Lightweight Process Documentation: Just Enough Structure in Automotive Pre-development

Kai Stapel[1], Eric Knauss[1], and Christian Allmann[2]

[1] FG Software Engineering, Leibniz Universität Hannover
Welfengarten 1, 30167 Hannover, Germany
[2] Audi Electronics Venture GmbH
Sachsstraße 18, 85080 Gaimersheim, Germany
{eric.knauss,kai.stapel}@inf.uni-hannover.de,
Christian.Allmann@audi.de

Abstract. Pre-development in the automotive sector is informally organized to support the engineers trying out new ideas and generally being creative. If feasibility studies reveal system's uncertainties or bad market opportunities and the development has to be discarded, all documentation attached is obsolete. As a result it is neither possible nor desirable to establish a document centric process in automotive pre-development.

However, without a defined development procedure it is hard to improve and validate development outcomes, respectively to repeat success strategies as well as to integrate new personnel. Furthermore, if new innovations do pass to series development, system characteristics and development activities certainly have to be documented.

In this contradictory situation we cannot apply traditional document centric process approaches. Instead we make use of our Information Flow Analysis. This way it is possible to document and analyze the pre-development activities. Based on our conclusions we developed lightweight concepts to systematically capture documentation from the engineers, without hindering their creativity. These concepts were incorporated in a semantic Wiki, in an effort to give a suitable starting point for comprehensive documentation in case of pre-development projects going into production.

1 Introduction

The role of automotive pre-development is to evaluate and demonstrate the technical feasibility of new ideas and technologies as the base for client and market studies. Generally speaking, pre-development is similar to research in general. The main difference between research and pre-development is the timeline. All pre-development activities focus on the contemporary (up to 3 years) vehicle deployment. Both have in common that the starting point can be described as a development on the "green field" [1, 2]. Meaning that the amount of requirements is indefinite, the technical realization is unknown, the identification of all relevant stakeholders is still ongoing, and the client usage can not be demonstrated. Therefore, the main goal of the development activities is to build up a system demonstrator regularly. In automotive pre-development this usually is a vehicle prototype. As in conventional systems engineering and software

R.V. O'Connor et al. (Eds.): EuroSPI 2008, CCIS 16, pp. 142–151, 2008.

engineering, these prototypes prove whether a technical solution is reasonable or not. The main challenge assembling the vehicle prototypes is caused by the implementation of new technologies (e.g. sensors and image processing) and by networking formerly independent functions. Both rely on a growing number of increasingly powerful and highly integrated mechatronic components. The amount of functions and the extent of network interaction are only two indicators for system and thus pre-development complexity. Another dimension originates from the diversity and the overlapping of development domains where a growing number of systems are a compound of mechanical, electronic and software components. Traditionally, the activities within these disciplines are carried out separately, often within independent departments at the original equipment manufacturer (OEM) and the supplier. This further increases pre-development complexity and demands high efforts for the communication between all involved parties and the coordination of the system development.

In case a new technology and system development verified by a vehicle prototype does not demonstrate feasibility, the collected experience still has to be documented in order to avoid developers to try the same again. However, in such a case comprehensive documentation should be avoided, because it does not add value to the final product.

A difference of prototyping in the automotive context in contrast to prototyping in conventional software engineering is that developers responsible for the prototypes are not necessarily responsible for the series product development. Therefore, if an innovation is proved to be valuable for series deployment, comprehensive documentation of this technical invention is needed after all.

Because of these conflicting aspects (creative and innovative solution finding vs. good documentation of proved solutions) the process of pre-development is a challenging object of research. On the one hand these projects need to be organized, because they contribute to the company's success. Therefore, it is important to improve chances of success. Consequently, a process model is needed to make these projects' success repeatable. This is a typical goal of traditional process- or maturity models.

On the other hand traditional process modeling approaches are not designed to deal with the exchange of experiences and technical know-how in coffee-corners. Because of the creativity involved in pre-development a specification of a strict sequence of activities and process artifacts is undesirable. The main task is to make relevant information available to all project members, support exchange of information between projects, and help project leaders to cope with an informally organized process. We also strive to create an overview of pre-development projects, to give new project members some orientation.

In this context we applied our Information Flow Analysis [3, 4] in order to capture the information flows between project participants. We were able to derive a project map that shows which information flows are important for success in pre-development projects. It turned out that especially the documentation of project experiences is difficult in this context. As a result, we enhanced a Wiki-Web application to better support these crucial information flows (e.g. introducing templates for quick documentation of experiences and observations).

This paper is organized as follows: Section 2 gives some foundations and shows how our approach distinguishes from other approaches. Section 3 contains the design of our study as well as the foundations of Information Flow Analysis. The actual analysis with its findings and highlights is presented in section 4. We give a short account of the direct benefits and consequences from the analysis in section 5. Finally, we draw our conclusions and give an outlook of open research questions and future activities in this area.

2 Related Work

Today, most of vehicle or system innovations arise from the intelligent integration of individual, complex (sub-)systems into the vehicle network which must fulfill vehicle constraints like real-time, safety and dependability aspects. On the other hand these (sub-)systems nowadays are developed by a network of suppliers together with the OEM. The manufacturer is in the role of the client and at the same time manager of the system to be developed. He defines the requirements and monitors the project through testing. One conflict arises because on the one hand the OEM is depended on the specific knowledge of the supplier when it comes to individual components (e.g. specific sensor devices). On the other hand the supplier depends on the OEM's knowledge of the environmental conditions he has to comply with, like construction space, physical ambient parameter, dependencies to other systems or the power supply. Therefore the interaction of OEM and the supplier has to be considered as a key factor for success in this kind of projects. Naturally both sides have a strong interest to protect their intellectual property. So an efficient development is not only difficult to realize due to system complexity, but also because of the gap of communication resulting from the diverging views of OEM and suppliers [5, 6].

At first glance applying a formal process model like the Rational Unified Process [7] or V-Modell XT [8] seems to be a good method to coordinate interactions between OEM and supplier. But, a strict process requiring a lot of documentation rather hinders creativity and innovation needed in pre-development instead of enabling them. Thus, our approach is a lightweight method to document the already working informal process. The method is called Information Flow Analysis [3, 4]. The resulting information flow map can be used as a base for process improvements. In this context lightweight means two things: First, the method to collect the information flow data and create the according model is easy to learn and easy to use. Furthermore, with the method it does not take a lot of time to get an overview of a process. Second, lightweight means process shaping without bothering the developers too much. Regular processes provide a lot of benefits but also demand a lot of duties. In the automotive pre-development context a lightweight process with fewer duties provides a better cost-benefit ratio.

3 Study Design

In this section we present the study design that we used to conduct the information flows in an automotive pre-development context. The basis for this is our Information Flow Analysis, which therefore will be introduced first. After that the actual study design will be presented. Finally, some notes on the execution of the elicitation are given.

3.1 Information Flow Basics

Information is the most important resource in development projects, this applies particularly to pre-development. The correct flow of information is essential for project success and product quality. Especially in pre-development information is not only being passed on by documents but also by e-mails, phone calls, and direct communication like in meetings.

Information Flow Analysis is our approach to conquer the challenges with different ways of information exchange. Information flows are modeled, analyzed and optimized based on the following fundamental concepts:

- *Information appears in different states. Fluid* information is verbal or non-objectively reproducible information including e-mails and personal notes third parties cannot access or reproduce. *Solid* information refers to written or recorded information (like documents or videos) which is long-term accessible even to third parties.
- *Experience is a special kind of information* which is being modeled explicitly. It often influences activities and acts as a catalyst. Experiences made in one project can be of value in others.
- *Coarse modeling of information content.* Just the type of information is being modeled, e. g. product requirements, experiences, or development decisions, not the exact content.

In order to be able to note down the information flows of a given project a simple, easy to understand, and easy to use notation is needed. All relevant aspects of information and its flow need intuitive representation. The following Information Flow Notation was designed to fulfill these needs:

Table 1. Information Flow Symbols

information state	store	information flow	experience flow	activity
solid	<identifier>	<information type> (optional)	<experience> (optional)	
fluid	<identifier>	<information type> (optional)	<experience> (optional)	<activity>
combined solid-fluid	<identifier>	<information type> (optional)	<experience> (optional)	

Especially these aspects were addressed:

- Means of expression for state of information for both, information stores and information flows. An information store can be *solid*, represented by a document symbol (since a document is the most prominent solid information store), or fluid, represented by a smiley symbol (fluid information is stored in peoples' minds).

Information flows can be *solid* and *fluid* as well. The state of an information flow is determined by its originating information store: If the originating store is solid the flow is also solid (represented by a solid line), if the originating store is fluid the flow is fluid (represented by a dashed line).
- Means of expression for experiences. To be able to distinguish experience flows from other information flows, experience flows and stores are depicted in a different color (e. g. gray).
- The ability to establish connections between information flow models and process models. The activity symbol is available in both notations and therefore acts as a connection point (all process notations have a symbol for activities/functions).
- Very easy understandable even for non computer scientists. Few, easy explainable symbols are used.
- Fast and Easy to use. Especially for pragmatic reasons the combined solid-fluid store and flow were introduced. In a few special cases it is not that important in what state certain information flows, it is just important that it flows at all.

With the Information Flow Notation and the basic principles of Information Flow Analysis in mind we designed the elicitation.

3.2 Elicitation Design

Besides the Information Flow Analysis concepts we incorporated the automotive pre-development context in our elicitation design. The following general assumptions express that.

General Assumptions
1. A fluid information culture is important in automotive pre-development.
2. Rework has to be done in case a new approach makes it to serial production because of sparse documentation in pre-development.
3. There are reoccurring project patterns indicating a certain process despite of the fluid information culture and the informal process.
4. Interviewees will be more open when interviewed by an intern staff member.

Based on these assumptions we designed the study. With the given time and resource constraints collecting data via interviews was most promising. As interviewees several project managers and engineers of different pre-development projects were selected.

Elicitation with interviews is a bottom up technique. Each interviewee gives a low level view on the project. In the following analysis step these local views are put together to build up a global pre-development view which then can be used as a base for further discussions and improvements.

Each interview consists of two parts:

1. A catalogue of general questions: employee background, project, involved roles and persons, general information flows, experiences, what works good, what does not, etc.
2. Information Flow data entry forms: For each work-task of an interviewee a form has to be filled. Data about the task, the executing role, required information, supporting information, and outgoing information is being collected.

To verify assumption 1 questions concerning the distribution of verbal and written communication or how the work result gets to its users were incorporated in the catalogue of general questions. Because of assumption 4 an intern staff member was used to conduct the interviews. A problem with that is that industry staff usually is not familiar with Information Flow Analysis concepts. Therefore we had to train the intern staff member.

3.3 Elicitation Execution

A widely accepted intern was selected as the intern staff member to conduct the interviews. We taught her the Information Flow Analysis concepts in a personal instruction. It turned out that the training of the new concepts only needed short time to be successful. Thus, we conclude that the Information Flow concepts are easy to understand. This adds to the fact that Information Flow Analysis is a lightweight method.

In total 6 interviews were held with staff members from 4 different pre-development projects. The intern scheduled and conducted the interviews herself and reported the results to us. We then analyzed each interview and built up the global information flow model.

4 Information Flow Analysis

After the elicitation phase the raw data from the interviews had to be analyzed to build up the global view. The results are presented in this section. First the unmodified answers from the interviews are presented. After that our interpretation and aggregation is given.

4.1 Results

The interviews confirmed assumption 1 (see section 3.2.). There is a mainly fluid information culture. The interviewees claimed that between 60% and 90% of the information being shared is fluid. Everybody stated that most information is exchanged in meetings. It was also stated that there is few required documentation in pre-development.

Assumption 3 could also be confirmed. Despite of working in different projects the interviewees mentioned many tasks that were common among the projects:

- Organizational and project management tasks
- Information search tasks
- Supplier analysis tasks
- Vehicle prototype construction tasks
- Coordination tasks with departments

Most interviewees summed up, that pre-development intern communication mainly works well, but coordination with departments often causes difficulties because of unknown competencies and missing documentation.

4.2 Aggregation

In this section we present the results of the interpretation and the aggregation of the raw interview data. This is the final and most important step in Information Flow Analysis.

First we created a communication network showing the staff and their direct communication paths without documents. Generally speaking, from such a communication network one can see who is talking to whom.

We then build a generic pre-development project model, because several tasks common to all projects could be identified (assumption 3). From this an employee new to pre-development can be instructed or a project manager setting up a new pre-development project can create an instance of the generic model and use it for e. g. project planning.

Experiences were not systematically used throughout all projects. Especially in frequently reoccurring tasks this leads to a lot of rework. A typical frequently reoccurring task in automotive pre-development is the construction of a vehicle prototype to test new approaches under real world conditions. A generic test vehicle construction process as derived from the interviews is depicted in figure 1.

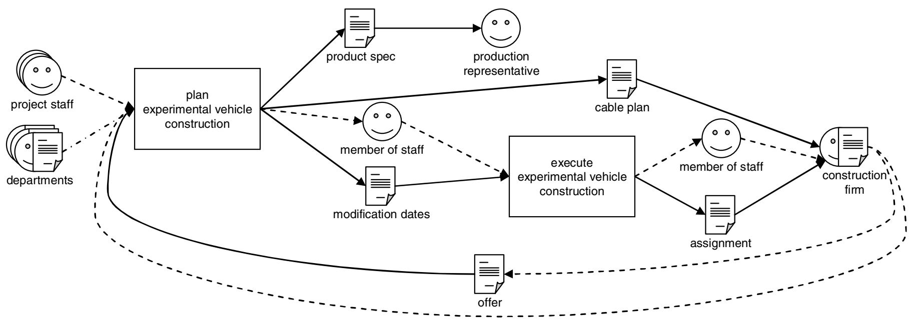

Fig. 1. Information Flow Model of Present Vehicle Prototype Construction Process

The vehicle prototype construction task is divided into two main activities: plan and execute. The planning activity incorporates information from project staff, departments and the construction firm. The outcome is a product specification, some calendar entries and a cable plan for the vehicle. The execution activity is mainly concerned with the assignment of the construction firm with the actual construction of the vehicle. The offer and other results from the construction firm are incorporated in the planning activity of the next iteration. In particular it is noticeable that no experiences are shared to be helpful in following iterations.

Based on these observations, we drew some direct consequences. On the one hand, we enhanced an existing Wiki-Web to better reflect the information flow demands. On the other hand we showed our results to the developers and started a discussion of the pre-development process model.

5 Drawing the Consequences

In the opinion of the analyzed pre-development department clear documentation between the roles on the side of the pre-development team, adjacent departments and involved suppliers is needed. Methods and tools like a customized Wiki are supposed to map both the horizontal and the vertical information flows to avoid that communication becomes the bottleneck of development.

In this context the horizontal flow represents the organizational hierarchy (development departments) and the vertical depicts the supplier chain. Therefore, all relevant stakeholders, developers as well as manager decisions, process steps, etc. can be mapped to the actual development state. Hence, the Information Flow Analysis revealed communication bottlenecks, knowledge drains and loosely coupled documents. This unbalanced relationship between stakeholders, documentation and system development activities call for suitable methods facilitating the communication, knowledge and project experience acquisition in pre-development projects. Scenario based development like it is described in [13-16] can be used to bridge the gap between less documentation and documentation centric approaches.

For future development in highly innovative fields, like hybrid technologies, x-by-wire or sensor fusion, documented project experience is inevitable. The mentioned distributed information flow is very sophisticated to organize and to lead. Passing experience on to further development means sharing knowledge, the knowledge about the actual developed system as well as the project experience each team member has collected over the years. The illustrated clash of development "philosophies" requires some kind of cooperation platform to organize the different development activities, determine the development chain and teams responsibilities. *Time-shared* and *artifact-shared* development is mandatory concerning the effort for system's development under the pre-development circumstances.

All mentioned implications can be subsumed to one essential point: a common platform for planning, documenting, and representing pre-development projects is crucial. To avoid misunderstandings this platform comprehends solutions for arising communication overhead, distributed access to the different knowledge basis, project management, and tools. A Wiki has proved to be a pragmatic technology for such a platform and has already been used in some projects at the analyzed pre-development department.

This specific Wiki is designed around scenarios. It is based upon an extension of the MediaWiki [9] and a semantic Wiki extension [10]. With the help of this extension it is

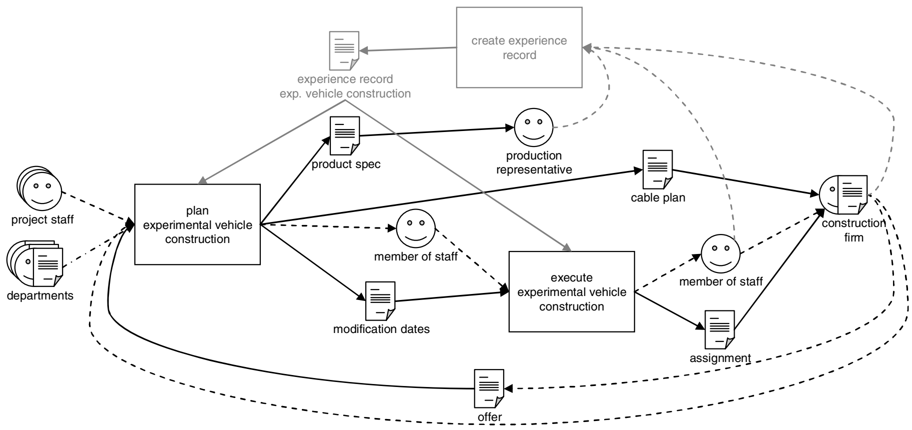

Fig. 2. Information Flow Model of Improved Vehicle Prototype Construction Process

possible to define templates as well as relations between pages in the Wiki (e.g. templates that allow scenario-based requirements engineering, project-management activities, and reports).

Based on our findings we decided to improve this Wiki: For example, the Information Flow Analysis revealed improvement potential for documentation of project experiences.

Figure 2 shows which information needs to be included in experience reports. Consequently we introduced links to observation and experience report templates at the according positions in the Wiki. For example, if the production representative reads the product specification, she has a direct facility to leave some experience-related remarks on that specification. Note that it is important to reduce the effort of documenting experiences in order to improve the chances of a developer actually doing it. [11, 12]

The documentation of experiences is only one of the challenges of learning organizations. The other big challenge is to apply relevant experiences to the development activities. In the example above we introduced overviews of existing experiences into relevant Wiki pages that relate to the activities of planning vehicle prototype construction and its execution.

6 Conclusion and Outlook

In this paper we showed that Information Flow Analysis is a fast and lightweight method to reveal and document informally organized processes like pre-development processes in auto industry. While traditional approaches proved to be unsuitable, we were able to capture the important aspects of pre-development with Information Flow Analysis. This is because the Information Flow concepts enable to distinguish between different states of information. The specialized Information Flow Notation makes it possible to depict non-document-based communication that is particularly important and common in pre-development.

The results of Information Flow Analysis give a good starting point to build supporting information systems. In our study we were able to enhance a Wiki system to better support the important information flows. While this Wiki is a good prototype for this area, further work in IT-support for pre-development is needed. On the one hand, effort for documenting project relevant information has to be further reduced. On the other hand tool support has to ensure that enough documentation is created to support the start of production. In our opinion, a good strategy of lightweight tools is to support developers doing their daily tasks while systematically storing additional documentation as a by-product.

Generally, our experiences show that an Information Flow Map of a pre-development project is a valuable artifact. It serves as an important foundation for discussion. This way such a map can lead to faster education of new project members or to process improvements.

References

1. Weber, M., Weisbrod, J.: Requirements Engineering in Automotive Development: Experiences and Challenges. IEEE Software 20(1), 16–24 (2003)
2. Allmann, C.: Automotive Pre-Development, requirements management based on the green field? Softwaretechnik Trends 27(1) (2007)

3. Stapel, K., et al.: Improving an Industrial Reference Process by Information Flow Analysis: A Case Study. In: Münch, J., Abrahamsson, P. (eds.) PROFES 2007. LNCS, vol. 4589. Springer, Heidelberg (2007)
4. Schneider, K.: Software Process Improvement from a FLOW Perspective. In: Learning Software Organizations Workshop. Springer, Kaiserslautern (2005)
5. Allmann, C., Oppermann, N., Kovacevic, S.: Simulation-driven functional safety evaluation of embedded automotive systems. In: 8th International Stuttgart Symposium "Automotive and Engine Technology", Stuttgart, Germany (2008)
6. Allmann, C., Winkler, L., Kölzow, T.: The Requirements Engineering Gap in the OEM-Supplier Relationship. Journal of Universal Knowledge Management 1(2), 103–111 (2006)
7. Kruchten, P.: The Rational Unified Process: An Introduction, 3rd edn. Addison-Wesley Professional, Reading (2003)
8. VMXT, V-Modell XT (Version 1.2), Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung (2006), `http://v-modell.iabg.de/v-modell-xt-html-english/index.html`
9. Wikimedia-Foundation, MediaWiki, http://www.mediawiki.org
10. Wikimedia-Foundation, Semantic MediaWiki, `http://semantic-mediawiki.org`
11. Basili, V., Caldiera, G., Rombach, D.H.: The Experience Factory. Encyclopedia of Software Engineering. John Wiley and Sons, Chichester (1994)
12. Schneider, K.: LIDs: A Light-Weight Approach to Experience Elicitation and Reuse. In: Bomarius, F., Oivo, M. (eds.) PROFES 2000. LNCS, vol. 1840. Springer, Heidelberg (2000)
13. Allmann, C., Oppermann, N.: Lightweight requirements management in the automotive pre-development. In: Software Engineering 2008, Munich, Germany (2008)
14. Jacobson, I.: Object Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, Reading (1992)
15. Benner, K., et al.: Utilizing Scenarios in the Software Development Process. In: Proceedings of the IFIP WG8.1 Working Conference on Information System Development Process, Amsterdam, Netherland (1993)
16. Sutcliffe, A.: Scenario-based requirements analysis. Requirements Engineering Journal 3(1), 48–65 (1998)