

Integrated Methods for Systems and Software Engineering

Martin Lang
EADS Deutschland GmbH
Airborne Systems
System&Defense Electronics
89077 Ulm, Germany
+49 (0) 731.39 2-33 52

Martin.Lang@sysde.eads.net

Peter Hoellring
EADS Deutschland GmbH
Airborne Systems
System&Defense Electronics
89077 Ulm, Germany
+49 (0) 731.39 2-36 84

Peter.Hoellring@sysde.eads.net

ABSTRACT

Productivity increase and obsolescence robustness have become key issues for military software development. Programs like RASSP (USA) and ESPADON (Europe) have elaborated the requirements for future development processes and tools. EADS Airborne Systems has performed a research project with the objective to develop a system and software development process that satisfies these new requirements. Using appropriate tools supports the development process. The efficiency and suitability of the defined process was tested in a pilot project, where metrics for the improvement of productivity were gained.

1. Introduction

As an ongoing process within a company, processes have to improve for various reasons.

Productivity increase is one of the most important reasons to stay competitive. However, looking through the live-cycle cost of a military product, a major expense is the product adaptation throughout the production phase due to parts, which are obsolete. Due to adapted hardware items, changes are also necessary to software items. A process, which allows mitigating the obsolescence issue, will result in an overall cost savings during the product live-cycle.

New types of development tools are another reason to adopt the development process. New tools allow to use a different and often more efficient process as they simplify one or the other step or even allow to abolish a step within the process. External programs or studies can also give some impulses and ideas on improvements.

Deficiencies recognized in the current process are also drivers to improve the applied process. One of these has been the transition from the systems to the software department. There was usually a gap in the development flow as none of the generated software could be reused. The new process should reduce this gap to allow a smooth transition between the departments.

Company certification requires adopting and improving the currently applied processes ongoing. Therefore defining a new process supports this goal.

2. Basic Process Requirements

This chapter summarizes the major requirements for a new development process.

The gap between the systems and the software department shall be avoided. A smooth transition within the workflow of the departments is required.

The process shall contain steps to identify and therefore reuse similar or identical algorithms and software components.

The process shall support an iterative, incremental development approach, which is compatible with the overall company process.

The testability shall be incorporated into the early stages of development. The tests shall be repeated after each stage of the process to validate the correctness of the implemented software.

The documentation effort shall be reduced. The software shall inherently include the documentation. This will also improve the software maintainability.

3. Selection of a Development Tool

The selection of the appropriate development tool has been split into three steps.

As a starting point, the tool requirements have been defined. Process specific requirements as well as company standards were taken into account.

The next step included a research of available tools on the market. Each of the tools has been assessed and a review procedure guaranteed the quality of the assessment.

Finally, the tool meeting most of the requirements has been selected.

3.1 Definition of Assessment Criteria

Derived from the process requirements stated in chapter 2, the following requirements are the most important to be met by the selected tool:

There shall be only one tool supporting the requirements, design and implementation phase. This includes a smooth transition from the requirements to the design phase.

The description language of the tool shall be data flow oriented, as this is the basic representation of signal processing software.

The representation of the functionality shall be separated from the implementation. It should be possible to change functionality without modifying the underlying implementation.

The effort of porting the application to a new hardware platform shall be minimized. The tools should support a layered software architecture, which hides all target platform specific implementation details in layers below the application.

The tool shall also be able include commercial available libraries. This includes low-level vector libraries as well as high-level component libraries.

As a consequence, all application programmer interfaces (APIs) shall be open.

The tool shall be available on common platforms. It shall also be able to handle multiple users and subprojects. The user interface shall meet today's standards.

The tool manufacturer shall provide an excellent support. There shall be a clear roadmap on how the product evolves.

3.2 Selection Process

With the defined tool requirements, a market research has been performed. Evaluation licenses were used to collect first experiences. Open issues have been tried to resolve together with the manufacturer or the distributor. A report was generated and reviewed for each tool.

As an outcome, GEDAE was selected as the tool which meets almost all the requirements defined.

4. Development Process

The GEDAE development process is subdivided into three major phases:

- Mode Requirement Analysis
- Architectural Mode Design
- Implementation and Integration

An overview of the process is given in Figure 1.

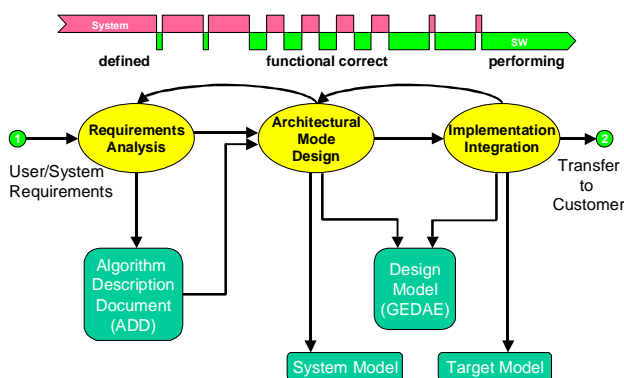


Figure 1. Development process overview

The 'Requirements Analysis' phase starts with the availability of the user and system requirements. The major artefact of this phase is the Algorithm Design Description (ADD), which is an implementation free representation of the algorithm

requirements. Systems engineers mainly perform this phase with some support by software engineers.

Within the next phase 'Architectural Mode Design' a functional correct representation of the algorithms in GEDAE is generated. The major artefacts of this phase are the Design Model, a graphical representation of the algorithms within GEDAE and the Host Code, a C-code equivalent of the graphical representation generated by GEDAE. The 'Architectural Mode Design' activities are supported by automated tests. Systems and software engineers work together as integrated teams throughout this phase to guarantee the correct functionality and consider all necessary implementation details.

Implementation and Integration is the last phase within the GEDAE development process before the executable software is transferred to the system integration. Within this phase, the functionality of the Design Model is ported to the required hardware and the Design Model is optimised to meet the timing requirements. The major artefact is the Target Code for system integration. Mainly software engineers perform this phase with some assistance of systems engineers to resolve issues coming up during this phase.

4.1 Mode Requirement Analysis Process

User and system requirements are the input to the mode requirements analysis phase. Ideally the requirements meet the following properties:

- Clear language
- Unambiguous
- In agreement with customer's needs
- Well structured
- Atomised

The mode requirement analysis phase is structured into (see Figure 2):

- Coarse Functional Analysis
- Functional Decomposition
- Functional Refinement
- Create Test Cases
- Synthesis

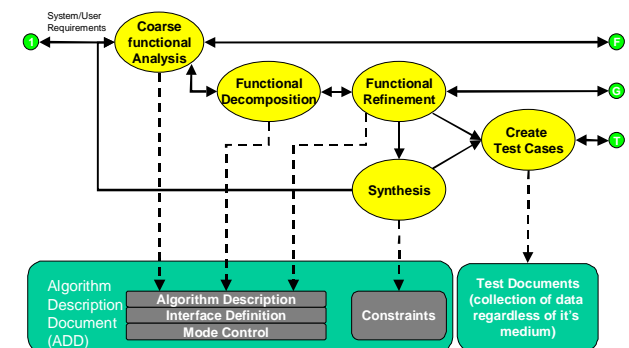


Figure 2. Mode Requirement Analysis

During the coarse functional analysis, the system and user requirements are analyzed with the goal to determine the

functional blocks and their interactions, which meet the requirements. The output of this activity defines the signal processing algorithms and shall give a mode overview.

The entire mode functionality shall be split up into a hierarchical structure during the functional decomposition.

During the functional refinement, the atomized functions and all hierarchies of functions shall be described with text. It shall be described what is done and how something is done.

The results of these phases are documented in an algorithm design description. The internal and external interfaces as well as the processing allocated to an algorithm are precisely described within the document.

The consistency between the lower and higher-level functions is checked during the synthesis phase. Constraints are determined and the limitations are feed back to the earlier stages in the phase.

Test cases for algorithms, functional entities and modes are defined to allow a conformity check of the latter model.

4.2 Architectural Mode Design Process

The architectural mode design process is structured into (see Figure 3):

- Develop Top Level Flow Graph
- Develop Detailed Flow Graphs
- Create a set of Test Data
- Develop Test Benches

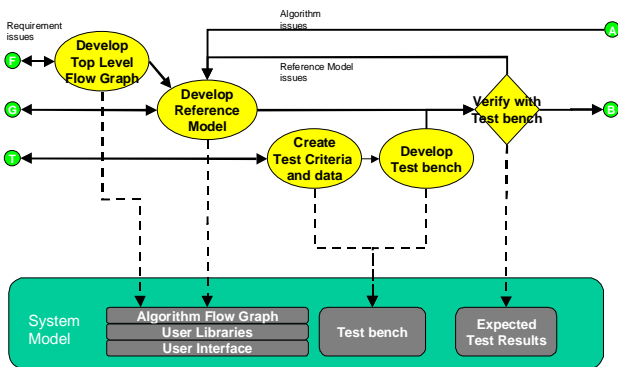


Figure 3. Architectural Mode Design

The top-level flow graphs are developed by the systems engineers within GEDAE according to the course coarse functional analysis in the requirements phase.

These top-level flow graphs are detailed during the reference model generation to a functional level. The functions itself can be realized as GEDAE models or a thin simulation layer supported by GEDAE can be included to interface to a reference implementation. This is an executable functional model respective specification.

Test criteria's for the test cases defined during the requirements phase are defined in this phase. Based upon the test cases and

criteria, test benches for algorithms, functional entities and modes, are generated to verify the functionality.

Automated tests are set up in this stage and are reused in the entire software life cycle.

4.3 Implementation Process

The Implementation phase is structured into the following sub-processes (see Figure 4):

- Develop detailed algorithm flow graphs
- Partition and locate functions into memory
- Optimize execution efficiency

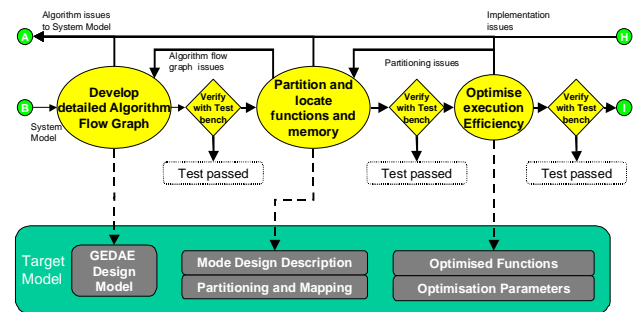


Figure 4. Implementation Process

The reference model of the previous stage is transferred to a complete GEDAE model by replacing simulation models by flow graphs and primitives during the 'Develop detailed algorithm flow graph' stage.

'Partition and locate functions and memory' is the process of dividing and unifying functions with the aim to have reasonable function groups which can be located on a processor and his memory. The outcome of this stage is documented in a mode design description. This document also includes a list of functions to be optimized to meet real time requirements.

'Optimize execution efficiency' is the process of full-filling real-time requirements

4.4 Integration Process

After the implementation phase, the GEDAE design model is passed to the integration stage (see Figure 5).

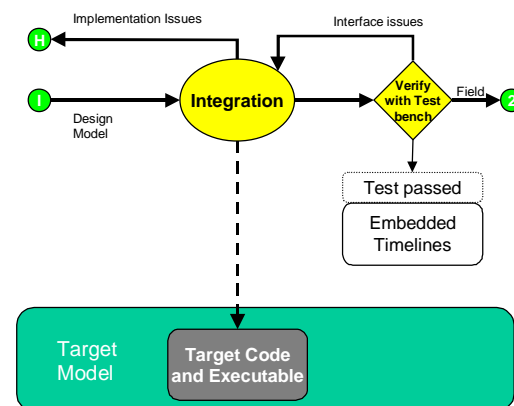


Figure 5. Integration Process

The integration stage is the embodiment of the developed system into its environment. The outcome is the target code and executable. The stage is usually divided into several sub-stages, which is project dependent. The output of the stage is passed to the formal acceptance test and finally to the customer.

5. Fitting Development Process to Overall Company Development Process

The EADS Deutschland GmbH Airborne System life cycle process model VM-GBV is shown in Figure 6. It is based on the V-Model defined by the German Department of Defence with company specific extensions such as the iterative and incremental development approach.

The model includes sub-models for system development (SD), software and hardware (SD-SW and SD-HW), integrated logistics (SD-ILS) as well as sub-models for project and configuration management and quality assurance.

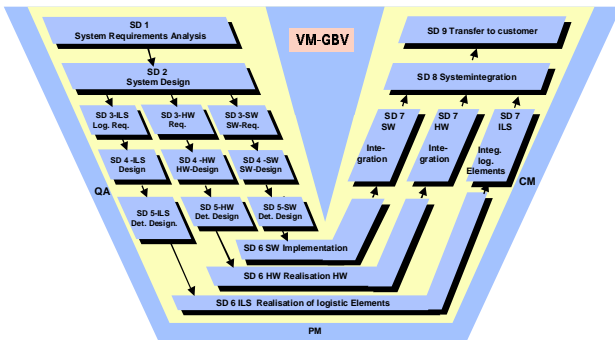


Figure 6. V-Model Overview

The GEDAE development process described in this paper relates to the sub-model software. The sub-model SD-SW describes the activities, which are necessary for the development of SW portions of the overall system. It includes the following steps:

- Analysis of SW requirements (SD-3-SW)
- Preliminary SW design (SD-4-SW)
- Detailed SW design (SD-5-SW)
- SW implementation (SD-6-SW)
- SW integration (SW-7-SW)

As a result, the GEDAE development process contains exact the same stages as the overall company process. The two stages preliminary and detailed SW design are part of the architectural mode design described in chapter 4.2.

An iteration within the VM-GBV is a planned sequence within the steps analysis (SD-3-SW), design (SD-4/5-SW), implementation (SD-6-SW) and integration (SD-7-SW). Not all steps have to be performed. The development plan defines the steps.

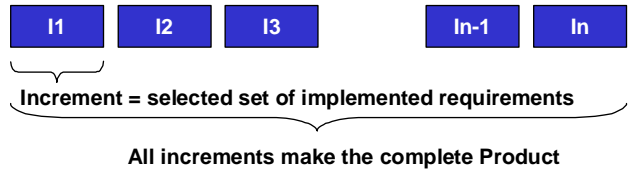


Figure 7. Increments forming the product

The product of one iteration is an increment. An increment is a selected set of requirements to be implemented (see Figure 7). All increments form the complete product to be integrated on system level (SD-8-SW) and deployed (SD-9-SW).

6. Verification of the Development Process

After the definition of the development process, a pilot study has been carried out. The pilot study had the following major objectives:

- Verify that the defined development process meets its requirements.
- Gain productivity figures.
- Collect extensive experiences working with GEDAE
- Gather additional requirements for GEDAE

To meet the pilot study objectives, a radar signal processing software component developed by a different team with well-known metrics has been selected.

Three signal processing software engineers carried out the study. A commercial off the shelf-target platform with eight processors has been used.

Vast experiences have been collected. The requirements of the development process have been met. A significant gain in productivity has been recorded.

A few improvements to the development process have been applied as a result of the pilot study.

7. Summary

EADS Airborne Systems is satisfied with the results of the defined development process. There is a significant gain in productivity. It is expected that this figure will rise with the experience gain with GEDAE and the process.

Within EADS Airborne Systems the process will be applied to all new signal-processing projects