Effective Project Management of Small Satellite Projects from the System Engineer’s Point of View, An Example of the Small Satellite Flying Laptop Project

By Toshinori KUWAHARA, Albert FALKE, and Hans-Peter RÖSER

Institute of Space Systems, Universität Stuttgart, Stuttgart, Germany

(Received April 24th, 2008)

The number of the small satellite projects is recently dramatically increasing and there are great demands for effective project management methods for them. The goal of this paper is to propose effective project management methods for small satellite projects, which are obtained through the real-life experience of the small satellite Flying Laptop project. The project management methods implemented in this project maximize the advantages of rapid and cost-effective small satellite approaches. The management of the project is based on project breakdown structures, which are derived from a combination of several existing standards and empirical methods. These management methods use a product tree as the backbone of the management architecture. The project management activities, such as the establishment of a work breakdown structure, drawing and documentation management structures, time scheduling, and cost management is described with real-life examples. Applications of project management tools, including open source software, which play important roles in cost-effective small satellite approaches, are also summarized and examples of them are illustrated. Finally, further possibilities of effective project management with up-coming new management tools are discussed.

Key Words: Project Management, System Engineering, Modeling Language, Small Satellite

1. Introduction

Recently, small satellites are of increasing interest all over the world for their attractive applications. An advantage of small satellites is the fast and cost-effective development possibility, which makes them a suitable platform for rapid new technology evaluations and demonstrations. On the other hand, small satellites have indeed cultivated their new innovative approaches in the field of earth observation and scientific experiments.

Project management is a coordination of resources to achieve a goal within a restricted time. For satellite development projects, project management plays a significant role strongly connected with all aspects of the projects, such as engineering activity, personnel affairs, financial management etc. Consequently, the traditional project management procedures such as that proposed by European Cooperation on Space Standardization (ECSS) are sometimes too over-specified for small satellite projects which are for example undertaken by universities or academic institutions. One good example of this is the customer-supplier hierarchy in this standard, which is not necessarily required or suitable for small satellite projects. It is the matter of course that the implementation of these over-specified procedures into small satellite projects causes a delay of project schedule and a rise of unnecessary costs and human resources. After the beginning of the trend of cube-sat, which is proposed by professor Twiggs of Stanford University, the number of small satellite projects is dramatically increasing and, accordingly, a considerable amount of them are educational purposes. According to these backgrounds, there are great demands for effective methods of project management for small satellite projects, enabling newly joining institutes to start their projects smoothly and supporting existing projects to be provided with further possibilities to stabilize and optimize their activities.

The Institute of Space Systems (Institut für Raumfahrtsysteme: IRS) is currently conducting a small satellite development program. The earth-orbiting 120kg-class small satellite Flying Laptop is the first satellite within this “Stuttgart Small Satellite Program,” in which the IRS is developing four small satellites aiming to launch the last one to the moon (LUNAR MISSION BW13) as the final goal. All these satellites are developed, built and operated by the institute. The Flying Laptop is a test bed for an on-board computing system with a reconfigurable high computational capability based on Field Programmable Gate Arrays (FPGA’s).

The Flying Laptop Project is currently driven by several doctoral students, tens of under graduate students and technical and electrical engineers. The time schedule, financial budget, personnel resources are managed under doctoral candidates’ responsibility. This paper describes the established project management structures and effective project management methods for small satellite projects, which are developed by authors, working as doctoral candidates. These methods reflect the real-life experience within the project and are especially suitable for small satellite projects with academic backgrounds.
Table 1. Satellite mass category

<table>
<thead>
<tr>
<th>Category</th>
<th>Mass [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large Satellite</td>
<td>&gt; 1000</td>
</tr>
<tr>
<td>Medium Satellite</td>
<td>500 – 1000</td>
</tr>
<tr>
<td>Mini Satellite</td>
<td>100 – 500</td>
</tr>
<tr>
<td>Micro Satellite</td>
<td>10 – 100</td>
</tr>
<tr>
<td>Nano Satellite</td>
<td>1 – 10</td>
</tr>
<tr>
<td>Pico Satellite</td>
<td>0.1 – 1</td>
</tr>
<tr>
<td>Femto Satellite</td>
<td>&lt; 0.1</td>
</tr>
</tbody>
</table>

Fig. 1. Statistics on launch mass of small satellites

Fig. 2. Statistics on user of small satellites

Fig. 3. Statistics on objectives of small satellites

2. Statistical Study on Small Satellite Projects

The definition of the satellite mass category is listed in Table 1. (For the estimated cost according to the satellite mass, refer to Barnhart, D. J. et al.11) Those satellites with a mass of less than 500 kg are defined as small satellites. In this sense, a satellite which is smaller or equal to the category “Mini Satellite” can be regarded as a small satellite. The Union of Concerned Scientists provides a satellite database which lists all satellites which have been ever launched since 1967 and are currently in operation. In this chapter, a statistical study on small satellite developments is summarized based on the knowledge on 7th January 2008. Relevant small satellites are selected from the database and analyzed. The project management methods presented by this paper are based on the experience gathered by the mini satellite Flying Laptop project, but the principle can be also applied to projects which are classified in even smaller categories.

The mass of launched satellites relative to their launch year is illustrated in Fig. 1. It can be seen that, the number of small satellites has been clearly increasing after the initiation of the Cube-sat projects in the year of 1999.

Fig. 2 illustrates the user type of the small satellites. As illustrated in the figure, the number of user type “civil” is recently increasing. This category includes universities and academic institutions. These statistics show a trend that there are many universities or academic institutions which have lately started and are currently conducting small satellite projects.

Fig. 3 shows the objectives of those small satellites. As illustrated, the communication satellites are dominant ones and most of them are amateur communication satellites. The following objectives to this are technology development, scientific research, and earth observation/remote sensing. According to this statistical study, it is clear that a large number of academic institutions are now developing small satellites for technical and scientific applications. Because, in most cases, those academic institutions had no experience on small satellite projects, it is significantly important that they are provided with enough information to conduct the projects efficiently.

3. Proposed Project Breakdown Method

3.1. Adjustment of standards

Applications of project breakdown structures basically provide a project with following benefits:

- Creation of common understandings by identifying items, associated tasks, and responsibilities.
- Identification of interfaces.
- Management of configuration and recording of changes.
- Enhancement of the effective project management.

There are several space project management standards and methods, which provide practical information on project management. One good example of those standards is the one specified by the ECSS.8) This standard specifies the application of project breakdown structures as graphically summarized in Fig. 4.
The first level customer’s requirements are the input of the project breakdown structure. These requirements are translated into a functional breakdown structure, at first, and then into a product breakdown structure and a work breakdown structure (WBS) in sequence, supported by system engineering processes. The WBS represents a summary of work packages, in which the descriptions of identified specific works, human resources and time durations are described precisely. This WBS becomes the backbone for other breakdown structures such as the cost and business breakdown structure, as well as for the derivation of the other project management activities, such as schedule management, documentation management and so forth.

This project breakdown method works quite efficiently in an industrial customer-supplier relationship, where the WBS is indispensable in order to make business agreements and contracts. This WBS ensures that the supplier fulfills specified requirements within a specified time schedule. On the other hand, however, there are some demerits of using this kind of standards in conjunction with small satellite projects. It is mainly because the WBS shall be almost perfectly established at the project’s initial phase, which is quite difficult and needs overwhelming efforts for small satellite projects in most cases. This is very obvious in students-driven projects because the responsible persons for project management are students and their turnover rate is very high, which means that well-experienced students leave the project with a few year interval. It is often the case that there is nobody who can spend time for managing this WBS through the project life time.

3.2. Characteristics of small satellite projects

Requirements of a small satellite are mostly derived from the own mission objectives because most of the small satellite projects are not in the customer-supplier relationship. According to these top level mission objectives functional requirements of the satellite can be derived. In case of students-driven projects, the personnel allocated to the WBS may change continuously and there is no need to allocated personnel cost. In this case, the project budget is directly allocated to the procurement and development of the hardware components. In this sense, therefore, a business agreement structure does not exist in this type of small satellite projects.

3.3. Application of a product tree as the central element in the project breakdown structure

The most important point of this proposed effective project breakdown method for small satellite projects is to place a product tree as the backbone of the whole project management activities, as illustrated in Fig. 5. This product tree represents the hardware and software components of the satellite system. Product trees can be treated as two separate hardware and software trees but each component listed in those trees shall be globally uniquely identified with an identification number. For the establishment of this product tree, deliberate functional analysis shall be performed in order to implement all mission requirements into a functional architecture, which is then translated into a corresponding product tree. System engineering activity shall support this process to establish an appropriate product tree.

3.4. Merit of using a product tree in the center

Putting the product tree in the center of the project management activities brings a lot of merits:

- The personnel management becomes indirectly and business agreement structure can be neglected.
- The WBS does not play a central role anymore and the effort for establishing it can be minimized.
- The time schedule can be directly derived from hardware development, making it flexible and agile.
- By associating obtained intellectual properties to a corresponding hardware component, the management of project properties becomes simple and clear.

There are more possibilities of using this structure for an effective project management. An example of the application is described in the next chapter in detail.

4. Applied Project Breakdown Structure of Flying Laptop Project

4.1. Derivation of requirements

In most cases, the requirements of small satellites are directly derived from payload components. It is also often the case, that an academic institute, which has been developing a specific instrument, starts a small satellite project in order to evaluate the instrument in space environment. In these kinds of cases, the top level requirements are defined by the institute itself. Requirements may consist of a variety of different kinds.
of requirements such as, technical, operational and environmental requirements as well as constraints such as launcher interface and financial budget. Before going to functional analysis, top level requirements can be decomposed into lower level requirements, which makes up a requirements tree. This requirement tree evolves rapidly at the first half phase of the V-model, a traditional project development phase model, and shall be maintained during the whole life time of the project.

4.2. Functional breakdown structure

Through a functional requirement derivation, the top level requirements are decomposed into lower level functional requirements and this decomposition builds up a hierarchical breakdown of the functional requirements. The functional breakdown structure can be represented as hierarchical matrices or integrated block diagrams. The functional breakdown of the Flying Laptop is done using SysML modeling language. The detailed discussion on this topic is given in Chap. 5.1.

4.3. Product tree

A Product tree is a thorough definition of the system elements and represents the hardware oriented breakdown of the system into successive levels of products, based on the functions identified. In case of the Flying Laptop satellite, it has only one on-board computing system and there is no need to divide this product tree into a hardware tree and a software tree, or their combination. Because the FPGA-based on-board computing system of the Flying Laptop does not have any software, but has hardware logic only, it is just denoted as “hardware tree” in the project (Fig. 6). This hardware tree is the core of the project management and the methods mentioned here are based on this hardware tree. It is convenient that we use this hardware tree as the backbone of the project management, because usually small satellites consist of a small number of components which allows a clearly arranged hardware tree. Most of the satellite development activities can be associated directly to one of those components. The personnel responsibility is also allocated to hardware units. The hardware tree gives one identical hardware number for identical hardware products.

4.4. Work breakdown structure

The project breakdown process goes through WBS elaboration. For this purpose, it is still convenient to divide the satellite system into several sub-systems and classify all hardware components in the hardware tree into the sub-systems. In this way, for example, one doctoral candidate can be assigned to one of these sub-systems.

4.5. Cost

The cost is allocated to the hardware tree. In case of students-driven project, the influence from personnel cost to the project total cost is much less. In this sense, the project cost is simply the cost for hardware development including tests, hence all cost can be allocated to the hardware tree. Attention shall be paid how to evaluate the cost for self-developing software or software tools.

4.6. Scheduling

The project scheduling plays also an important role for small satellite projects. One of the great advantages of small satellite projects is their agility, which enables rapid technology demonstrations and evaluations in space. The project scheduling for small satellite projects shall be organized to maximize this advantage. However, planning a time schedule for small satellite projects is quite troublesome because of many types of uncertainties, such as personnel resources, financial resources and testing opportunities. In addition to these, it is common for small satellites that the launch date is tend to be fixed at a late phase of the project.

The solution for these problems is not the application of a GANTT chart but of a PERT chart as the main scheduling method, especially for the initial phase. According to the time duration estimated for each hardware development and their order, a PERT chart can be established as illustrated in Fig. 7. Based on this chart the critical path can be identified and the progress of the project can be maintained by monitoring this critical path. Although, a GANTT chart is necessary for defining project phases, the use of GANTT chart does not necessarily accelerate agile project management because of its effort-requiring maintainability. It is quite difficult to establish a meaningful plan without having any experiences and trends from past projects. The PERT chart represents rather a logical order of the work packages and the packages already done can be simply erased showing the present status of the project. It is common that one includes appropriate margin to establish a GANTT chart, but the logical connection of PERT chart does not necessarily need the idea of “margin.” The PERT chart therefore can accelerate the progress of the project.
Table 2. Example of numbering structures

<table>
<thead>
<tr>
<th>Breakdown Structure</th>
<th>Numbering Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware Number</td>
<td>MMMSS - BBB</td>
</tr>
<tr>
<td>Documentation Number</td>
<td>XXX-Type - MMMSS - NN- EEE-RR</td>
</tr>
<tr>
<td>Drawing Number</td>
<td>XXX-ZZ - MMMSS - BBB - DD-RR</td>
</tr>
</tbody>
</table>

4.7. Documentation and drawing management

An example set of requirements of documentation management can be seen in ECSS standard. For the documentation management of the Flying Laptop project, following numbering structures are introduced (Table 2). This structure is based on the hardware number “MMMSS-BBB” (Fig. 6), where:

- MMM : Module Number.
- SS : Sub-module Number.
- BBB : Component Number. 000 is the assembly.

The documentation number uses the first part of the hardware number as illustrated in Table 2, where:

- Type : Document type. A list shall be prepared.
- MMMSS : Hardware module/sub-module Number.
- NNN : Consecutive number for same Type/MMMSS.
- EEE : Editor organization.
- RR : Issue/Revision Number.

All documentations can be allocated to certain hardware in this way. If the topic of the document can not be assigned to certain hardware it receives the top number, which is “00000.” For example, a Technical Note which describes about the interconnections of a star tracker receives “FLP-TN-00100-001-IRS-10.”

Similar to this documentation numbers, the numbers for technical drawings are also derived from the hardware number adding some additional options (Table 2), where:

- ZZ : Model Type. EM, FM, and so forth.
- DD : Drawing Type.

The drawings can be directly allocated to the corresponding hardware components in this way. These documentation numbers and drawing numbers can be integrated into one documentation structure if necessary. However, the numbering structure tends to become longer, involving unnecessary information.

5. Project Management Tools

5.1. SysML

5.1.1. Introduction

SysML stands for “Systems Modeling Language,” which is specified by the Object Management Group (OMG) as a general-purpose graphical system description language for systems engineering applications. SysML uses a subset of UML and provides additional extensions. It supports object-oriented modeling methods, known from modeling languages like UML for software architectures, for complete technical systems including both hardware and software.

SysML utilizes a variety of different kinds of diagrams such as block definition diagram, internal block diagram, state machine diagram, activity diagram and much more. Supported by these diagrams, SysML enables most of the system engineering activities through the project life time such as specification definition, requirements management, functional analysis, interface definition, operational planning etc. including model verification of a wide range of systems with graphical representations. Thanks to this integrated functionality, SysML encourages the reduction of risk and uncertainty in the iterative system engineering processes and brings better solutions to produce and manage engineering artifacts.

5.1.2. Application example on Flying Laptop project

SysML is implemented in Flying Laptop project. The implemented model is illustrated in Fig. 8. The first diagram is the block definition diagram “bld,” which represents the same satellite components illustrated in above Fig. 6. In this way, the product breakdown structure can be represented as a block definition diagram, which enables the integration of a project management structure and engineering model by SysML into one single system.

The second diagram is the state machine diagram “stm.” In this example the satellite’s operational modes are illustrated. This state machine diagram can be decomposed into lower level state machines and they can be associated with a corresponding functional flow diagram, which is described below. This decomposition of the satellite behavior makes up the functional breakdown and enhances software implementations.

The third diagram is the enhanced functional flow block diagram “efbld” of the satellite system, in which functions performed by the system, assembly and components can be illustrated. This diagram can illustrate both serial and parallel execution of functions and even items which are exchanged between functions. Besides these, this diagram becomes the basis of failure and risk analysis.

The last diagram is the internal block diagram “ibd” of the star tracker unit. The connection between star tracker main electronics and two camera heads, as well as the connection with power control and distribution unit are illustrated. This diagram describes interactions between components, which enhances the implementation of electrical architecture design of the whole satellite system.

As described above, SysML helps project management activities from wide variety of aspects.

5.1.3. Further possibilities of SysML

A study on architecture framework for small satellites shows that spacecraft’s physical architecture, not only hardware components but also software components, can be represented using layer structures: mechanical, electrical, data protocol and data content layers, based on block definition diagrams and internal block diagrams.

Fig. 8 illustrates that it is possible to integrate most of the system engineering activities into SysML structure. After the definition of the SysML specification, a lot of companies are now developing SysML software tools as an extensional function of UML software tools.
Once it becomes possible that all kind of engineering parameters allocated to each product such as mass, size and power consumption are managed using certain kind of integrated database, SysML becomes not only a comfortable system engineering tool but also a great project management tool. On the other hand, the state machine diagram, which is one of the block diagrams of SysML has the possibility to be used for software reliability prediction. By using these tools, system budget can be easily maintained and the concurrent engineering activities can be further accelerated. Recently, companies such as ARTiSAN studio, Aonix etc. are working on the development of SysML software tools.

5.2. Application of open source software

5.2.1. Advantages of applying open source software

To achieve low cost approaches for small satellite development projects, utilization of Open Source Software (OSS) plays important roles. OSS is also seen by many in the software industry as a mechanism to promote cost effective re-use and indeed it can reduce development costs and schedules. The selected items of useful project management tools for practical implementation for small satellite projects are listed in Table 3. All these tools are implemented in the **Flying Laptop** Project.

OSS is developed in a public and collaborative manner, being driven by public interests, and the human-readable source code is made available under copyright licenses. The feature of free of charge enables cost effective implementation to projects, which leads to reduction of projects’ over-all cost. Furthermore, OSS can be changed, improved and extended by the customer’s demands and even be redistributed afterward. Consequently, the supporting body of OSS becomes quite huge having developers spreading around all over the world. Thanks to this, OSS is mostly very rich in supporting languages as well as in interfaces between other software tools providing plenty of plug-ins. It is seldom that the development of OSS is canceled, and therefore, projects can rely on its high availability.

The application of software tools generally plays powerful roles in the following aspects:

- Concurrent engineering.
- Smooth information sharing.
- Secure documentation.
- Prevention of duplicated expenses.
- Automation of procedures.

In addition to this, a combinational use of OSS can dramatically enhance the efficiency and quality of development activities. An example application for **Flying Laptop** project is described in the following chapters.

5.2.2. Documentation management

The official documentation software tool which is applied for **Flying Laptop** project is OpenOffice.org. It provides sufficient documentation capabilities including a drawing tool. It is also practical that OpenOffice.org directly supports pdf-file export. Internal documentations of the project are centrally stored in and managed by means of the Concurrent Versions System (CVS).
Table 3. Open source software tools

<table>
<thead>
<tr>
<th>Category</th>
<th>Tool</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduling</td>
<td>Open Proj* (Projity)</td>
<td>Provides compatible GANTT &amp; PERT charts</td>
</tr>
<tr>
<td></td>
<td>PHProjekt</td>
<td>Group activity coordination, Information sharing</td>
</tr>
<tr>
<td>Documentation</td>
<td>OpenOffice.org</td>
<td>Flying Laptop Project</td>
</tr>
<tr>
<td></td>
<td>- Text</td>
<td>introduced OpenOffice.org as the standard documentation tool. *Drawing is only for simple diagrams</td>
</tr>
<tr>
<td></td>
<td>- Spreadsheet</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Presentation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Drawing</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>MySQL</td>
<td>OpenOffice.org Base is used as the user interface</td>
</tr>
<tr>
<td></td>
<td>OpenOffice.org - Base</td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td>CVS</td>
<td>Enables effective concurrent engineering</td>
</tr>
<tr>
<td>Management</td>
<td>MediaWiki</td>
<td>Useful for knowledge management and information sharing</td>
</tr>
<tr>
<td>Intellectual</td>
<td>Eclipse</td>
<td>Software Programming</td>
</tr>
<tr>
<td>Property</td>
<td>OpenAmeos</td>
<td>UML Modelling</td>
</tr>
<tr>
<td>Programming</td>
<td>Linux</td>
<td>Provides suitable simulation environments</td>
</tr>
</tbody>
</table>

Even though CVS is originally designed for software development, it can be applied for normal documentation and drawing managements. CVS ensures secure documentation and keeps track of all work and all changes in a set of files and allows multiple developers to collaborate, so that the project members can work concurrently.

What is remarkable at the Flying Laptop project is that the MySQL database maintains the following information with the database user interface of OpenOffice.org Base:
- Documentation numbering.
- Drawing numbering.
- Hardware tree.
- Satellite Configuration.
- Parameters of satellite components (position, mass, size, consumption power, cost, physical properties etc.).

In this way, all documentations and drawings are maintained in the database and associated with each other through the numbering structure described in Chap. 4.7. All of these elements are centrally well managed and are forming the basis of the documentation management activities.

5.2.3. Programming and software management

A combinational use of OSS enables integrated software engineering as illustrated in Fig. 9. The software engineering facility of the Flying Laptop is established by the UML modeling tool OpenAmeos and the programming environment Eclipse, based on the operation system of OpenSUSE Linux. OpenSUSE Linux provides a suitable programming environment for both tools. Input information to the software engineering is based on the documentations established as described in the previous chapter. OpenAmeos UML design tool is used for graphical modeling of the satellite’s software model which can be translated into software frameworks by the use of a code generator. Based on the automatically generated software frameworks by OpenAmeos, the satellite software model is refined under the development environment of Eclipse for simulation purposes. Eclipse is a powerful programming tool and able to work directly together with CVS. The software codes developed at this stage are stored in and managed by CVS.

The source codes of the satellite’s on-board computer are developed with a specific commercial development tool. These are translated into hardware logic and implemented into the computer. This computer can be integrated in the above mentioned simulation environment in a hardware-in-the-loop configuration for simulation and verification purposes. The simulation environment can be configured by means of XML configuration files which read out required information from the database. Simulation case specific conditions and additional parameters such as component mass, size, position and the required amount of power supply can be configured. These source codes of the computer and XML configuration files are also stored in CVS.

According to the progress of the development of software tools for SysML, many of the above mentioned engineering activities, namely specification definition and documentation, graphical software modeling, and central management of information in the database, can be further integrated into a compact activity as illustrated in Fig. 9.

5.2.4. Schedule and activity management

Open Proj* is applied for managing the project schedule. This tool provides scheduling capability compatible to both scheduling methods mentioned in Chap. 4.6, and the project phase, timelines and logical relationship between work packages are managed by this tool. Schedules are classified into subsystems and then divided into hardware component items identified in the hardware tree. Each work package can be allocated to the corresponding item.

It shall also be emphasized that the organization of project activities is very closely related to maximizing the agility and flexibility of small satellite projects. For organizing group activities and individual schedules of project members, the groupware PHProjekt is applied.
PHProjekt is a web-based modular application for the coordination of group activities and enables the information and documents sharing. It supports group calendars, project management, contact information manager, mail client and many other modules.

5.2.5. Internal intellectual properties management

A part of acquired intellectual properties during the project is stored and summarized in a single knowledge source based on MediaWiki. It is shown that this kind of database is very useful for students-driven projects in terms of a smooth takeover of project and organization specific information. Especially, new coming project members can learn the past history and present status of the project by their own.

These project management structures described in the above chapters are based on OSS tools. By making the most of these tools, project management and even development of satellite components including software can be dramatically reinforced. It is especially beneficial for middle size small satellite projects with ambitious goals and a restricted financial budget.

7. Conclusion

The statistical study on recent small satellite development activities illustrates that the number of small satellites with academic backgrounds is significantly increasing. In this paper, the effective project management methods for those small satellite projects are proposed based on the experience obtained through the real-life small satellite project Flying Laptop. This paper describes the methods in sufficient depth and provides further information in order to maximize the merits of rapid and cost-effective small satellite approaches. The proposed project management methods use the product tree as the backbone of the whole project management activities, which are adjusted from existing project management standards. This method allows simple structuring of the project activities indicating the association with the corresponding hardware components. This paper also describes the derivation method of product trees and the relation with other management activities. Moreover it suggests the combinational use of modeling language SysML in order to integrate system engineering activities and project management activities in an effective manner. Application examples of SysML are illustrated and the further possibilities on the future application are discussed. Finally, selected items of open source software, which can be effectively implemented into small satellite projects, are listed and the implementation examples of them are described. A special emphasis is put on the establishment of a software engineering facility based on a combinational use of open source software tools and it is illustrated that it can dramatically enhance the efficiency and quality of development activities.

References