On One Approach to Scientific CAD/CAE Software Developing Process

George Sergia[1], Alexander Demurov[2], George Petrosyan[3], Roman Jobava[4]

**[1] Exact and Natural Sciences department, Tbilisi State University
Tbilisi, 0160, Georgia**

**[2] Exact and Natural Sciences department, Tbilisi State University
Tbilisi, 0160, Georgia**

**[3] EM Consulting and Software Ltd., (EMCoS)
Tbilisi, 0160, Georgia**

**[4] EM Consulting and Software Ltd., (EMCoS)
Exact and Natural Sciences department, Tbilisi State University
Tbilisi, 0160, Georgia**

Abstract — **Development of science-intensive software products is a complex and time-consuming process that requires good management and coordinated teamwork of all members, involved in the development. Each complex software product has a number of special features; as well as each development team has some unique capabilities. In such conditions it is not always possible to apply standard methods of software engineering without adapting to the specific case. In this paper, we present a development process of large-scale CAD/CAE systems for investigation of electromagnetic compatibility issues.**

Keywords: **S**oftware Development, Software Engineering, CAD Systems, CAE Systems, Development Process

1. Introduction

For many years, software engineering has been continuously evolving and providing solutions to improve practices of software development. Many different methods and approaches have been invented to overcome difficulties, arising during software development process. Some of them have been standardized as typical software engineering techniques.

All of these standards and approaches, like waterfall model[1], agile methods[2] and extreme programming[3], have shown their benefits and disadvantages when applied to specific situations. Despite the fact, that each approach offers a fairly clear advice and guidance, the practice shows that quite often these rules cannot be applied without some modifications and adaptations to specific processes.

This happens due to many factors. Among main reasons that can be identified, one of the most important is, for example, the nature of the product being developed. Development process of science-intensive programs, that perform complex calculations and simulations, differs from developing a client/server application, creating website or online store. Also, structure and members of development team have a major impact on the process, because team consists of people with different characters, skills and experience. Success of the development largely depends on how harmoniously and professionally team members cooperate.

These and other factors must be considered when choosing and applying a particular approach to a specific software development process. If necessary, selected approach must be optimized and adapted according to the known recommendations, or by introducing new recommendations, based on existing experience.

2. CAD/CAE system description

As new technologies appear, penetrating all spheres of human activity, need for high quality scientific software grows. Today's scientific software products are sophisticated CAD/CAE systems for simulation, visualization and analysis of real world objects and processes. A lot of work is required to create such systems, together with knowledge and experience of people from different fields and with different skills.

The process, described in this paper, has been established during development of several CAD/CAE systems, used for modeling and studying the problems of electromagnetic compatibility in electrical systems of modern vehicles.

The software products introduced below represent the result of years of collaborative efforts of developers, academic experts in the field of electrodynamics, testers, technical writers and designers of graphical user interface. These systems are being constantly improved, updated and expanded to meet the needs of customers and target market, taking the increasing competition into account.

The first complex system we will shortly describe is EMC Studio [4] - a powerful program package for the sophisticated computer analysis of electromagnetic compatibility (EMC) problems. This system allows user to simulate, conduct physical calculations and analyze the results. To handle the complexity of the numerous required tools and methods all functions and modules were integrated into one, easy to use program interface. This interface combines all tools for model generation and various methods for calculation.

Fig.1 shows a screenshot of the program with an automobile model.

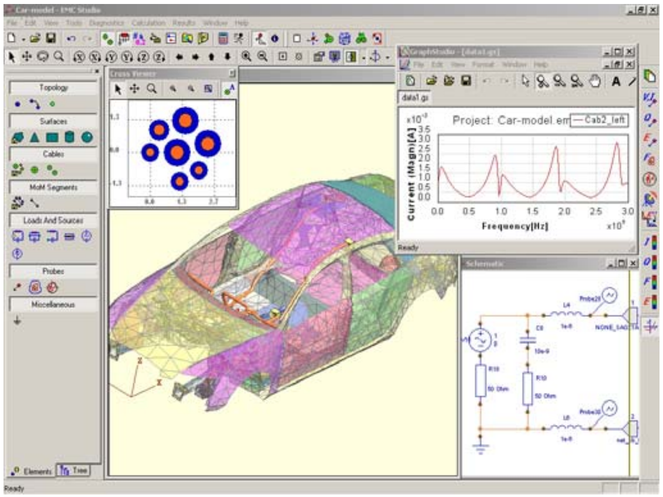
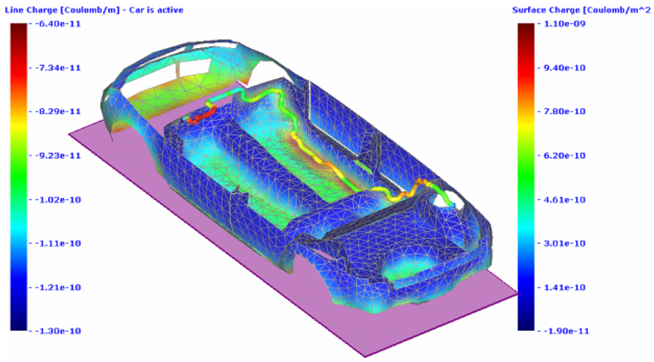


Fig. 1. Screenshot showing model and a result of a hybrid EMC Studio calculation

For the calculation of large systems the following analysis types are necessary and are provided by EMC Studio:

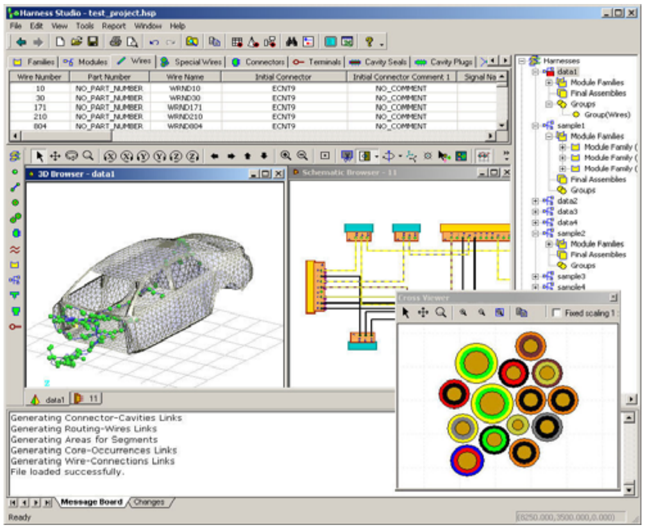
- **EM-Analysis** of linear electromagnetic field and current coupling problems in frequency and time domain with MoM. Electrical field integral equation for harmonic excitation is applied to the calculation model.
- **Circuit Analysis** of linear and non-linear circuits in frequency and time domain with a SPICE 3f5 compatible calculation core.
- **Static Analysis** of linear and non-linear low frequency problems in frequency and time domain with a quasi-static approach.
- **Cross Talk:** Analysis of linear and non-linear terminated complex cable structures in frequency and time domain. A circuit model of an arbitrary transmission line structure is generated. The included 2D field solver calculates the transmission line parameters with respect to the reference conductor. An arbitrary shaped metallic surface structure can be the reference conductor.
- **Radiation Hybrid:** Emission analysis of linear and non-linear complex cable-antenna coupling problems in frequency and time domain. A circuit model of an arbitrary transmission line structure is automatically generated. Common mode currents on transmission lines are calculated and converted into impressed current sources on MoM segments. MoM calculates radiation and coupling to antenna.
- **Susceptibility Hybrid:** Immunity analysis of linear and non-linear complex field-cable coupling problems in frequency and time domain. Field distribution along the cable structure is calculated. The transmission line circuit is supplemented with sources reflecting the incident field. Analysis model is calculated with a SPICE 3f5 compatible solver.

Calculation results can be viewed and analyzed in EMC Studio Post-Processing module. It is possible to view results as 3D and 2D charts:



Another CAD/CAE system for analysis of EMC problems is Harness Studio [5]. This program allows user to create, analyze and maintain a database of cable harness of vehicles.

Harness Studio supports four different visualizations of cable harness database – Table View, 3D View, Schematic View and Cross Section View. Extended modules for harness development and analysis, like an Expert System Option or Electrical and Thermal Analysis Option, enable fast design, rule checking or the calculation of physical properties of the harness.



Wire Number	Part Number	Wire Name	Initial Connector	Initial Connector Comment 1	Signal Na
10	NO_PART_NUMBER	WRND10	ECNT9	NO_COMMENT	
30	NO_PART_NUMBER	WRND30	ECNT9	NO_COMMENT	
171	NO_PART_NUMBER	WRND171	ECNT9	NO_COMMENT	
210	NO_PART_NUMBER	WRND210	ECNT9	NO_COMMENT	
804	NO_PART_NUMBER	WRND804	ECNT9	NO_COMMENT	

```
Generating Connector-Cavities Links
Generating Routing-Wires Links
Generating Areas for Segments
Generating Core-Occurrences Links
Generating Wire-Connections Links
File loaded successfully.
```

Fig. 3. Screenshot showing Harness Engine structure

- **Table view** contains detailed information about different objects in a current database. Data in table can be sorted, searched and arranged in multiple ways.

- **3D Browser** is a powerful viewer. It supports visualization of any objects of a harness database as well as the topology. Comfortable functions for zooming, rotating, and moving make evaluation of any harness database object or complete harnesses very easy. 3D Browser has auto-rotate functionality, which allows rotation of a model in arbitrary direction with user-defined speed. Also 3D Browser has Auto Locate function. This function brings selected object to the center of the 3D Browser area and zooms the view to provide the best fit for the model.

- **Schematic Viewer** can schematically visualize connectors or splices and their connections with other connectors or splices in database. It provides user with convenient tools for scheme rearrangement and manipulation.

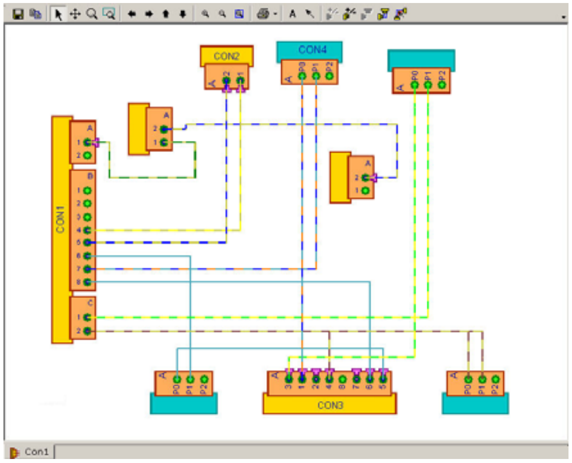
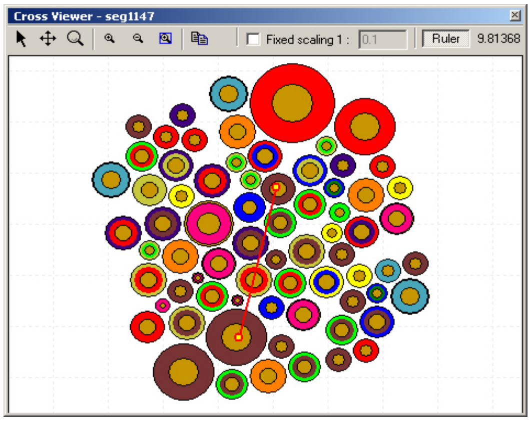


Fig. 4. Harness Engine schematic viewer

- Cross Section Viewer shows the cross-section of any segment interactively. User can click on any cable and the appropriate selection in the cable table will be shown. Colors are shown according to a customizable color list, which gives the user more realistic view of cable bundles. Cable objects are linked to the database. It provides convenient way for distance measurement between arbitrary points in cross section area.



☐ Fixed scaling 1 : 0.1

Ruler 9.81368

Fig. 5. Harness Engine schematic viewer

Expert System module extends Harness Studio to powerful rule checker with full functionality, developed on the basis of Expert System language CLIPS. Complex cable harnesses can be checked for compliance with arbitrary electrical, mechanical and economical rules. The use of Expert System functionality is very comfortable due to the sophisticated pre- and post-processing functions. Similar to the natural language, IF-THEN rules can be formulated and applied to cable harness systems.

As it can be seen from the description of complex software systems, many different specialists are involved in the development process. Correct and optimal organization of the process is crucial for achieving success.

3. Software Engineering Process

Any software development process begins with specification of basic requirements for the final product. When it comes to upgrading of an existing version, then new requirements must be formulated, which define a changed project scope. For software products described in the previous section, we deal with this particular case. These products are being developed for several years and constantly require innovations and new functionality. The introduction of new features and technologies does not always go smoothly. This often leads to the need for significant changes in software architecture to meet the new requirements and conditions.

Requirements are formulated based on market analysis and requests from customers, using earlier versions of the product. Since the described systems are positioned as commercial products, targeting certain categories of customers, the goals and development objectives are mainly generated by marketing group, which monitors and analyzes the current market trends, determines which requirements must be implemented, and what can be postponed to the future releases.

Based on the specified requirements, the list of tasks is compiled, which basically involves changes and additions to the existing functionality, improved user interface and other required features. The list of tasks is then extended to the basic implementation plan. When creating a general plan, sometimes it is difficult to determine what amount of work has to be performed during each task, so entire development period is divided to several main stages, each ending with a specific milestone [6]. Within these stages, a preliminary assessment of work is performed and more realistic schedule is determined. Milestones provide an opportunity to get a limited, but working version of the product at some stage during development. At the same time Milestones help to evaluate the performed activities and identify general problems of the development process.

The number of milestones should not exceed 4, including the phase of beta testing of the product. The duration of each phase should be limited to 1.5 - 2 months. This is a flexible approach for easily switching from development version to beta testing stage, when for some reasons, the product must be released earlier, or it is necessary to correct task lists in case of a risk of falling behind the schedule.

During development, resources for tasks are allocated according to their priority and complexity. Higher priority and time-consuming tasks must be performed at the early stages and should be included in the first and second milestone versions. Priority should be given to the tasks, which introduce new important functionality, because such approach provides additional competitive advantages to the product. It should be noted that the priority might also be affected by the complexity of the task. Since the complex systems consist of a set of subsystems, or modules, often changes may affect multiple subsystems. Sometimes improvements and optimizations may be considered for modules that already have passed an extensive testing. As a result, the task becomes complicated and requires a careful approach for further development. In order to maintain the overall quality of the product, number of scheduled time-consuming tasks should be kept to a minimum as development process reaches milestone of beta-version preparation.

Practice has shown that when working on a specific task, if the estimated time required for the task exceeds 3-4 weeks, better results can be achieved by breaking it into several subtasks.

As it was already mentioned, a subtask may require development, or changes in the different software modules of the system. Different modules within a single task developed and managed by independent groups, often lead to the loss of overall control, failure to meet the scheduled deadlines, reduction of product quality, etc. The main reason for this is the problem of communication between groups that may occur due to objective reasons. Past experience, gained during development of the mentioned software products shows us that when working plans of each independent group are poorly coordinated, the development process slows down. Since these products are being developed during several years and many different specialists were being involved in the process at different stages, situations occurred when there has been a conflict between software modules due to the inconsistent approaches or development technologies.

Problems may also arise in case, when the requirements contain specific scientific information. Usually, when such requirements are considered, scientific experts can initiate theoretical researches in order to investigate tasks in depth. After researches, a prototype of calculation module is created and verified for reliability. Information about research results and overall task must be clearly formulated to all groups involved in the implementation of the subtask. Therefore, if the work is carried out in parallel, there is a need for scientific experts to participate in all working groups at the same time, which may be physically impossible.

Human factor is also very important in software development process. Conflict of interests between the group leaders due to technical or organizational issues may seriously harm the development process.

Naturally, increasing the time for discussions and agreements on various aspects of development can minimize such problems, but frequent discussions will inevitably arise the idea of bringing together the necessary human resources under the single management mechanism and uniting the all these works as a one whole.

Our practice has shown that separate project team should be created for each subtask (project). As mentioned above, building a complex scientific system needs resources, knowledge and experience of people of different professions. Typical human resources that are needed for such projects are: project manager, graphical interface developers, computational methods developers, technical writer, graphics designer, tester, customer (or product manager), and science field experts.

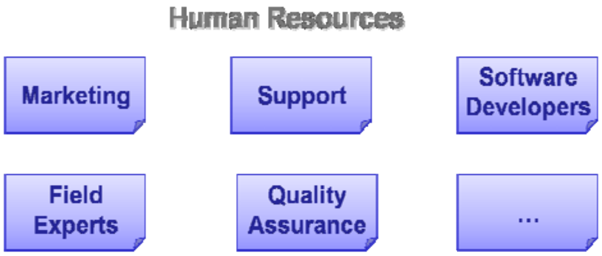


Fig. 5. Available human resources

The project team is created from the available human resources (Fig. 5). Specialists in different areas are involved in the process, depending on the main goals. During team creation a situation may arise, when the necessary human resources are members of other projects teams. In this case, based on the priority of the task the decision should be made to postpone the new task until all the necessary resources are available, or launch a new project, pausing the current project(s). Such decision should be made as a result of agreement between all responsible persons involved in the current and upcoming projects. This means that the working team is not a constant group of developers, but is created "on-the-fly" for each specific project.

For example, lets consider following requirements: it was decided to create new calculation solver to introduce new optimized methodic for calculation and analysis of specific electromagnetic problems. New solver must be included in system, therefore GUI support for this solver must be provided. This means that several changes must be made in system interface: user must be able to set up the task and run calculation with different parameters, using this solver and system must give possibility to analyze results of calculation in 2D/3D post-processing of software. Based on this description, we can decide who must be included in project group in order to fulfill task. These must be: developer of solver and field expert (sometimes is the same person), developers of graphical user interface (2-3 persons, because changes are separated on several modules – calculation module, post processing module etc.), technical writers (describe new solvers and changes in GUI), testers (no more than 1-2 persons – testing and benchmarking of solver and testing of GUI can be done separately at the beginning of project, but overall testing must be conducted at the end). Accept mentioned persons, project manager and product manager are included in the group as well. So, we have got a project group consisting of maximum 8-10 persons. Here we considered complicated task, but usually tasks are not so complicated and in one project group there can be up to 5-7 persons at average.

Given the above recommendations, described problems can be eliminated within a project team. Estimated total amount of work, the execution flow and the role of each participant of the project team can be clearly stated and agreed. A single person - project manager - is responsible for the project during development process.

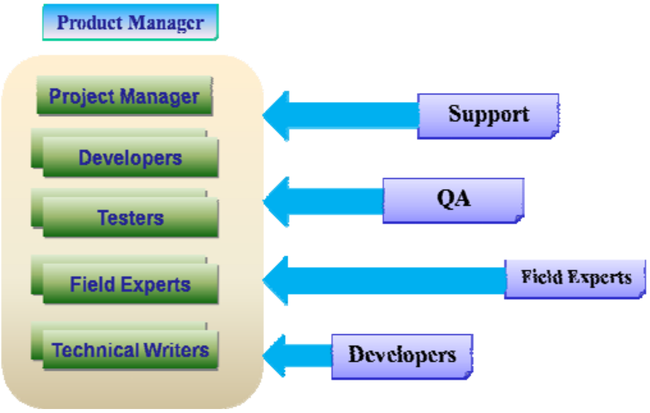


Fig. 6. Project group creation

Each project manager can choose any approaches of development from the wide range of existing methods, including extreme programming model, because one of the main conditions of this approach is met - a customer (product manager) is a member of the project team. For the described products project managers have been using iterative development techniques. Based on our experience, the average period of iterations should not exceed one week, or 10 days at the most. Iteration includes planning, development and testing. The main planning works are done during the first iteration, and are subject to revision at the beginning of each subsequent iteration. Testing can begin by the end of the first iteration and must be performed continuously until the completion of the project.

For improving control and eliminating misunderstanding, the development process must be well documented. The documents usually include proposal, software requirements specification, GUI design presentation, architectural design, technical documentation, overall description of the project and test report. Each project team member is responsible to prepare documentation concerning its part of process, but project manager is the person who accumulates all documents, archives them and passes to all interested departments (for example, to quality assurance department, or support department).

When stage of beta release arrives, the intensity of product's quality control must be increased, at the same time reducing the intensity of the development. After beta version release, all processes related to implementation of new features must be finished.

4. Conclusions

Present paper describes some recommendations for development of complex CAD/CAE systems. These recommendations have formulated based on experience that has been gained in a specific environment. Using mentioned rules and recommendations in practice, helped us to improve overall development process thus making it more flexible and well controlled.

Despite the fact that described approach does not apply for universality, formulated ideas can become the basis of more general recommendations for development processes of similar complex software products

There are some questions, not mentioned in this paper, which should be optimized and improved, so researches in that direction will be continued.

References

[1] Winston W. Royce, "*Managing the Development of Large Software Systems*", Proceedings of IEEE WESCON, 1970

[2] Alistair Cockburn, "*Agile Software Development*", Addison-Wesley, 2001

[3] Kent Beck, "*Extreme Programming Explained: Embrace Change*", Addison-Wesley, 1999.

[4] EMC Studio Manual, Version 5.0, EMCoS Ltd., http://www.emcos.com
[5] Harness Studio Manual, Version 4.0, EMCoS Ltd., http://www.emcos.com
[6] Ed Sullivan, "*Under Pressure and on Time*". Microsoft Pr., 2001

George Sergia: PhD research scholar at the Exact and Natural Sciences department, Tbilisi State University. Area of research is software engineering and development of complex scientific CAD/CAE systems. Has a Master degree in Computer Science from Tbilisi State University. Awarded with "Best Scientific paper" by Georgian National Scientific Fund for Paper, presented on WSEAS European Computing Conference (ECC'09). Currently works as a project manager in one of the leading EM software development companies, EMCoS (www.emcos.com), Georgia

Alexander Demurov: PhD research scholar at the Exact and Natural Sciences department, Tbilisi State University. Area of research is development of complex CAD systems for simulation of PCB problems. Has a Master degree in Computer Science from Tbilisi State University. Currently works as a product manager in one of the leading EM software development companies, EMCoS (www.emcos.com), Georgia

George Petrosyan: Has received his Master degree at the Exact and Natural Sciences department, Tbilisi State University in 2010. Area of research is development of expert system for analysis of automotive cable harnesses. Currently works as a project manager in one of the leading EM software development companies, EMCoS (www.emcos.com), Georgia

Roman Jobava: (M'97) was born in Georgia, in 1965. He received the M.S. and Ph.D. degrees in Radiophysics from I. Javakhishvili Tbilisi State University, Tbilisi, Georgia, in 1987 and 1990, respectively.

Since 1996 he has been working at I.Javakhishvili Tbilisi State University as Associate Professor. From 1987 to 2000 he was a Senior Researcher in Laboratory of Applied Electrodynamics of the same University, responsible for computer simulation of transient electromagnetic phenomena and development of numerical and software tools. In 2001 he began leading EM Consulting and Software, EMCoS Ltd. His research interests include numerical methods in applied electrodynamics, computer simulation of EMC problems, ESD, and transient field calculations, software development for solving EM/EMC problems.