

Stray lamb - misalignment in a socio-technical structure of an enterprise when transitioning to intelligent products

Ilia Bider, Sofia Olsson, Erik Perjons

Department of Computer and Systems Sciences (DSV), Stockholm University, Stockholm, Sweden

ilia@dsv.su.se, sofia.at.olsson@gmail.com, perjons@dsv.su.se

Abstract. Products from traditional engineering companies, such as cars and refrigerators, are evolving to become intelligent via combining hardware, software, sensors, and connectivity/networks. In such products, the importance of software increases exponential. Therefore, new software development units are emerging. This may cause misalignment between the traditional parts of the engineering companies and the new emerging software development units. The misalignment concerns both the technical part of the socio-technical structure of the companies, i.e. a conflict between different project management methodologies in the units, and the social part, i.e. the power roles of different units and coordination between them. Viable System Model (VSM) has been applied to study the nature of misalignment in one such company, a large car manufacturer. This paper reports on the experience obtained during this study. With the help of VSM, misalignment was identified and diagnosed as a “stray lamb” – a pathological archetype related to a new and important unit of the company not being properly incorporated into the rest of the system.

1 Introduction

This paper is an experience report on using Viable System Model (VSM) [1] for detecting and analyzing a particular class of misalignment in a socio-technical structure of an enterprise. The misalignment in question is induced by technological changes in the products/services that the enterprise delivers to its customers. More exactly, the misalignment in question is the one that sprang from moving from manufacturing traditional or “dumb” products to delivering “intelligent” products as well as services around them.

The move from “traditional” to “intelligent” products concerns many traditional engineering companies, like Volkswagen and Electrolux, who historically have been developing hardware products like cars and refrigerators. Such traditional engineering companies must now also develop software for these products [2] in order to make these products intelligent. Several of these companies have already been involved in software development, but of different kind - embedded software. The embedded software, for example software that controls the car engine, has characteristics different from the software that is needed for making a product intelligent. The latter

software is highly interactive and need to interact with both human beings and other “intelligent” objects (Internet of Things – IoT).

While the embedded software can be developed, more or less, in the same way as hardware, methods of development of interactive software are different. Interactive software is more appropriate to develop using some brand of agile software development methods [3,4]. As the result, introducing an interactive software development unit into a traditional engineering company may result in misalignment between different parts of the socio-technical structure of the company. This misalignment concerns both technical and social parts of the system. In the technical part, there may appear two radically different methods of project management, a traditional waterfall or milestone method and an agile one. These two are quite difficult to integrate. In addition to these different methods, there may appear two different cultures that correspond to these methods.

People who are accustomed to traditional methods are more prone to specialization and working with explicit knowledge, while agile proponents are more of the generalists kind of workers and do a lot of work on tacit level [5]. These differences are not easy to bridge, and they may lead to misalignment in the social part of a company’s socio-technical structure. This misalignment can lead to personal conflicts and power struggle.

The first step to fix potential misalignment in the situation described above is to pinpoint particular places in the organization where misalignment happens. One way of doing this is via building a model of the organization that shows where the problem lies and gives a direction of how to fix it. As it seems that there is no accepted method to model situations as described above, we decided to test whether VSM [1] will be suitable for this end.

Though VSM is not a brand new modeling technique, it has never reached the mainstream, neither in research, nor in practice. There are a number of practitioners that successfully used VSM in practice and develop practical methodologies for this end [6,7]. However, we are not aware of any publication, research or practical, that applies VSM to the situation in the focus of our investigation. Therefore, the result of our trial could be of interest for both research and practice.

Our trial has an additional particularity that makes it slightly different from other cases of using VSM in practice. We have built a VSM model not for the whole company, but for the product development department/division. Though one of the main principles of VSM is recursive decomposition of subsystems, usually, the recursion is applied to so-called operational units that belong to System 1 in the VSM terminology. The product development department is considered as belonging to system 4 (i.e. intelligent, or future, see Section 2.3 for details). Thus the result achieved in the trial also shows that recursion can be used to decompose not only units that belong to VSM System 1, but also to the units that belong to other levels of VSM (System 4 in our case).

In our trial, VSM was applied in a business case of a large car manufacturer that, while moving to intelligent products, had introduced a new unit/department to deal with the development of interactive software. The case represents an engineering company that struggles to adjust itself to the new business reality, and therefore is

quite suitable for the goal of our trial. The material for building the VSM model was gathered during a case study at the newly introduced department. The material from the case study is fully presented in [8]; in this paper we only give an overview of the main findings that were used for building the VSM model.

The rest of the paper is structured in the following way. In Section 2, we present the background of this research that consists of three parts: (1) short overview of the literature that discusses the problem we deal with, (2) short presentation of business case to which VSM has been applied, (3) short overview of VSM. Section 3 presents material from the case study based on which the VSM model has been built. Section 4 presents the VSM model built for the case and misalignment diagnostics based on it. Section 5 presents lessons learned and discusses the future work.

2 Background

2.1 Transition to intelligent products

As has already been mentioned in Section 1, the products from traditional engineering companies are evolving to become intelligent via combining hardware, software, sensors, and connectivity/networks. This combination of hardware and software is critical for success in the stiffening market competition for such intelligent products [9]. As a result of this development, many new services are provided to the customers, either as separated supplement services to traditional engineering products or as bundled together with the engineering products [10].

In the automotive industry, the importance of software has increased exponentially. Today, up to 40 percent of the production cost of car comes from electronics and software according to [11]. Many new innovative features are based on software solutions, many of them come from the Connected Car concept [11]. In a competitive market, such as the automotive industry, these type of features are utterly important or even decisive [11,12,13,9]. They are often provided as services where a car is bundled in as part of the value proposition towards the customer.

The growing importance of software within the automotive industry has brought a number of challenges to the car industry, see [11,12,13] for examples of such challenges. One such challenge is that new development processes need to be designed and introduced in practice [11]. The traditional mechanical engineering development needs to adapt to the way innovative software is developed [11]. Moreover, in traditional mechanical engineering approaches, independent modules are developed separately to be assembled in the end [11]. This is more challenging when cars are built with more interactive, communicative and autonomous systems. Today, the behavior of the car is more programmable, where different software modules communicate automatically without the drivers direct actions [11]. This requires new ways of developing the cars including new ways of collaboration and communication among developers of different types, such as mechanical and software engineers [14], as well as new ways of leading the overall development process. To the best of our knowledge, there is lack of research about the challenges of and

solutions for coordinating and leading the developments of cars in car production companies when the importance of software and software functionality increases.

2.2 Business case

Vehicle Cooperation (VC) is a large international car manufacturing organization with production plants in different parts of the world.

Connected Car Services (CCS), which has been in the focus of our study, is a new and growing department under VC Group IT organization introduced for developing services related to the Connected Car concept, that is, the ability of a car to share internet access with other devices, both inside and outside the car. Examples of services being developed by CCS are search and payment for parking lots, access to internet radio, remote start and start heating via smart phone, warnings for crashes and notification of speeding.

CCS employs around 100-150 people, a majority of these are contracted consultants. The developers work within a number of autonomous and self-contained teams that use agile development methods. The latter methods focus on rapid and test driven development, continuous work iteration and integration, and intensive customer interaction [4]. The working culture at the department promotes innovation and constant and open communication among the teams.

Besides cooperating with each other, the development teams of CCS need to collaborate with other, more established traditional departments, such as IT, Marketing and Sales, Customer Services, Manufacturing Engineering and Research and Development. The established departments work in a traditional way, with pre-defined controlling structures called Work Breakdown Systems (WBS), while CCS employs agile methods. The difference creates a misalignment that could be harmful for the overall performance of the company's product development. The goal of the project reported in this paper was to diagnose this misalignment and suggest the ways of eliminating it.

2.3 Viable System Model and its usage for organizational diagnostics

Viable system model (VSM) has been developed by Stafford Beer [1] and his colleagues and followers, see for example [15]. It represents an organization as a system functioning within its environment and consisting of two parts: *Operation* and *Management*. In its own turn, *Operation* is split into a number of semi-autonomous operational units, denoted as System 1, that have some communication mechanism to ensure their coordination. The latter communication mechanism is denoted as System 2. *Management*, in turn, is split in three parts, denoted as System 3, System 4, and System 5. Dependent on the author, these systems may be dubbed differently, see Table 1, but they have more or less the same meaning, see the last column of Table 1.

Note that the components listed in Table 1 do not need to coincide with the organizational structure of a particular organization. Different components can be manned by the same people. This, for example, happens in a small enterprise where the same group of people does the job on all levels. The components in this case are differenti-

ated not through who is doing the job, but through the nature of the job done, e.g. policy document writing belongs to System 5, while completing a customer order belongs to System 1.

Table 1. Components of VSM (adapted from [16])

Identifi- cation	Naming	Function
System 1	Operations, Im- plementation, Delivery	Producing and delivering products and services for external customers, thus actively interacting with the environment.
System 2	Coordination	Coordinate work of operational units included in System 1.
System 3	Control, Deliv- ery management, Cohesion [15],	Managing operational units (System 1), and estab- lishing/maintaining coordination mechanism (Sys- tem 2). Making the semi-autonomous units func- tion well as a whole (cohesion) in the current busi- ness environment.
System 4	Intelligence [15], Future	Forward looking adaptation to possible future changes in the environment through identifying trends and preparing to changes or affecting the environment in the desired direction (intelligence). System 4 is considered as including development, marketing and research.
System 5	Identity (man- agement), Policy [15] (manage- ment)	Solving conflicts between System 4 and System 3. Permitting System 4 to introduce changes despite the conservatism of System 3, and not allowing System 4 to change the identity of the whole sys- tem that exists via functioning of Systems 3, 2, and 1. This is done through designing, maintaining and imposing policies that stay in place even when changes designed by System 4 are implemented in Systems 3, 2, and 1.

The viability of the system with a structure like the one suggested by Beer is attained in two ways. Firstly, the viability is attained through each component being responsible for interacting with its own part of environment (although the parts that fall into responsibility of different components can partially intersect). This ensures fast (non-bureaucratic) response to fluctuations and changes. Secondly, it is attained through the recursive decomposition of components so that each of them has a structure of a viable system in respect to its own part of the total system environment (such decomposition concerns the units of System 1, in the first place).

Though VSM does not belong to the mainstream models, it is used in practice by some management consultant, mainly for diagnosing various organizational problems. There are a number of management books, e.g. [6,7], that present methodology of

how to conduct such diagnostics and list so-called pathological archetypes (patterns) that quite often can be found in real life. In our work, we used the set of pathological archetypes suggested in [6] to diagnose a problem in the case.

3 Case study overview

In this section, we present a summary from the case study [8] that was used for building our VSM model and making diagnosis. The information was gathered mainly via interviewing the staff of the department Connected Car Services (CCS). In total 12 interviews were conducted with members of staff in various positions, e.g., managers, project leaders, developers. Some information came from observations in CCS office, and from studying internal documentation, e.g. the organizational chart.

Based on the information gathered in the study, the following issues were clarified:

1. Organizational structure of the company's car development – which departments are engaged.
2. How collaboration between CCS and other departments is currently organized.
3. What problems exist in current arrangements from the point of view of CCS.
Considering only the CCS view point is a limitation that follows from the limited time and resources available for the case study.

3.1 Departments and functions engaged in product development

Below we list and explain functions and departments directly engaged in new product development, see also Fig. 1 showing a simplified organizational structure of the part of the company engaged in the new product development:

1. Concept Development (CD) is a functional unit headed by CEO; it includes managers from the following department listed below: R&D, M&S, CS, ME. The main goal of the CD is to envision the next generations of the company's products (cars). Note that CD has no representative from CCS, and it does not interact directly with CCS.
2. Product Strategy & Line Management (PS) is a department responsible for planning and managing the development of the next generation of products. It works in cooperation with CD and plans and oversees the development of all car models from concept to production. The department sets requirements on the cars, functions and related products to be designed, developed and produced. PS functions as an internal customer in relation to the following departments: R&D, M&S, CS, ME. Note that PS does not interact directly with CCS.
3. Research and Development (R&D) is a department that designs new car models and modification to the existing ones. R&D communicates directly with CCS serving mostly as an internal customer for the latter. The relationship could be inverted as well, as providing an intelligent service may require changes in the car design, e.g. adding more sensors. In this case, CCS may send requirements to R&D.

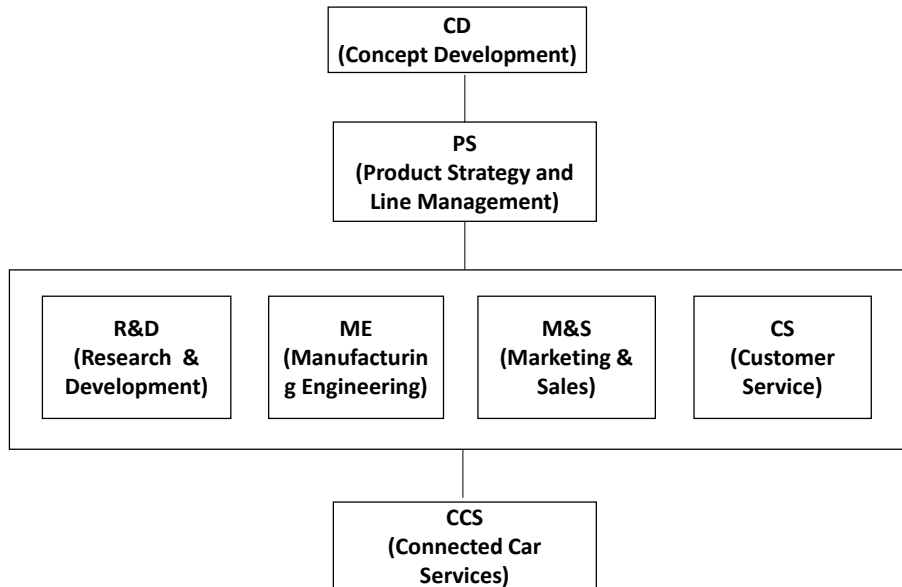


Figure 1. Departments involved in new product development presented in form of a simplified organizational structure.

4. Marketing & Sales (M&S) department that functions as an internal customer to CCS providing requirements and change requests for intelligent services that ensure the car connectivity, e.g. between the VC (Vehicle Corporation) and a car, its owner, the service person that served the car or the retailer who is selling/sold the car.
5. Customer Service (CS) is a department that functions as a company's product owner after the car has been sold, and provides services to the cars owners. CS analyzes input from customers collected by the systems connected to the Current Model Quality department, see below. CS communicates directly with CCS serving as an internal customer for the latter; it defines requirements on what kind of data the CCS systems should provide to the service.
6. Manufacturing Engineering (ME) is a global function with responsibility to launch new car programs and processes in VC joint ventures and to ensure an efficient production process with optimum balance of the production. A new car program will require interaction between ME and CCS. More precisely, ME sends requirements to CCS to deliver the CCS's solution to the production process in a certain way. CCS can also send requirements to ME to adapt the production process to facilitate incorporation of CCS's solutions.

Besides the above six functions/departments that directly participate in the new car production and services development, CCS also communicates with the following functions/departments that are indirectly engaged in the process:

7. Current Model Quality (CMQ) is a department that collects data/information from cars already on the roads. The data/information is provided by service staff, dealers (retailers) and intelligent systems installed in the cars. Information from CMQ is analyzed by both CCS and CS. Based on the information, a request for solutions on customer-related problems can be sent from both CMQ and CS. If the change required concern other departments, the data/information from CMQ is also sent to PS.
8. Supplies/Subcontractors (worldwide) are the companies that deliver third-party software or develop software on behalf of CCS (outsourcing).

3.2 Collaboration

In summary, the process of development of a new car model from the point of view of CCS goes as follows:

1. CD (Concept Development), which includes representatives from R&D, M&S, CS, ME, together with PS decides on starting development of a new car model.
2. PS (Product Strategy and Line Management) works out general requirements and an overall plan which are sent to R&D, M&S, CS, ME for execution.
3. R&D, M&S, CS, ME make more detailed plans and execute them under the leading of R&D.
4. When needed R&D, M&S, CS, ME send requirements and overall plans to CCS to plan and provide smart software for the new model. These requirements and plans are analyzed and discussed. Feedback is sent to the respective department. The feedback can include requirements on changing the physical design of the new model, production processes or services planned to be delivered with the new model.
5. After the requirements are finalized, CCS starts a number of subprojects to deliver solutions. They are completed by autonomous teams that often include contracted consultants or subcontractors worldwide. A team could use any project methodology they choose, but it needs to deliver solutions according to the plan. In case of problems to address in time, these problems need to be discussed and solved as early as possible, for example by adding resources to the team in question.

The projects in the company are usually driven by a plan based Work Breakdown Systems (WBS) method using so-called gates that are checkpoints in the process where the deliveries from different departments are evaluated. If the deliveries are accepted, the gates will open, and the next steps in the project can be carried out. Though CCS is free to choose a different approach to project management, they need to abide to WBS when integrating their solutions with the components produced by other departments.

3.3 Issues in collaboration arrangements

During the interviews conducted in the frame of our case study, the following groups of issues that may hamper collaboration have been discovered:

1. Issues related to the position of CCS in the VC organization
2. Issues related to the coordination between the departments
3. Issues related to the internal way of working within CCS

The main issue related to group 1 is expressed in statements from CCS employees that they become involved in the new model development too late, after the major decision has been taken. The latter is confirmed by the description of collaboration in the frame of new development from Sections 3.1 and 3.2. In addition, other departments, and R&D in particular, do not understand the nature of work completed by CCS and the dynamics of technological development in the area of CCS. This partly explains why R&D that heads the new development projects, by itself, does not come to the insight that CCS needs to be involved from the early stages of the project.

The main issue related to group 2 is the difference in the project management methodology used by R&D, which plans the whole project, and the methodology used by CCS. The former uses a plan-driven WBS, while the latter uses agile methods. This misalignment results in the following type of statements from CCS employees: “The budget process for managing the running cost is too strong, too governing for the development work, i.e., it hinders/obstructs the flexibility needed for the development work.”; “It is hard to work with Scrum within CCS and other departments because of necessary deadlines in a large organization. A project only allows a deviate of 10% + - in time and money otherwise the project is considered as a failure.”

The third group of issues is connected to that CCS, so far, has not achieved the high level of maturity in understanding and using the agile approach to software development. This group of issues is outside the scope of this paper and it will not be considered in the rest of the paper.

4 Using VSM for diagnosing misalignment

4.1 Building a model

According to the VSM hypothesis, a viable system, i.e. a system capable to adapt to changing environment, should consists of semiautonomous operational units (System 1) that collaborate (System 2) under the common control (System 3), while this structure can be changed under the governance of higher level systems (System 4 – Intelligence/Future, and System 5 – Identity management). Furthermore, each unit of System 1, on its own, can be represented as a viable system.

To build a VSM model for the whole VC was not included in our task, as we were focused only on diagnosing the problems in the part of VC that concerns new product development. New product development is considered as belonging to System 4 (Intelligence/Future), which, usually, is not subjected to recursive decomposition.

However, in the current state of the automobile industry, and other manufacturing industries as well, the role of new product development is changing against the role of manufacturing. The role of the former becomes more important for winning the competition and bringing revenues than the role of the latter. Therefore, it is possible to consider the new product development as a viable system on its own¹, which we do in this section.

Based on the information obtained during the case study, we have built a simplified VSM model of the new product development at VC presented in Fig. 2. In this model, we consider System 1 as including, in the first place, the following three organizational entities:

1. R&D – Research and Development
2. CCS - Connected Car Services
3. ME - Manufacturing Engineering

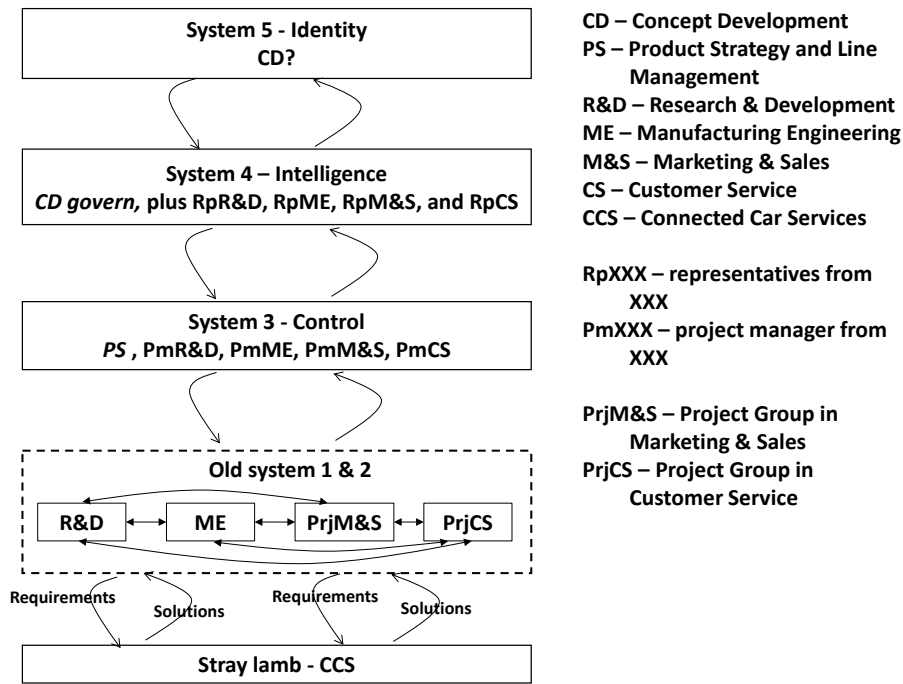


Figure 2. VSM for new product development and involved units

Besides these System 1 units, project groups from M&S (Marketing and Sales) and CS (Customer service) could be involved in the development related to the systems that connect the car to VC, or provide information from the car to Customer service (CS). These project groups are also considered as System 1 units in our model. In Fig. 2, we denote these units as PrjM&S and PrjCS.

¹ The idea was suggested by Patrick Hoverstadt in a private communication.

Regarding System 2, our case study does not provide full information on how it works, except the part that concerns coordination between CCS and other System 1 units. In this respect the coordination is mostly based on the customer–supplier type of relationships, where other units pass on requirements on the software to be produced and expect getting a solution on time and budget, which is depicted in Fig. 2. The latter does not exclude the reverse relationships when CCS, in turn, can send requirements to the other units, but this happens rarely.

As far as System 3 (Control or Delivery Management) is concerned, based on the information from the case study, it can be considered as consisting of PS (Product Strategy & Line Management), which creates overall plans, and Project managers from R&D, denoted as PmR&D, who oversees the whole development. We assume also that System 3 may include Project Managers from other units, except CCS. These are denoted as PmME, PmM&S and PmCS in Fig. 2.

It is reasonable to assume that System 4 (Intelligence/Future) coincide with CD (Concept Development) which includes CEO and representatives from R&D, M&S, CS and ME, denoted as RpX in Fig. 2. Note that no representatives from CCS are included in CD, and thus in System 4.

As far as System 5 is concerned, the case study does not provide enough information on Identity Management. For the sake of completeness, we, tentatively, assume that CD is also responsible for Identity Management, which is denoted in Fig. 2 (the question mark represent tentativeness of the assumption).

4.2 Diagnosing a problem

As has been discussed in Section 2.3, one of the main principles of achieving viability is having semi-autonomous units, each of them adjusting to the changes in the part of the environment with which the unit interacts. As the result, the system achieves viability/adaptability without much overhead and bureaucracy of the central command and control. In the theoretical terms, the whole system achieves the variety equal of multiplication of varieties of its System 1 units, and thus can achieve requisite variety to cope with the whole environment in which it operates.

Three of the System 1 units in Fig. 2 deal with technical development - R&D, ME and CCS. These units need to follow the technological development in their respective fields and use new technologies in a prompt fashion in their parts of product development. The technological fields related to these different units substantially differ. R&D is engaged in technology that is related to vehicles, like new type of energy/engines to be used. ME concerns are with manufacturing technologies, e.g., robotics. CCS is related to the modern ICT technology. While all these fields are quite dynamic, the speed of changes in them is different. In particular, the speed of technological development in ICT, with which CCS is dealing, is very high compared with the field related to R&D.

In addition to the difference in speed of technological development, there is the difference in the products/modules produced by the units. CCS products are highly interactive, and interaction includes both interaction with other devices (e.g. other cars), and interaction with human beings (e.g. owners or salesperson). Both, the high

speed of technological development in the field and the interactive nature of the products/modules developed makes it hard or impossible to define all requirements before the development begins. This is a known factor in ICT, and it is a major reason behind the rise of agile methods in software development. The agile methods became popular due to the need to mitigate the impossibility to get all requirements at the beginning of the development. CCS just follows the trend in trying to embrace agile methods.

At the same time R&D and ME continue to use a traditional plan-based method, which suit their part of development quite well. As R&D manages most complex project that stretch over all System 1 units, it is no wonder that they impose the traditional method to be used in the whole project. The latter comes into the conflict with the method CCS is trying to establish internally. The conflict is on both on the technical side, and the social side. On the technical side combining two completely opposite project methodologies is difficult on its own. The conflict on the social side consists of the fact that people who have not been subjected to the environment which requires application of an agile method have difficulties to imagine why it is needed. This is especially true considering the long experience of plan-based projects in an engineering department.

Note that the conflict as above does not exist, or is less significant, in the projects where R&D and ME are not much involved. In the projects where M&S and CS served as internal customers, CCS was able to successfully use agile development.

The conflict described above is difficult to solve in the current structure of the product development organization. As we can see from Fig. 2, CCS are not on the same level as other units, as it is considered as internal vendor to other units, not as an equal partner in the development. In addition, CCS has no representatives in System 3, and 4, which makes it difficult to consider the special needs of CCS when envisioning a new product and planning its development. With such a structure, it is difficult to find a right compromise between the plan-based and agile methodologies that is needed for successful development of intelligent products and services.

From the diagnostic point of view, the current situation at VC product development can be characterized by a pathological archetype called *stray lamb* [6]. The *stray lamb* is a pathological archetype where primary activities, that is, activities that give direct value to the customer, are not part of the formal organizational structure, but, given the activities' importance, should be. A cure for the stray lamb diagnosis is relatively straightforward – to adjust the organizational structure to the new reality on the ground. This transition can be initiated by introducing representatives of CCS in System 3 and System 4 of the model in Fig. 2.

5 Conclusion and lessons learned

As was stated in the introduction, the focus of our investigation is on misalignment in the socio-technical structure of an enterprise that may occur when transitioning from manufacturing “dumb” products to delivering “intelligent” product and services. This type of misalignment is relatively new, though it has, at some extent, been reported in

the literature. Our investigation was carried out as a case study at a car manufacturer where this kind of misalignment was detected.

To understand the causes of misalignment, i.e. making a diagnosis, we built a VSM model of the new product development in the organization that we investigated. Based on this model, we were able to make a diagnosis using a pathological archetypes suggested in [6]. More exactly, we could establish the presence of a so-called “stray lamb”.

The following lessons have been learned during the project:

1. A misalignment that may occur when transitioning from manufacturing “dumb” products to delivering “intelligent” products and services concerns both technical and social parts of the socio-technical structure of the enterprise. The first one concerns conflicts between the project management methodologies, the second one concerns the culture of people – their ways of thinking and doing – accustomed to these methodologies.
2. The misalignment could be diagnosed, i.e. causes of misalignment could be pinpointed by using VSM and a set of pathological archetypes suggested in [6].
3. Though product development belongs to System 4 of the VSM of the whole enterprise and not to System 1, it still can be recursively decomposed and represented as VSM subsystem on its own. This runs against the classical understanding of VSM [1], where only units of System 1 can be decomposed. The explanation here is that new product development in a modern enterprise is where the real value for the customers is created, which demands this business activity to be as viable as the units of System 1.
4. To make a diagnosis of the type we made, there is no need to develop a very detailed model of the VSM type; a high-level model of the kind presented in Fig. 2 is sufficient. Note that the model was built based on the investigation that concerned only the CCS department. Thus, some parts of the model remained unclear, e.g. who was responsible for System 5. However, this limitation did not hinder us to create a model sufficient for making the diagnosis.

Though the *stray lamb* archetype, so far, has been discovered only in one case of the transitioning to “intelligent” products and services, it is reasonable to assume that this archetype may also be present in other such cases. However, establishing the frequency of “stray lamb” in similar circumstances needs additional investigation. Furthermore, our investigation leads to another open research question/problem: “How to integrate a traditional plan-based project management with an agile one?”. Note that this problem becomes the next one to solve after the suggestion to adjust the organizational structure to the new reality has been implemented. A starting point for such investigation could be to study existing hybrid project approaches that aims to combine agile and traditional waterfall approach, such as PRINCE2 Agile [17].

Acknowledgements: Many thanks to all people participated in the case study that led to building the VSM model presented on this paper, especially, to Johan Isaksson and Isac Antblad. The authors are also grateful to Patrick Hoverstadt for his explanation on recursion in VSM in a private email communication.

References

1. Beer, S.: *The Heart of Enterprise*. Wiley, Chichester (1979)
2. McFarlane, D., Giannikas, V., Wong, A., Harrison, M.: Product intelligence in industrial control: Theory and practice. *Annual Reviews in Control* 37(1), 69-98 (2013)
3. Agile Alliance: *Manifesto for Agile Software Development*. (Accessed 2001) Available at: <http://agilemanifesto.org/>
4. Shore, J., Warden, S.: *The art of agile*. O'Reilly (2008)
5. Bider, I.: Analysis of Agile Software Development from the Knowledge Transformation Perspective. In Johansson, B., ed. : 13th International Conference on Perspectives in Business Informatics Research (BIR 2014), Lund, Sweden, pp.143-157 (2014)
6. Hoverstadt, P.: *The Fractal Organization: Creating Sustainable Organization with the Viable System Model*. John Wiley & Son (2008)
7. Pérez Ríos, J.: *Design and Diagnosis for Sustainable Organizations. The Viable System Method*. Springer (2012)
8. Olsson, S.: *Governing the complexity of interactions when a new IT department is introduced within a traditional engineering organization*. MS Thesis, DSV, Stockholm University, Stockholm (2015)
9. Porter, M., Heppelmann, J.: How smart, connected products are transforming competition. *Harvard Business Review* 92(11), 11-64 (2014)
10. Yang, X., Moore, P., Chong, S.: Intelligent products: From lifecycle data acquisition to enabling product-related services. *Computer in Industry* 60(3), 184-194 (2009)
11. Broy, M.: In : *Proceedings of the 28th International Conference on Software Engineering (ICSE'06)*, pp.33-42 (2006)
12. Grimm, K.: Software technology in an automotive company: major challenges. In : *Proceedings of the 25th international conference on Software Engineering (ICSE'03)*, pp.498-503 (2003)
13. Heinecke, H.: Automotive system design-challenges and potential. In : *Design, Automation and Test in Europe*, vol. 1, pp.656 - 657 (2005)
14. Martini, A., Pareto, L., Bosch, J.: Communication factors for speed and reuse in large-scale agile software development. In : *Proceedings of 17th International Software Product Line Conference (SPLC '13)*, New York, NY, USA, pp.42-51 (2013)
15. Espejo, R., Reyes, A.: *Organizational Systems: Managing Complexity with the Viable System Model*. Springer (2011)
16. Bider, I.: *Towards Process Improvement for Case Management. An Outline Based on Viable System Model and an Example of Organizing Scientific Events*. In : *forthcoming - BPM workshops 2015*, Springer, LNBIP (2016)
17. AXELOS: *Prince 2 Agile: An overview of PRINCE 2 Agile*. Available at: http://prince2agile.wiki/An_Overview_of_PRINCE2_Agile