Systems Engineering Value Stream Modelling

Dieter Scheithauer
Breitensteinstr. 26, 83727 Schliersee, Germany,
dieter.scheithauer@hitseng.eu

Abstract. The V-Model denotes the overall systems engineering value stream. The value stream consists of configuration baselines referencing the information handed over across the system architecture for top-down design and bottom-up integration. In addition, system integration preparation is depicted by direct flows from the left leg to the right leg of the V for each system. In order to exploit the full power of value stream thinking, the value streams for the generation of each particular configuration baseline have to be considered. This paper justifies the precedence of the value steam approach compared to document centred and process oriented approaches applied to particular documents. Configuration management in systems engineering is adjusted with respect to the value stream based approach. The process definition model for defining the value streams for generating individual configuration baselines is fully described. The paper concludes with considerations how to establish these work product generation sequences in practice.

Introduction

In his book on product development flow Donald G. Reinertsen uses models from queuing theory to derive compelling evidence for the benefits of controlling the flow in development (Reinertsen 2009). He establishes a high amount of considerable criteria for implementing efficient product development flow. However, the queuing models applied are fairly simple compared with the intricacies of the systems engineering process. He pays little attention on assuming or proposing specific development process models. Similarly, he is rather brief in advising how to manage the product development flow in practice. Nevertheless, the conclusions drawn are matching observable patterns from systems engineering projects in the real world.

Complementary, systems engineering is rich in terms of product development process models and corresponding systems engineering management approaches. Systems engineering is originally based on a comprehensive flow model comprising the whole system life cycle. However, it would go too far to claim that all proposed and widely accepted systems engineering process models are fully compatible with a strict application of the flow principle during the whole system life cycle, or, even limited, during product development. Instead, many systems engineering narratives emphasise just a part of the whole product development flow, and are positioned in competition and partly in contradiction to each other. From the systems engineering history, requirements engineering, system architecting, and model based systems engineering may be taken as prominent examples for competing and sometimes contradicting systems engineering narratives addressing in each case just a different part of the whole product development in isolation.

Other systems engineering process models and narratives focus on the whole product development flow indeed. At first, the V-Model has to be mentioned (Forsberg and Mooz 1991; Forsberg, Mooz and Cotterman 2005). Lean systems engineering adopts lean thinking principles from manufacturing and applies them to development (Oppenheim 2011). And, agile management approaches are borrowed from software engineering (Pichler 2010).

Strictly aligning the V-Model with the lean principles known from lean thinking (Womack and Jones 1996 and 2003) has led to an interpretation of the V as the overall systems engineering value stream (Scheithauer and Forsberg 2013). In this interpretation, the overall systems engineering value stream is defined by the hand-over of configuration baselines, see Figure 1. On the left leg of the V, systems and system elements are defined by established configuration baselines before allocated requirements are forwarded to the system elements on the next lower architectural level. In parallel, configuration baselines are also required to allow the preparation of systems integration. On the right leg of the V, configuration baselines from system elements on lower architectural levels are evaluated to assess the integration readiness of a particular system, and to start integration eventually.

Note, that the left-to-right direction is labelled as logical sequence, and not as the time line. This indicates that the overall value stream may be executed subsequently or concurrently several times during the development or, for further evolution, later in the life cycle of a particular product. With this provision, the V may be easily applied in conjunction with concepts of batch-size reduction, or of continuous improvement and learning typical for designing innovative products. The implications of iterations become clearly visible within the V. When re-entering the V, all activities right from the re-entry point may be impacted by changes.

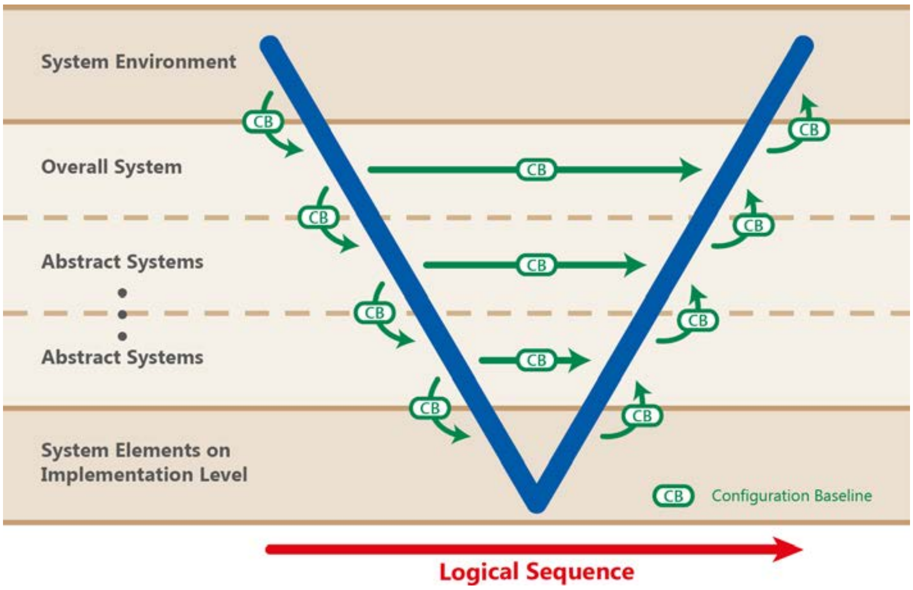

Figure 1. The Flow of Configuration Baselines in the V-Model

Below the coarse grained level of the overall systems engineering value stream another type of more detailed value streams exist: the systems engineering value streams for establishing configuration baselines for a particular system or system element. Those systems engineering value streams are designated as work product generation sequences as introduced and described in previous papers (Scheithauer and Schindler 2000; Scheithauer 2012). The earlier paper mainly outlines the process definition model for establishing work product generation sequences when a work product generation sequence was defined for the first time in the context of a development team with more than hundred members. The later paper describes retrospectively the experience gained with the application of work product generation sequences and the resulting benefits for managing concurrency following a value stream based approach. A detailed description of the process definition model for a work product generation sequence with an explanation of the attributes of all model elements has not been published so far.

This paper concentrates on four main topics. First, the benefits of a value stream approach are derived compared to document centred and process oriented approaches focussing on individual documents. Second, the role of configuration management in systems engineering is adjusted with respect to the value stream approach. Third, the process definition model for constructing a work product generation sequence is described in detail. This process definition model features distinct description levels with specific semantics and attribution. Finally, information is provided on how to construct a work product generation sequence. A stepwise approach embedded in setting up a project and interwoven with the team building process is proposed.

The Precedence of the Value Stream Approach

Configuration baselines are manifestations of the system itself. They are mainly referencing information contained in other documents. In addition, they refer to concessions granted due to incompleteness and unresolved issues. Occasionally, configuration baselines may include more elaborated content summarising general system features and essential system boundaries. Nevertheless, this information is only a portion compared to the information contained in all the documents referenced by the configuration baseline.

The quality of the system is one-to-one related to the quality of the configuration baselines describing it. For this reason, it is important to turn to the question how the generation of configuration baselines and the referenced documents is organised and managed. Traditionally, a distinction is drawn between document centred approaches versus process oriented approaches. As a third favourable alternative, a value stream based approach will be included in the comparison below. Three criteria will be applied: (1) How does it work, (2) how it is managed, and (3) what are the consequences in terms of quality and efficiency.

The Document Centred Approach. Figure 2 shows a simple example of a document centred approach. Three documents are shown: a specification, a design document, and an integration summary. Obviously, the information content of the three documents is dependent on each other following a logical sequence. According to a work breakdown structure of the project, statements of works and reasonable resources are allocated to the generation of each document. Within the project organisation a responsible author is appointed for each document. Due dates for the delivery of each document are defined with a project manager controlling the achievements regarding the documents and the establishment of the final configuration baseline.

Let us first imagine what does not happen in practice usually: Due to the logical sequence, the specification is compiled first. When it is finished, the design is accomplished followed later by the system integration. There is a simple reason why reality turns out differently: efficiency. Let people just wait for work without generating useful results may easily be categorised as waste (Oppenheim 2011). But note that Reinertsen provides evidence that waiting is not the worst category of waste (Reinertsen 2009). Furthermore, following a pure sequential approach on the timeline may lead to a rather late detection of that is something missing in the specification. The remaining time for corrective actions until the scheduled final milestone may be too short. Consequently, the milestone date may slip or the quality of the resulting configuration baseline may be impacted adversely. Not uncommonly, both may happen.

More realistically, work is performed on all three documents in parallel. May be not up from the beginning, but at least when initial information becomes available to allow useful activities with respect to more than one document. From this point onwards, the content of all three documents evolves simultaneously to complete the information, to improve the design, and to incorporate changes induced by upstream documents or due to feedback from downstream engineering activities. These iterations are managed informally only. The project manager is controlling the scheduled document and system delivery milestones. The responsible authors' main objective is to find a compromise to keep those delivery dates and to provide results of reasonable quality. The communication of issues found with respect to a particular document and its implications on the other documents is at the hand of the responsible authors. There is a high risk that the resulting development dynamics exceed the capability of this informal management process.

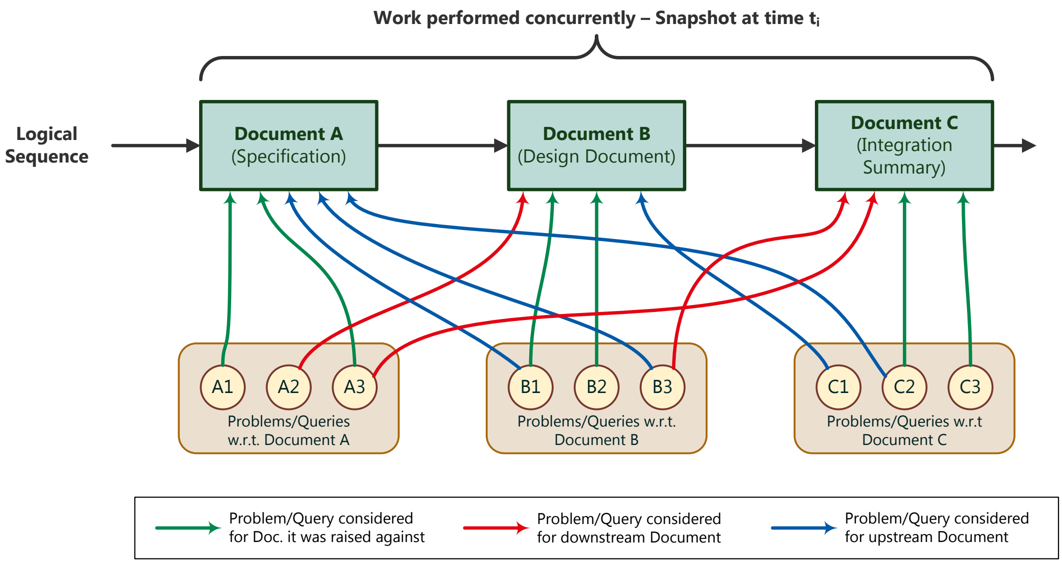

Figure 2. The Document Centred Approach

The Process Oriented Approach. The process oriented approach is depicted in Figure 3. In the process oriented approach, emphasis is put on the process of incorporating changes into a particular document. No change is incorporated in a document baseline without a corresponding change request. The process oriented approach improves the situation with respect to the consistency of the individual documents. Even in the case of high development dynamics, there are proper controls to support the document's responsible author to oversee the consolidated content of the document. With this tool at hand, a tendency may develop to incorporate more and more, even diverse content in a single document.

However, without changing the management style the mutual dependencies between the documents have still to be controlled by the same informal process as for the document centred approach. As before, high development dynamics may result and may overstress the capabilities of such an informal process.

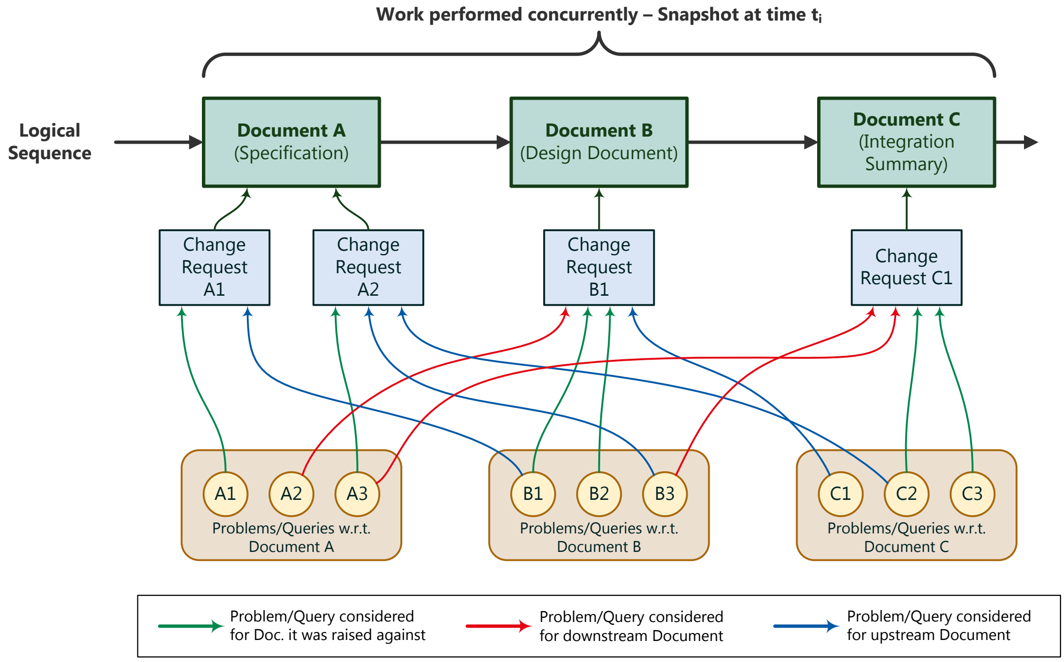

Figure 3. The Process Oriented Approach

A common downside of the document centred as well as the process oriented approach is the inability to ensure a consistent informational state continuously throughout the development. As sketched in Figure 3, there is no guarantee that the document baselines with incorporated change requests are ever able to define a consistent configuration baseline until the very end. Thus, the first opportunity to check the overall consistency and quality of the resulting configuration baseline is at the end of the project.

In the broader context of the system architecture, concurrent engineering suffers from inconsistent and deficient information exchanged across the system architecture intermediately. In consequence, issues are spread over the system architecture. Corrective actions become more cumbersome because they are not encapsulated to a single system element even for issues that could be completely solved in the scope of a single system element. Change control across the system architecture may become saturated with such nuisance corrective actions. Due to this overload, improvements of the system architecture may become unfeasible in especially because a huge pile of more urgent corrective actions is also waiting for processing.

Adverse impacts on the quality and consistency of the final configuration baselines handed over to production at the end of the development are as likely as development milestone delays due to the high amount of corrective actions initiated late in the project. Considering that correction costs increase exponentially as later a deficiency is found, the devastating effect to an organisation’s pricing, profit and finally reputation are imaginable.

In conclusion, the overall systems engineering value stream may only be efficiently executed, if all configuration baselines are consistent and of adequate quality. But the document centred approach and the process oriented approach cannot ensure this. Harvesting the promises of concurrent engineering across the system architecture is jeopardised.

The Value Stream Based Approach. If we aim for consistent and high quality configuration baselines we need to manage them up from the beginning putting the primary focus on them. This demands not at least changes to the management styles applied. In order to end up with consistent and high quality configuration baselines after sometimes years of development, we need to control the evolution of the configuration baseline up from the beginning. In other words, we should be able to establish a consistent configuration baseline everyday throughout development, saying what is achieved and what has to be done with the reference to the corresponding documents. Only a value stream approach as shown in Figure 4 is suited to achieve this without further augmenting controls.

The first fundamental difference of Figure 4 compared to the Figures 2 and 3 is the substitution of the term document by the term work product. This shift is caused by the narrower scope of work products. Adopting the strict value definition from lean thinking, work products contain information that is needed as a point of reference by downstream processes.

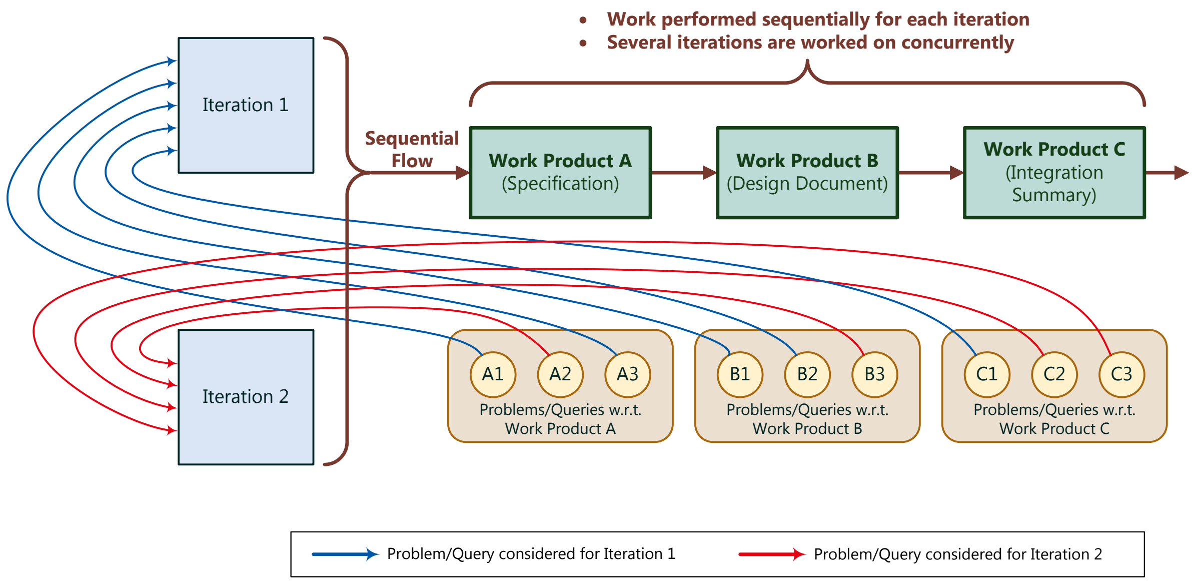

Figure 4. The Value Stream Based Approach

Note, a lot of other information is generated during development as well including change control records, trade studies about solution alternatives, and review reports. We may summarise all this information as supporting data. The difference between supporting data and work products may be simply expressed in the following statement: Work product define what the system is while supporting data provide the evidence why a system has evolved to what it is.

The second fundamental difference concerns the information flow between the work products. For the document centred approach and the process oriented approach, the information flow is more an assumed logical sequence which provides general guidance for the development. In the value stream approach, the work product generation sequence is actually strictly followed. In order to achieve efficiency, the utilisation of work product generation sequences implies controlled iterations performed simultaneously in order to exploit the efficiency benefits of concurrent engineering and batch-size reduction.

With respect to the process oriented approach, it was mentioned that documents tend to become more and more comprehensive because information consistency within a document can be properly controlled. For work products, the scoping of content is driven by other criteria.

First, the number of work products in a work product generation sequence should vary depending from the opportunities for applying concurrent engineering on multiple iterations executed in parallel. For example, it would be a nonsense to define hundred work products for a team of ten engineers. For the same systems engineering activities on a larger scale, defining ten work products for a team of hundred engineers would make no sense as well. In the first case, the administrative overhead would be too high without leading to any reward. In the second case, opportunities for concurrency may not be exploited, and a lot of information flow would again be managed informally leading to the issues introduced above.

Second, the scoping of work products is driven by reducing the waste of waiting. Engineers performing no productive activities still have to be paid. Therefore, they may be forced to do some work deemed to be useful in advance. To some extent this is a fair argument. However, the short term advantage should be calculated against the correction costs implied by introducing inconsistencies resulting from acting without a sound point of reference as input. In most cases, it will be favourable to let the engineers wait and to try to improve the project planning skills regarding appropriate resource staffing, iteration scoping and scheduling, and possibly improvements of the work product generation sequence. It should not be unmentioned that no waiting at all could be less a sign of efficiency, but an indication of overstressing the existing resources and loosing agility to react on unforeseen events in due time (Reinertsen 2009).

The Role of Configuration Management

The value stream based approach fosters the importance of configuration management for systems engineering. Configuration baselines reference work products containing the information needed as point of reference for downstream activities. On the right leg of the V, they refer to the configuration baselines of the system elements on the next lower architectural level as well. In addition to configuration baselines and work products, supporting data has to be kept in the context of the corresponding work products. Supporting data provides the evidence why the system has developed into a particular direction.

In this view, released versions of work products are the smallest unit of value in the systems engineering value stream. All other information management regarding specific work product content is subjected to data management. For identifying individual requirements and their traces or test procedures and test results, specific requirements management and test management methods and supporting tools are employed. In many cases, they enable fine grained control for atomic information entities. But, a uniform configuration management approach considers this information just by controlling the work products containing the respective information. The primary objective of configuration management is to enable efficient value stream management towards consistent and high quality configuration baselines.

So far, the reasoning above provides convincing evidence about the role of configuration management in a value stream based approach to systems engineering. But this narrative may be challenged from two directions. First, in lean systems engineering value is simply defined as information (McManus 2005; Oppenheim 2011). And second, the role assigned to configuration management is extended beyond its traditional role (TechAmerica 2011). Both concerns deserve further considerations.

Value in Systems Engineering. The designation of information as value in systems engineering is well comprehensible to characterise a main difference with respect to value streams in production. In lean manufacturing there is a flow of material while in lean systems engineering there is a flow of information. However, the definition of flow in manufacturing does not stop at this level. Otherwise, it could be claimed that the material itself is value in car manufacturing. Not the metal itself is the value, but for example, a mud guard produced according to its specification respecting allowable tolerances and further quality criteria. Corresponding evidence for the quality is kept in records about tool calibration and measurements performed on the particular mud guard produced.

The analogy to the overall systems engineering value stream and work product generation sequences is quite obvious. A particular car produced is the result of a single iteration on the assembly line like final and intermediate configuration baselines are the results of iterations over a work product generation sequence. Note, according to the system architecture, configuration baselines may also refer to other configuration baselines in addition. This finds its equivalent in sub-assemblies not produced on the final assembly line, but delivered from other sources to the final assembly line. Individual versions of work products are analogous to manufactured parts. The evidence demonstrating compliance with the specification of a part are equivalent to supporting data.

However, any analogy also has its limits. In an ideal lean manufacturing line, every car produced is independent from the other cars manufactured before or after it. In contrast, iterations over the work product generation sequence and the resulting configuration baselines are building up on each other to achieve the full featured system in the end.

Another issue is the view on validation and verification results. The commonality comprises the necessity to check the quality of a manufactured part like to examine the quality of the information contained in any work product. Afterwards, a manufactured part is handed over to the next assembly station, or a version of a work product is released for being used downstream in the work product generation sequence. However, assurance activities in systems engineering also lead to work products on their own, solely concerned with validation and verification like test procedures and test reports. The strong value proposition in lean manufacturing that value comprises only what is delivered to the final customer is not followed here. The economic value of a design after being validated and verified is of course higher than without. With this reasoning borrowed from Reinertsen (Reinertsen 2009), the claim that assurance results are contained in work products does not deviate the value paradigm with respect to a work product generation sequence.

The Scope of Configuration Management. Turning to configuration management itself, the entry point for the following considerations is the definition of configuration management contained in EIA-649-B (TechAmerica 2011), one of the latest standards dedicated purely to configuration management: "*A technical and management process for establishing and maintaining consistency of a product's functional and physical attributes with its requirements, design and operational information throughout its life.*" There is a slight sign of inconsistency in this definition. On the one hand, the definition claims to be applicable throughout the products life cycle. On the other hand, it takes the requirements, design and operational information as inputs. The system life cycle definition within systems engineering sets the starting point of the system life cycle earlier considering the requirements, the design and operating instructions already as outputs of the system life cycle.

The definition of configuration management in the older ISO 10007 (ISO 2003) does not contain this flaw: "[Configuration management comprises] *coordinated activities to direct and control configuration.*" Admittedly, this definition is otherwise hard to understand due to its somewhat self-referencing content. Despite the different definitions, both standards share a common pattern. The input is taken for granted, and it is the task of configuration management to control the correct transformation. This principle fits well to the described role of configuration management with respect to the overall systems engineering value stream and work product generation sequences.

Consequently, the occurrence of issues further upstream in the product life cycle is an abnormal case. Corresponding change control approaches are rather heavy. They are primarily defined for handling justified changes considering the impact on all current achievements and the further product life cycle.

In systems engineering where the understanding of stakeholder needs is steadily improved by performing the systems engineering activities, the claim for static inputs does not hold anymore. Success in a first-time-right manner is undermined. Less heavy change control processes are more adequate for the engineering of innovative systems. Throughout system design new stakeholder needs may be discovered, and recognised needs may be better understood. This causes iterations in systems engineering independently from any flaws in performing the systems engineering activities itself. In any case, a faster commencement of corrective actions minimises the costs for implementing those corrective actions.

In conclusion, this section has emphasised the important role of configuration management for managing the systems engineering value stream. Agile change control activities and strict configuration identification has to be applied to configuration baselines and work products.

Process Definition Model for the Work Product Generation Sequence

For general process modelling, methods like IDEF0 or SADT (Ross 1977) are commonly available. For modelling work product generation sequences, they are less suitable. Without repeating all the considerations contained in the first paper on work product generation sequences (Scheithauer and Schindler 2000) the three main issues are summarised below.

First, IDEF0 and SADT allow the creation of circular dependencies. Circular dependencies are highly discarded in value stream thinking as they compromise the flow. In the systems engineering practice, they lead to deadlocks regularly (Scheithauer 2012). The modelling techniques would need to be augmented by rules beyond the model language semantics.

Second, IDEF0 and SADT set no constraints on the level of detail for modelling processes. This provides users with a high degree of flexibility. But it also may lead to rather unbalanced models. Generated models may be very detailed in some areas, and very brief in others. Usually, they are fine grained in areas where the process modeller has a lot of expertise, and rather coarse grained where this is not the case. Or in other words, the resulting models are rather strict in some areas, and rather relaxed in others. Not uncommonly, they over-specify processes well understood leading to frustration and refusal of those performing those processes. On the other hand, they are not supportive to govern areas with a lot of uncertainty, and for which users of those processes would really like to consult the process definitions. Again a number of augmenting rules would be necessary for modelling a work product generation sequence, and a lot of discipline too for not doing what the method would easily allow.

The third issue is related to the second, but highlights a rather different aspect. As the method can be applied recursively to an arbitrary level of depth, it consequently lacks the opportunity to provide specific semantics to particular modelling levels. This affects the attribution of processes as well as the interpretation of the information flows between them.

The process definition model developed for modelling work product generation sequence avoid the traps of IDEF0 and SADT. It features four distinct modelling levels and two types of flows between the model elements. All have a dedicated meaning with respect to define and to perform a work product generation sequence. A simple generic example of a work product generation sequence illustrates the scheme as shown by Figure 5.

Blocks on all levels have a few common attributes, e.g. an identifier, a name, a responsible, and a description. To use a consistent scheme for identifiers is recommended. The name should be descriptive. The responsible is the human individual who takes care that the process activities comprised by the particular block are taking place in accordance with the defined process and lead to the intended outputs. The responsible may also have the authority to release the outputs, but this is not a must. The expected content of the description attribute varies with the type of block and will be further considered when describing the individual block types below. The content of the description attribute should always be kept short. The description should foster understanding of the purpose of the block in the context of the whole work product generation sequence. Any methodical details should be kept separate in procedures describing the applied methods and their application.

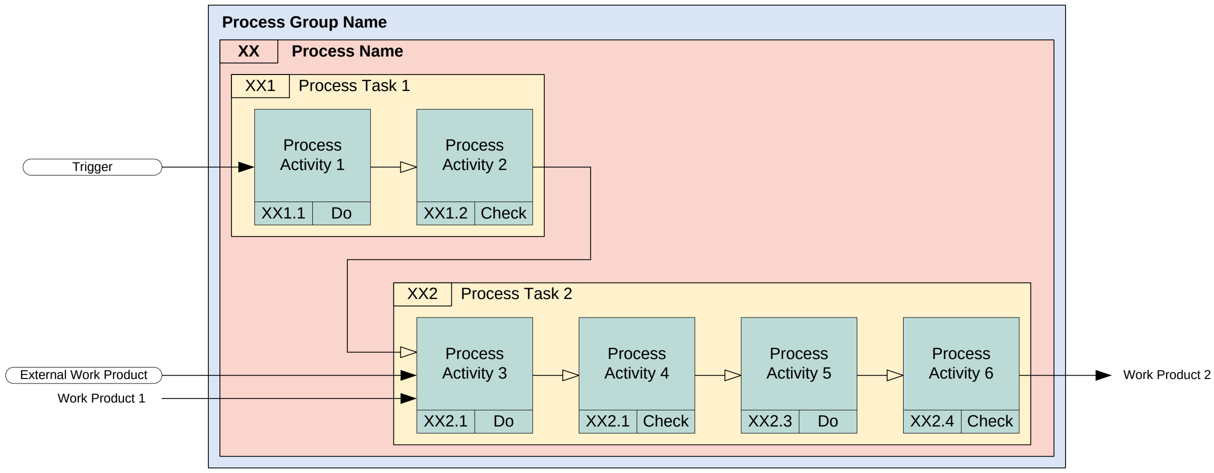

Figure 5. The Process Definition Model

Process Tasks. The essential elements for the work product generation sequence are the process tasks drawn in amber. Process tasks are concerned with the generation of a particular work product like a system requirement document. Sometimes, families of work products may be the output of a single process task like different test procedures following the same principal path in the work product generation sequence.

In extraordinary cases, the output of process tasks may be not work products but supporting data. This is fully justified for modelling the change control part. For other purposes, this is discarded except for modelling legacy processes. For example, if the diverse content of two documents lead to a circular dependency between, migration of existing information may be avoided by some hacks to comply with the semantic rules postulated for work product generation sequences. However, such solutions limit the ultimately achievable efficiency improvement.

If the output of a process task is a work product, this work product should be identified and the description attribute should briefly address the information content of the work product. Further attributes should hold the names for preparing, approving, releasing or authorising, and optionally agreeing to the work product. Preparing means authoring the work product or being the lead author. Approval comprises a check of the work product contents. According to the role of the approver, the checks performed may comprise the examination of the whole content or of individual aspects of the content only. Release or authorisation means to allow that the resulting version of a work product may be used for downstream activities. The different designations have been introduced to allow a further distinction. Release stands for releasing a version of a work product for internal use. Authorising means for binding commitments of the organisation to the outside world. Optionally, individuals not involved in the generation, checking, or releasing activities of a work product may acknowledge the release by their signature as defined to testify their agreement.

In the work product generation sequence, process tasks are connected by a directed graph featuring branching and merging of branches, but without any feedback loops. The resulting overall graph indicates the information flow between the process tasks over the whole work product generation sequence. In particular, it denotes the work product release sequence.

Elementary Processes. Elementary processes group a number of associated process tasks together. An elementary process fulfils a number of objectives attributed to it. It is appropriate to express the activities for example of the stakeholder definition process within a single elementary process.

The main reason to include elementary processes as a distinct level in the proposed process definition model is concerned with flow management. The principle to commence downstream activities only on the basis of clearly identifiable input information with known quality is not implemented on the process task level alone. The dependencies between the process tasks define the work product release sequence. They provide no indication whether the release sequence between subsequent process tasks allows to be matched in a finish-to-finish style, or in a finish-to-start style. The elementary process introduces this distinction. Within the boundaries of an elementary process concurrent work on subsequent work products are allowed as long as the release sequence is obeyed. The flows crossing the boundaries have to consist of released versions of work products.

Of course, this is a relaxation of the basic flow principle. But there are good practical reasons to allow concurrent work in small teams working on a number of distinct, interdependent work products in parallel. Assume an elementary process comprising system requirement definition and system design (Scheithauer and Forsberg 2013). For each of the three complementary system views, e.g. system requirements, functions and architecture, separate work products may be defined. In this case, the release sequence of the three work products alone is not a good model for representing the close interaction to generate and improve the three views, or in other words, the content of the three work products, simultaneously. It would generate massive administrative work and the release of a high number of intermediate versions of the three work products before the design would be mature to hand it over to further downstream processes.

In Figure 5 dependencies are shown inside the elementary processes by lines with unfilled arrows. Filled arrows indicate flows of released versions of work products. They have to be used always for flows crossing the boundaries of elementary processes. Triggers reporting external demands or issues with released information are also crossing boundaries of elementary processes as inputs. Therefore, they are also represented by lines with filled arrows. The frame around the name of the trigger indicates that this is external information not to be found as an output of any process task of the particular work product generation sequence. External work products being inputs to the work product generation sequence are also framed to indicate the same. An example for an external work product are the allocated requirements generated by the upper level engineering team as input for the elementary process system requirement definition and system design of a system element.

Process Groups. Process groups are a means to cluster elementary processes. All elementary processes of a complete work product generation sequence may reside in a single process group. Or, they may be distributed over several process groups that integrated into a higher level process group until the level is achieved where a process group represents the complete work product generation sequence. That means process groups may be nested to an arbitrary level of depth. Process groups are therefore well suited to map organisational structures of development organisations. So far, no more than two process group levels have been defined in practical applications.

For process groups, a change control board may be defined. A change control board is defined by attributes holding a description, the triggers received as inputs to act on, and the different roles populating the change control board. The roles of chairman, change controller, and members are defined.

Process Activities. With the three levels of the process definition model described above the work product generation sequence is clearly visible, but the process is not fully defined yet. Process activities are concerned with the internals of process tasks.

Process activities feature a number of specific attributes: activity category, participants, applied standards, and applied tools. The activity category indicates whether the process activity is a do activity, e.g. generating new information, or a check activity, e.g. assessing the quality of the generated information. There may be a number of different do and check activities hosted by a single process task. Partially, there may be branches to indicate, for example, that some analyses may be performed in parallel. All variations are allowed as long as two constraints are met: (1) A process task must at least contain one do activity and one check activity. (2) The last process activity in the process task has always to be a single check activity.

The attribute participants holds the names of individuals or roles contributing to the job performed in the scope of the particular process activity. For a design activity, the participants may list all application domain and specialty engineering disciplines contributing. For a check activity, all demanded reviewers for the work product's review may be enumerated, for example.

The attribute applied standards refers to all documents describing how the process activity has to be performed. Public standards, organisation wide procedures and project specific procedures may be referenced. This helps to keep the content of the description attribute short, and to re-use assets from enterprise wide process standardisation and process improvement activities.

Finally, the attribute applied tools lists all the tools required and used for performing the particular process activity.

Establishing a Work Product Generation Sequence

The establishment of a work product generation sequence needs to deliver two distinct outputs. Obviously, we have to define the work product generation sequence according to the process definition model introduced above. We will concentrate on the construction aspect below. Secondly, we have to build successful teams. The proposal is to make the establishment of the work product generation an essential part of the team building process as emphasised earlier (Scheithauer 2012).

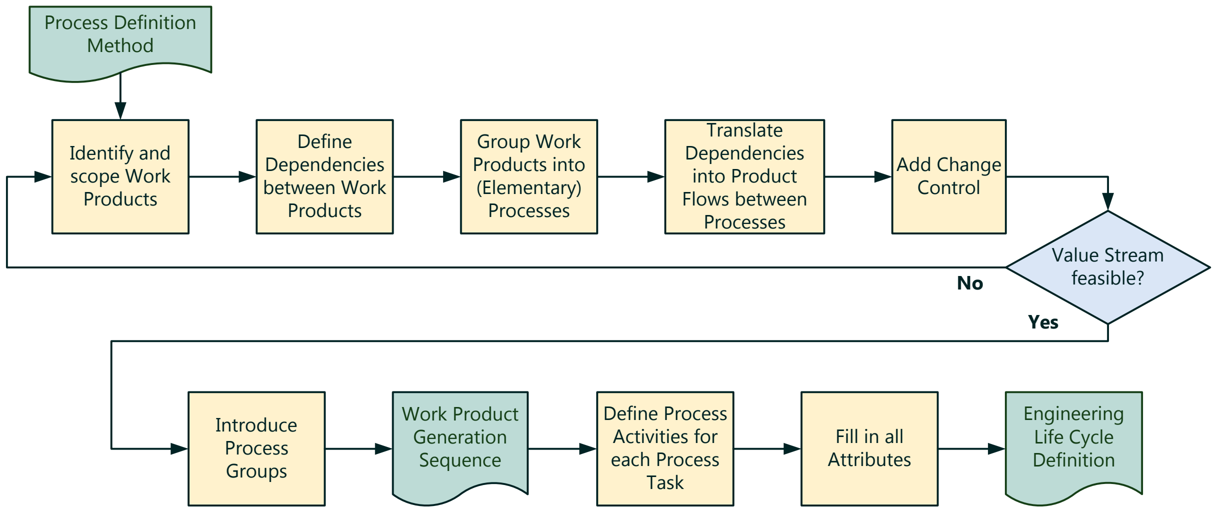

Figure 6. Work Product Generation Sequence Generation

The process for constructing a work product generation sequence is outlined in Figure 6. The sequence is close to the sequence chosen for describing the process definition model above. The first step is to identify the work products to be produced. Basic inputs are applicable process requirements contained in organisational and public process standards, existing work product templates and the document content from similar past projects. Contributing criteria need to be derived from the size of the system to be engineered, resource and time constraints, and the size of the project team. The main objective of the work product generation sequence is to enable development flow and to allow efficient concurrent engineering. The content of the work products have to be scoped appropriately.

When the work products are defined, the dependencies between them have to be clarified. The release sequence of the work products is the result. As soon as circular dependencies become visible, they have to be eliminated. In many cases, this means to go back to the stage identifying and scoping the work products.

Then work products are grouped into elementary processes. Decisions are influenced by process quality demands, the expected content of the work products as well as team organisation and team size. Next the flows have to be categorised as elementary process internal dependencies or elementary process boundary crossing work product flows. The change control process can be added in front of the work product generation sequence. See one of the previous papers for more detail on change control (Scheithauer 2012). Before going further into the details it is important to investigate, if the work product generation sequence is feasible under all quality, time and cost constraints applicable to the system and the project.

Afterwards, the process groups may be defined and agreed. Finally, the work product generation sequence is fully visible, and the process definition is turning to the details of defining the process activities. The definition of the process activities may be accomplished not by the engineering team as a whole, but by the stakeholders within the team concerned with the particular process tasks.

Conclusions

The overall systems engineering value stream marks the starting point of the considerations within this paper. The flow is modelled as the hand-over of configuration baselines across the V. The advantages of a value stream based approach for managing the evolution of individual configuration baselines have been shown compared to document centred and process oriented approaches. Consequently, the demand for work product generation sequences is derived.

The important role of configuration management for controlling the flow according to the work product generation sequence is emphasised. This includes enabling concurrently performed iterations over the work product generation sequence.

A process definition model has been described satisfying the stated criteria. It consists of four distinct levels in the model hierarchy with level specific descriptive attributes and two types of flows. Finally, hints are provided on how to establish a work product generation sequence during the team building process.

The benefits of work product generation sequences regarding the quality of configuration baselines and the support to project planning, reporting and control have been demonstrated in practice already.

References

Forsberg, K. and H. Mooz 1991. “The Relationship of System Engineering to the Project Cycle,” Proceedings of the National Council on Systems Engineering (NCOSE) Conference, Chattanooga, Tennessee, pp. 57-65, October.

Forsberg, K., H. Mooz, and H. Cotterman 2005. Visualizing Project Management: Models and Frameworks for Mastering Complex Systems. 3rd Edition. Hoboken, NJ (US): John Wiley and Sons.

ISO (International Organisation for Standardisation). 2003. ISO 10007. *Quality Management Systems – Guidelines for Configuration Management*.

McManus, H. L. 2005. “Product Development Value Stream Mapping.” Release 1.0, Massachusetts Institute of Technology, Lean Advancement Initiative (Cambridge, MA, US).

Oppenheim, B. W. 2011. *Lean Systems Engineering - With Lean Enablers for Systems Engineering*. Hoboken, NJ (US): John Wiley and Sons.

Ohno, T. 1988: *Toyota Production System: Beyond Large Scale Production*. Boca Raton, FL (US): CRC Press.

Pichler, R. 2010. *Agile Product Management with Scrum*. Boston, MA (US): Pearson Education.

Reinertsen, D. G. 2009. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Redondo Beach, CA (US): Celeritas Publishing.

Ross, D. T. 1977. “Structured Analysis (SA): A Language for Communicating Ideas.” *IEEE Transactions on Software Engineering* 3 (1): 16-34.

Scheithauer, D. 2012. "Managing Concurrency in Systems Engineering." Paper presented at the 22nd INCOSE International Symposium, Rome (IT): 9-12 July.

Scheithauer, D., and K. Forsberg 2013. "V-Model Views." Paper presented at the 23nd INCOSE International Symposium, Philadelphia PA (US): 24-27 June.

Scheithauer, D., and Schindler, A. 2000. „A Standardisation Concept for Non-Standard Development Projects." Paper presented at the 2nd European Systems Engineering Conference, Munich (GE): 13-15 September.
TechAmerica. 2011. EIA-649-B. *Configuration Management Standard*.
Womack, J. P., and D. T. Jones 1996 and 2003. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. New York, NY (US): Free Press.

Biography

Dieter Scheithauer studied electrical engineering with special emphasis on automatic control at the Universität der Bundeswehr München. He received the degree of a Diplom-Ingenieur univ. in 1980 and a doctor's degree (Dr.-Ing.) in 1987. After twelve years, his service as Technical Officer in the German Air Force ended in 1988.

From 1988 to 1999 he has been employed by Industrieanlagenbetriebsgesellschaft GmbH (IABG) mainly delivering technical expertise to the German Ministry of Defence and other government organizations. Throughout this time he contributed in various roles to the flight control system development for major European military aircraft and helicopter programs. Furthermore, he acted as project manager and chief engineer for unconventional airborne and ground-based systems demanding high-integrity technical solutions.

In 1999 he has joined the company that is now the Airbus Group. During the initial years he was in charge of process improvement for the EF2000 flight control system development. Later he held a position as Senior Expert Systems Engineering Processes within the division Airbus Defence and Space. He was influential progressing systems engineering within the division and on group level. In 2014 he left the company.

Recently, he started his own business as independent systems engineering trainer, coach and consultant.

He is a former president and an honorable member of GfSE e.V. – The German Chapter of INCOSE. He became an INCOSE CSEP in 2010, and an INCOSE ESEP in 2012.