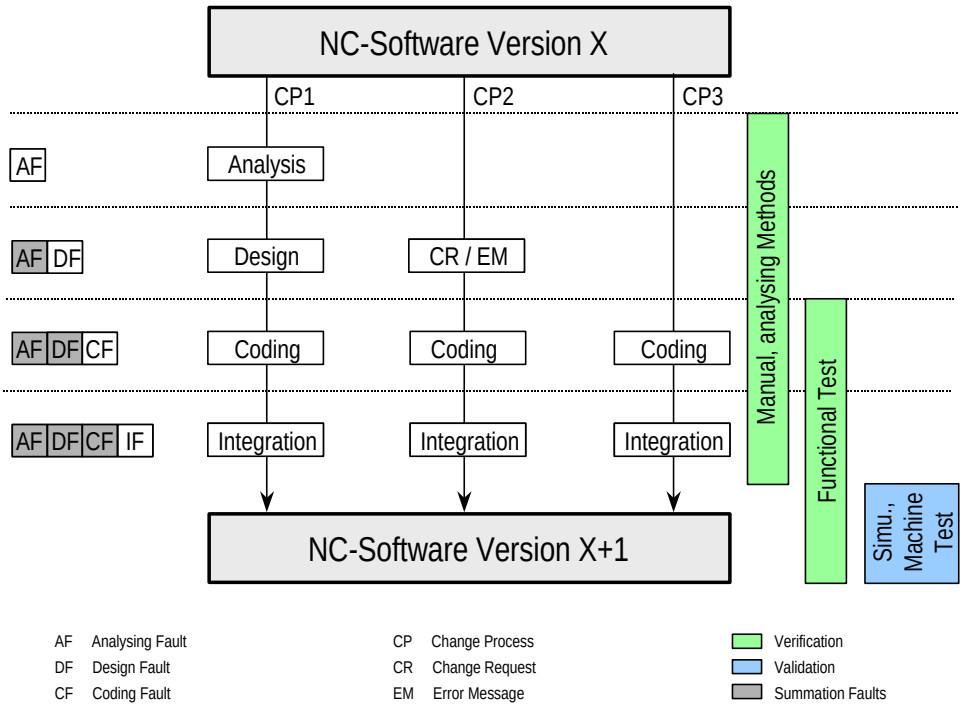

Fig. 7: Constructive QA activities at ISG

5 Glossary

ISG	Industrielle Steuerungstechnik GmbH
TEAM	Test Execution and Test Management for Numerical Control Software
NC	Numerical Control
QA	Quality Assurance
WP	Work Package
DOW	Description of Work
SW	Software
IST	Information Society Technology
HLI	High Level Interface
HMI	Human Machine Interface

6 References

/1/ ISG: TEAM Project - Description of Work Version 5, Annex I of IST Contract No. IST-1999-20333, Stuttgart, 2000.

/2/ N.N.: Entwicklungsstandard für IT-Systeme des Bundes - Vorgehensmodell, 1997.

/3/ ISG: TEAM Project - Deliverable D1 - Tailoring

/4/ ISG: TEAM Project - Deliverable D2.1/D2.2 – Set up of analytical QA activities