Information and Software Technology 67 (2015) 44–64

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Requirements communication and balancing in large-scale software-intensive product development

J. Pernstål [a,*], T. Gorschek [b], R. Feldt [b], D. Florén [a]

[a] Volvo Car Corporation, SE-405 31 Göteborg, Sweden
[b] Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden

ARTICLE INFO

Article history:
Received 5 February 2015
Received in revised form 17 June 2015
Accepted 18 June 2015
Available online 25 June 2015

Keywords:
Organizational management and coordination
Requirements engineering
Software engineering
Automotive industry
Case study
Lean and agile product development

ABSTRACT

Context: Several industries developing products on a large-scale are facing major challenges as their products are becoming more and more software-intensive. Whereas software was once considered a detail to be bundled, it has since become an intricate and interdependent part of most products. The advancement of software increases the uncertainty and the interdependencies between development tasks and artifacts. A key success factor is good requirements engineering (RE), and in particular, the challenges of effectively and efficiently coordinating and communicating requirements.
Objective: In this work we present a lightweight RE framework and demonstrate and evaluate its industrial applicability in response to the needs of a Swedish automotive company for improving specific problems in inter-departmental requirements coordination and communication in large-scale development of software-intensive systems.
Method: A case study approach and a dynamic validation were used to develop and evaluate the framework in close collaboration with our industrial partner, involving three real-life cases in an ongoing car project. Experience and feedback were collected through observations when applying the framework and from 10 senior industry professionals in a questionnaire and in-depth follow-up interviews.
Results: The experience and feedback about using the framework revealed that it is relevant and applicable for the industry as well as a useful and efficient way to resolve real problems in coordinating and communicating requirements identified at the case company. However, other concerns, such as accessibility to necessary resources and competences in the early development phases, were identified when using the method, which allowed for earlier pre-emptive action to be taken.
Conclusion: Overall, the experience from using the framework and the positive feedback from industry professionals indicated a feasible framework that is applicable in the industry for improving problems related to coordination and communication of requirements. Based on the promising results, our industrial partner has decided upon further validations of the framework in a large-scale pilot program.

1. Introduction

Software is rapidly becoming a substantial component that is widely seen as the main driver and source of innovations in a number of traditionally hardware-focused industries [1–3]. In the automotive domain, the value of automotive software is expected to grow from 127 billion Euros in 2002 to 316 billion in 2015 [4].

With the relative contribution and size of software, the entire complexity of developing large-scale software-intensive systems escalates, introducing new challenges among development organizations [3,5]. Requirements engineering (RE) is commonly identified as crucial and plays a vital role in the potential success of such development efforts [3,6,7]. In particular, the effectiveness of both formal and informal coordination and communication of requirements across organizational boundaries is critical [8,9]. This was also identified as a core challenge at the Volvo Car Corporation (VCC), a developer of luxury cars based in Sweden, as a part of their continuous improvement programs. The dramatic increase of interacting software-intensive automotive systems had increased the interdependencies among the requirements on functions, systems, and components, leading to more complicated organizational dependencies and needs across a multitude of different engineering disciplines and departments (inter-departmental).

There is a range of different solutions addressing requirements communication and coordination. The importance of reducing interdependencies through modularization [10,11], and adopting

* Corresponding author.
E-mail addresses: jpernsta@volvocars.com (J. Pernstål), tony.gorschek@bth.se (T. Gorschek), robert.feldt@bth.se (R. Feldt), dfloren@volvocars.com (D. Florén).

http://dx.doi.org/10.1016/j.infsof.2015.06.007
0950-5849/

techniques for enhancing requirements specification quality have been emphasized, but the increased complexity has made these solutions less applicable [3,6]. In addition, using standards as a coordination mechanism is a key to manage interdependencies in development work [12,13]. For this, software process improvement (SPI) frameworks, such as CMMI [14], ISO/IEC 15504 (a.k.a SPICE) [15], are central to software-intensive systems development. However, they provide little guidance on what the actual implementation should look like, and adopt a one-sizefits-all view without considering the specific needs of organizations and projects [6,16,17].

Solutions to intensify organizational communication, building on team-based methods [18,19], visual management [20,21], goal-oriented approaches [22], and recent agile methods (e.g., Scrum [23]) have also been suggested. However, to the best of our knowledge, it is unclear whether these approaches can contest with the complexity of the large-scale development of software-intensive systems, taking inter-departmental and multidisciplinary aspects into account [24–26].

In this paper, we present a flexible and lightweight RE framework for improving requirements communication and coordination in the design and development process, called Balancing Requirements and Solution Space (BRASS). BRASS was developed in close collaboration with VCC in order to resolve the company's problems identified in the process of balancing requirements and solutions as reported in [27]. The main problems concerned the insufficient quality of requirements and solutions specifications and the lack of communication in the early phases of the development cycle. The importance of solving these problems at VCC is high as they are causing costly loop backs in the final part of the project and inefficient solutions. The cost for this over the life cycle of a car model, which is typically seven years [3], could be estimated at roughly 10MUS$ [27]. Using three real cases, we demonstrate how BRASS was applied in practice and evaluate its usefulness as perceived by professionals at VCC.

We use the term balancing in BRASS, as it is established in the RE-process and is well understood in its pragmatic use at VCC (e.g., [28]). The notion of term balancing is broad, including a combination of the following main coordination activities: (1) creating and breaking down different stakeholders' needs (e.g., business objectives, strategies, and requirements) and identifying the necessity of balancing the needs, (2) analyzing and trading-off the stakeholders' needs and specifying understandable and agreed requirements and solutions, and (3) establishing implementation plans and validating how the actual solutions work. The overall goal of the balancing process is to identify and level out the deviations in the stakeholders' needs and deliver specifications and schedule and report validation tasks for evaluating agreed requirements and solutions developed.

Referring to existing literature on RE, the balancing process at VCC covers all parts of the RE-process (elicitation, analysis and negotiation, specification, validation and verification and requirements management) [29,30]. This means that BRASS does not focus on solving problems in a particular part of the RE-process, such as requirements negotiation, but rather takes a comprehensive view from an RE perspective.

The concept of alignment can also be related to the term balancing and has been conceptualized in various ways in different research fields. In information technology (IT) literature, alignment is, for example, defined as the degree to which the mission, objectives, and plans contained in the business strategy are shared and supported by the IT strategy (e.g., [31]). In software development, technologies focusing on the alignment of activities within and between different development phases, for example, between software architecture and implementation (e.g., [32]) and software requirements and testing [33], have been suggested. Even though the terms alignment and balancing have their similarities, we believe that using alignment can be more confusing due to its deviating contextual interpretations and VCCs unawareness of it.

The remainder of the paper is structured as follows. Section 2 presents the research context, the case company, and elaborates on the challenges identified at VCC, and related work. Section 3 describes BRASS and describes how it can be tailored and applied to the cases studied at VCC. The evaluation of BRASS through validation and the results from using it, as well as its limitations, are presented in Section 4. Section 5 presents the conclusions and Section 6 outlines the future work.

2. Background and related work

This section describes VCC and its core challenges and the specific problems identified and addressed in this paper. We also present related work with regard to these challenges.

2.1. Case description

VCC is a premium car manufacturing company and has approximately 22,000 employees all over the world and produces roughly 450,000 cars per year (2011) [34]. New cars are developed in large-scale projects, involving several 100 of person years and billions of dollars. VCC is organized as a matrix organization and uses a traditional plan-based approach characterized by planning everything from the start of the vehicle program and focusing on documentation. For this, VCC uses a stage-gate model [35] with its milestones to govern the development of the complete car and the V-model [36] to present an overview of design and validation and verification of the complete vehicle and inherent functions, systems, and components. The V-model cycle can be iterated several times during the complete vehicle development. Furthermore, the requirements are mainly specified in written text and administrated by computer-aided RE tools (see also Section 3.1.1). Fig. 1 maps the overall phases in the stage-gate model for developing complete vehicles along with the three main coordination activities of the balancing process and the requirements and design levels in the V-model to a generic product development model developed by Peters et al. [37]. This setting is commonly used in the automotive industry [6,20,38]. The main focus of this study is to resolve problems in the balancing process covering the generic phases pre-design development and design and development process.

The breakdown of product requirements follows a top-down process starting with overall business and user requirements (e.g., product strategies, legal demands on reduced gas consumption, and increased safety) derived from the business development process and ending in component requirements on both hardware and software solutions (e.g., engine control software and engine hardware parts like actuators and sensors), via complete vehicle attributes and function/systems requirements. The annual work typically consists of assessing business and customer requirements and refining them to the vehicle attributes and developing functions that accommodate the required attributes. The primary task in the subsequent pre-program study involves the balancing of functions that have been developed with underlying architectural constraints based on the complete vehicle design. During the concept phase, identified problems in the needs of stakeholders (e.g., function and systems owners, and manufacturing engineers) and system solutions are developed, assessed and selected. The prerequisites for designing the systems' software and hardware components are then compiled and finalized in specifications such as software requirement specifications and design prerequisites for hardware. In addition, verification plans and the need of both

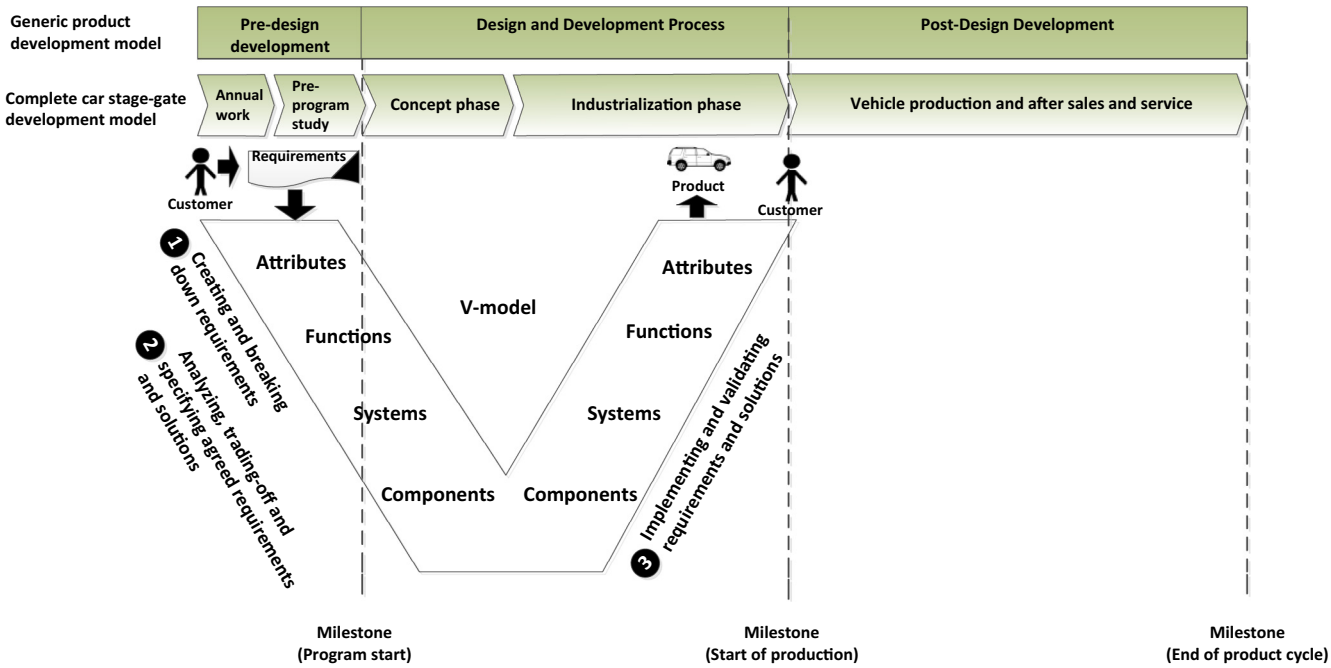

Fig. 1. Mapping of a generic model for product development (adapted from Peters et al. [37]), the overall complete car development stage-gate model, and the V-model.

virtual and physical test objects are elaborated. The next phase, industrialization, focuses on developing, integrating and verifying the design of hardware and software solutions based on the specifications delivered in the previous concept phase. Building models in the form of digital-mock ups constitutes a major part of the verification of hardware designs while techniques, such as Unified Modeling Language (UML) [39], Matlab/Simulink and Hardware In the Loop (HIL), are used to model and validate software designs.

The automotive companies face various challenges of which one is to effectively manage the enormous amount of product documentation. Typically, luxury cars are built out of over 300 systems and 2500 functions, produced in a large amount of variants [3]. In order to design and integrate these systems and functions into a complete car and in order to manage the variants, a large amount of product documentation, such as requirements (about 100,000 textual requirements in VCC), specifications and standards must be administrated [6]. An example of this is the tailgate system in a vehicle. User and functional requirements are allocated to the tailgate system generating systems requirements, which are broken down into requirements for mechanical components (e.g., tailgate latch and hinges), electrical components (e.g., lock motors, sensors and cable harnesses) and software components controlling several functions (e.g., locking, maneuvering and remote function logics). In addition, manufacturing requirements must be considered in order to secure that the tailgate system fits the manufacturing processes—how shall, for example, the tailgate be configured (e.g., downloading software to electrical control units for controlling the tailgate locking and maneuvering) and quality assured in manufacturing? Furthermore, the documentation for the tailgate system must be maintained and updated during the lifecycle of the car [3,6]. This causes large overhead costs and, for example, it is estimated that if the number of variants is doubled, the overhead costs rise between 20 and 30 percent because of the increased complexity [5].

Another challenge is to manage the high complexity of their organizational structure while the uncertainty and interdependency of development tasks are increasing. Usually, to attain economy of scale, the development work is organized in complex matrix systems with a dual-hierarchical form where development tasks are performed in several project teams, which are staffed with hundreds of people from several parts of the functional organization. To minimize the time-to-market, the complexity is elevated as the development process is accelerated by performing development tasks concurrently (e.g., engineering the tailgate system and the affected manufacturing solutions) [20,21]. In addition, the substantial growth of software increases the interdependencies between the requirements and development tasks and their uncertainty, leading to much larger requirements and solution space that must be managed [3,5,6]. The challenge of uncertainty and interdependency in development work is not unique to the automotive industry as it has been recognized in the development of other large software-intensive systems [7–9], but also in general product development [12,13,40].

2.2. Background and motivation

The study presented in this paper is part of a software process improvement (SPI) initiative focusing on the inter-departmental interaction between Manufacturing (MAN) and Product development (PD) in the development of software-intensive automotive systems. PD is concerned with the design and development of software-intensive automotive systems (e.g., the development of power train and chassis control systems for vehicles). MAN is concerned with managing these systems when producing vehicles (e.g., vehicle manufacturing operations affected by power train and chassis control systems).

The results of the process assessment and improvement planning showed that a major problem revolved around requirements communication, and in particular, the balancing of requirements and solutions [27]. The balancing process and its three main coordination activities and the current situation of how this process is performed between PD and MAN are illustrated in Fig. 2 (unbroken line). It starts with an initial and broad set of requirements and possible solutions ("Provisional requirements and solution space" in Fig. 2), which should converge continuously by refining requirements and narrowing the set of alternatives to a superior solution ("Final selection" in Fig. 2). The main goal of the balancing process is to agree and decide upon most of the requirements and solutions

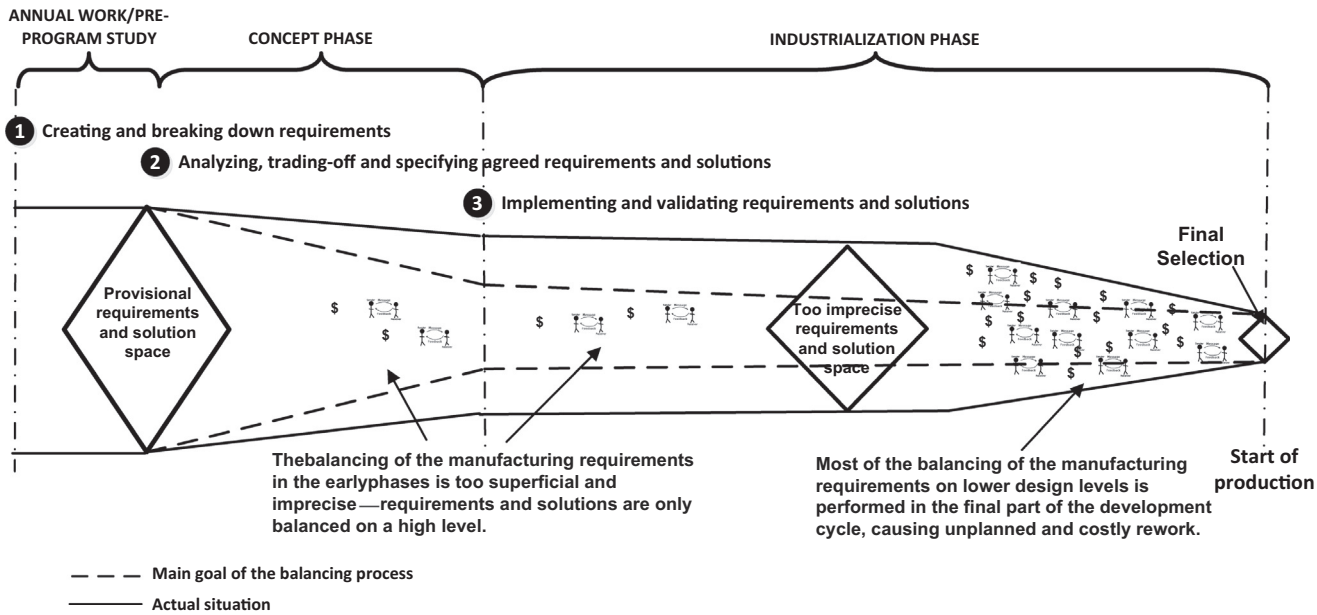

Fig. 2. Overview of the requirements and solution balancing problems between PD and MAN at VCC.

in the early phases of the development cycle (dashed line in Fig. 2), setting the base for a further breakdown, analysis and trade-off, and validation of requirements as well as solutions for lower design levels (e.g., electrical and software components). In general, this goal can be better achieved for hardware-based systems as these are relatively stable, especially in the final stages of development, than for software-intensive systems since software is characterized by its being highly responsive to changes with short and frequent iterations throughout all development phases.

Typically, during the concept phase, MAN initiates the balancing through specifying and communicating generic and high-level (attribute level, see Section 3.1.1) manufacturing requirements dealing with, for example, the calibration of car functions in manufacturing, to PD. The requirements are then broken down in order to identify balancing needs, and the balancing continues among lower design levels (e.g., the specific function, sub-systems and components affected by the manufacturing requirements on calibration). However, the balancing is only performed on high design levels (e.g., electrical architecture and base functions), causing a lack of shared understanding of the requirements and detailed specifications, providing insufficient information about how agreed solutions and implementations should actually look like (e.g., calibrating a specific car function).

Even though the balancing process descriptions at VCC prescribe that a major part of the balancing should be performed proactively in the early phases, there is a lack of both formal and informal means for supporting and coordinating this. Much of the balancing is instead primarily done on an ad-hoc basis without systematic guidance on best practices. In Pernstål et al. [27], we showed that this caused significant costs for managing late defects and changes and jeopardizing the manufacturing performance at the worst possible time, when the market demand is peaking after the launch of a new product. Thus, it is important to improve the balancing process in order to resolve the current situation, but it is even more important to manage larger requirements and solution spaces and more frequent changes due to the increasing amount of software-intensive automotive systems in future vehicles (see Section 2.1).

VCC expressed a direct need for developing a method for the specific problem of balancing requirements and solutions between PD and MAN. One alternative that was also discussed was changing the established and formalized RE tools and processes. However, this would have caused costly changes in tools and increased complexity of the processes without any guarantees of improved performance. BRASS was designed and tailored based on the needs and experience of line and program managers and engineers in PD and MAN at VCC. Furthermore, in order to clarify and enhance the practical use of BRASS, examples of real balancing issues in an ongoing new car model project were extracted and used during the creation, evolution and validation of BRASS.

2.3. Related work

Several approaches have been proposed to resolve requirements communication by either reducing interdependencies, using techniques for enhancing requirements specification quality, standards prescribing best practices, or intensifying organizational communication.

To reduce the interdependencies between requirements and development tasks, the idea of modularization has been promoted in systems engineering [10] and software engineering (SE) [11]. It is a useful approach for dividing the development of complex products into independent and manageable development and manufacturing tasks. However, the growth of software in many products (e.g., vehicles) has led to previously independent functions, sub-systems and components having the need to interact with each other to an extent not previously seen [3,5]. This has lessened the possibilities of modularizing.

2.3.1. Requirements specifications quality

The requirements specification quality is a critical success factor in development projects. Requirements can be documented in several forms, such as use-cases, requirements modeling [41,42], and formal specifications [43]. Specifications in natural language are the most common. However, in practice, specifying precise and understandable requirements for large and complex systems is impossible to achieve [44]. For example, Weber and Weisbrod [6] found that automotive development is too complex to be managed by just textual requirements. In addition, manufacturing requirements are experienced-based and tacit rather than being detailed specifications of purposes and goals, and clearly describe expected results [28,45].

Techniques for modeling and validating requirements have also been proposed (e.g., Unified Modeling Language (UML) [46], Matlab/Simulink and Hardware In the Loop (HIL). However, model-based development and the testing of software-intensive automotive systems are in their infancy. For example, Broy et al. [3] point out that, because of the lack of a formalized modeling language, modeling is only applied to certain steps in development work. Insufficient integration possibilities, such as linking engineering data to models and compatibility between different tools, also need to be resolved. Furthermore, the cost of improving the specification quality of complex systems through detailing textual requirements or modeling is likely to be high, and must be evaluated in relation to the benefits [3,5].

2.3.2. Standardization

Adopting effective techniques and standards (e.g., IEEE std. 830-1998 [47]) that help produce clear and precise requirements (e.g., correct, unambiguous, complete, consistent, and verifiable) is considered to be crucial. Standardization can also be seen as a key coordination mechanism, enabling organizations to deal with interdependencies in development work [12,13,20]. To reach a more mature RE process there are several well-known SPI frameworks, such as CMMI and ISO/IEC 15504 [48,49]. The CMMI reference model comprises 22 key process areas (KPA) while ISO/IEC 15504 relies on external sources (e.g., ISO standards for systems lifecycle processes (ISO/IEC 15288:2008 [50]) and software life cycle processes (ISO/IEC 12207:1997 [51]). These frameworks provide a high level guidance regarding what to do, but do not detail how the actual implementation should proceed and no special consideration is given to specific organizational needs [16,17]. For example, the KPA of requirements development in CMMI only prescribe using proven models and performing a risk analysis when analyzing requirements to balance stakeholder needs and constraints. Furthermore, these standards look too narrowly into engineering aspects within a single project, which is no guarantee for a successful product as a project measure (e.g., level of requirements fulfillment) is only the first level to consider in RE [52].

However, when developing and implementing new technologies, they should be adaptable and integrated with established and standardized tools and ways of working, and should also be easy to use [6,20]. Otherwise there is a significant risk that users would reject the technologies as well as the corresponding process improvements.

As mentioned above, there are also standards and guidelines for specifying requirements. In addition, there are standards for the requirements and design of software-intensive automotive systems, such as the Automotive Open System Architecture (Autosar) [53] and ISO 14229-4: 2012 [54]. Even though more standards are used, they allow variability and it is estimated that 90 percent of the current software must be changed from one generation of vehicles to the next, while only 10% of other systems, e.g., exterior (doors) and interior (seats), are changed [3].

2.3.3. Organizational communication

Many lean principles and practices originate from Lean Product Development (LPD), but recent agile methods have also had a strong influence on approaches that have been developed to reinforce organizational communication.

Establishing cross-functional teams, consisting of members representing different departments (e.g., MAN and PD) and roles (e.g., design and manufacturing engineers) is a core practice in LPD [20,55]. The purpose is to intensify the communication between the team members, rather than coordinating development tasks and departments, groups and individuals. Karlsson and Åhlström [55] found benefits in cross-functional teams, such as better product solutions and improved communication of project information to departments other than PD, but also difficulties in creating and maintaining them. A primary reason for this was that departments other than PD had difficulties in selecting and allocating the required resources to actively work with the projects, and in particular, in the beginning of the projects.

In order to elevate active cross-departmental development work, practices such as integrated problem-solving [19,21] and Set Based Concurrent Engineering (SBCE) [18] are used in LPD. Integrated problem solving and SBCE emphasizes the need of communication between upstream and downstream members of a development project (e.g., designers and manufacturing engineers). Both SBCE and integrated problem-solving have been successfully applied to primarily hardware development in the automotive domain—but it is unclear how the practices address the fact that software as an artifact, and SE as a discipline, are becoming a central component in the products developed.

Attaining simple visual communication through effective visual management is one of the most critical success factors in concurrent development [20,21]. Typically, when visualizing the state of the software project, boards divided into different areas with the aim of uncovering project abnormalities (e.g., time and defects) are often used, (e.g., [56–58]). In Peterson and Wohlin [58], for example, the number of requirements were measured and displayed, visualizing the undesired behavior of the development (e.g., bottlenecks).

Agile software development methods (e.g., Scrum [23]) promote coordination and communication of requirements through practices, such as daily Scrum, product backlogs, story cards, and screen mock-ups. There are only a few studies on agile RE. Ramesh et al. [59] found that obtaining intensive communication is the most important factor for successful agile RE. However, if there are communication breakdowns caused by, for example, the rapid turnover of personnel and growing complexity of the products, the minimization of design specifications can create problems, such as the inability to scale the software, evolve the application over time or induct new members into the development team. This is in line with Salvonen et al. [60], who found that highly skilled people, especially in requirements engineering, are a prerequisite when implementing the agile development of embedded systems.

Even though organizational communication is central in lean and agile methods, and they also influence more traditional industries, such as the automotive one, the transition to more lean and agile methods have only started and is not yet widespread in many large-scale software engineering contexts [24–26].

To the best of our knowledge, the only method explicitly aimed at intensifying requirements communication in software development is a pragmatic technique called Handshaking with implementation proposals, developed by Fricker et al. [22]. It organizes the requirements communication into a goal-seeking element (program manager) and a goal-implementation element (development team). The handshaking process consists of three main phases: (1) taking position, (2) negotiation and (3) confirming agreement. With promising results, the handshaking was applied to a large-scale industrial setting for improving inter-departmental requirements communication problems between marketing and PD (product managers and development teams). In this setting the roles of the goal seeker (product manager) and implementer (developer) are rather clear and static over time. However, in other settings, the roles the actors play may vary depending on factors such as the reason for the communication and what is being communicated, and can also shift during the development cycle [31]. Despite this, there is a need to obtain clarity of roles and responsibilities in the PD/MAN interface, as it is a key success factor [61,62].

To address the core challenges in the development of large-scale software-intensive systems (see Section 2.1) and the direct need for resolving the specific problems identified at VCC

Table 1
BRASS dimensions.

Dimension	Description
A. Communication	Specifies the processes and mediums used for controlling and coordinating the flow of information being communicated (e.g., face-to-face and formal meetings, e-mail, formal and informal information management systems)
B. Content	Specifies the required information being communicated for clarifying trade-offs between requirements and alternative solutions (e.g., requirements, functional specifications and models, descriptions of solutions and verification plans)
C. Actors	Specifies the key parties involved (e.g. individuals, roles, groups, teams, departments, projects, and organizations) and their responsibilities
D. Connections	Specifies the relationships between BRASS and its surroundings (e.g. product development processes, project information management systems, and standards)

(see Sections 2.2), we developed BRASS. BRASS primarily builds on the refinements of Handshaking with an improvement proposal [22]. Also, the lean-based concept of converging inter-departmental requirements and alternative solutions in SBCE [18] has influenced the design of BRASS.

3. The BRASS framework

This section gives an overview of BRASS and its generic parts and objectives, and a detailed description of how BRASS was tailored and applied at the case company. An evaluation of the industrial applicability of BRASS is given in Section 4.

BRASS is designed as a framework consisting of four generic dimensions, *communication (A), content (B), actor (C) and connection (D)*. These dimensions are representative abstractions of the objectives (see Table 2) and are aspects that always need to be described and considered in one specific application of BRASS. One reason for this is to achieve a broad applicability of BRASS for different balancing situations in organizations facing other types of business and settings. Another reason is that methods, being too rigid, are less efficient in development work than methods that can be adapted to the specific problem in question [63]. Table 1 briefly describes the dimensions, which are further detailed in Section 3.1, with an example.

BRASS has four main objectives, which are described in Table 2. BRASS evolved primarily based on the results of a process assessment presented in Pernstal et al. [27]. It reports on a communication model, called software communication and redundancy analysis model (SCREAM), developed for postmortem analysis of communication problems and applied to a set of 16 real communication events at VCC. The events represented different communication problems that had occurred in earlier projects, where the interplay between PD and MAN in the development of software-intensive automotive systems was crucial. In the study presented in this paper, BRASS is tailored to address the specific communication problems identified at VCC in [27]. However, BRASS is not dependent on the type of assessment approach used and can be applied to address different requirement coordination and communication problems in other contextual settings, see also Section 3.1. Furthermore, during the development of BRASS, the views of the company representatives were continuously considered.

3.1. Tailoring and applying BRASS at VCC

BRASS is a generic framework for improving requirement communication and coordination that needs to be tailored. In practice, before using BRASS, its dimensions must be adapted in order to fit the context and an organization's needs and prerequisites. For this, a tailoring process including three main steps is used as described in. Fig. 3 gives an overview of the prior process assessment and improvement planning and the tailoring process for BRASS conducted at VCC.

First, an initial tailoring of BRASS is conducted based on the results of a process assessment and improvement planning using SPI frameworks, such as iFLAP [65] or CMMI [14], or postmortem analysis as, for example, described in [27], and driven by representative cases exemplifying the specific balancing problems identified. For example, in order to capture and document relevant information being communicated, a number of properties were initially proposed and formulated (see Tables 5 and 6 in Section 3.1.3). In the second step, the initial tailoring is then introduced to parts of the organizations (e.g., pilot projects) for evaluation and refinement in order to provide decision support for moving to a full-scale implementation. For example, the properties used for information collection and documentation are added, split and reformulated. Third, when the tailoring of BRASS is implementation ready, it is released and fully deployed and continuously assessed and improved, and subsequently adapted to changes in the organization. The study presented in this paper covers the first step and parts of the second one, and the following details, and exemplifies how the tailoring was conducted at VCC.

Table 2
Brass objectives.

Objective	Description
A. Intensifying communication	BRASS should initiate and maintain the active involvement of the actors through intensifying and extending the analysis and negotiation of requirements and solutions during the design and development phase (see Sections 2.1 and 2.3.3 and [18,19,22,59])
B. Clarifying and specifying content	BRASS should be applicable to imperfect requirements (e.g., low quality or tacit and not documented) and have the capability of improving the understanding and quality of requirements by effectively clarifying and specifying the content being communicated (see Sections 2.3.1 and 2.3.3 and [6,22,28,44,45])
C. Clarifying actors involvement	BRASS should clarify the roles and responsibilities of the actors involved and their connections to the surroundings that influence the balancing (e.g., organizational affiliation, assigned requirements and work tasks in projects) (see Section 2.3.3 and [61,62])
D. Aligning connections	BRASS and its connections to the surroundings should be adapted and seamlessly integrated into established and standardized tools and processes in the organization and be lightweight (a simple process and unburdensome and clear documentation that is adaptable to specific organizational needs) in order to support engineers in their daily tasks (e.g., should not counteract the existing requirement management tools and processes but also work independently of them since tools and processes are likely to change over time) (see Section 2.3.2 and [6,20])

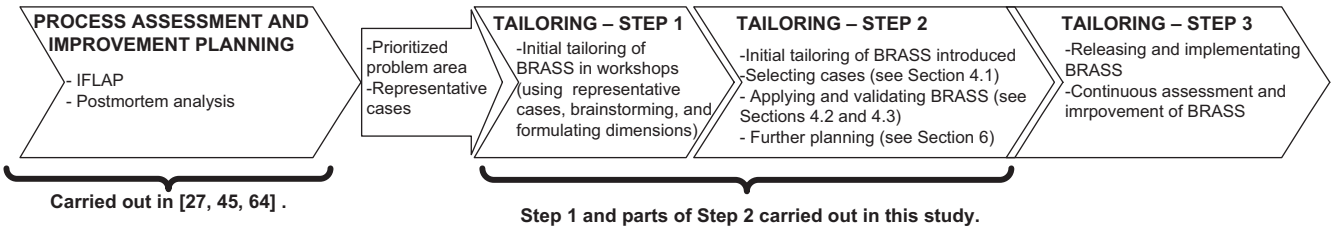

Fig. 3. Overview of the BRASS tailoring process at VCC. (See above-mentioned references for further information.)

The tailoring of BRASS was primarily example-driven by using relevant examples of communication problems identified in [27] and the three cases to which BRASS was applied. The cases were also selected in order to enhance the validity of the results by increasing the number of involved subjects the range of engineering disciplines and roles. All three cases involve real balancing issues between PD and MAN in a new car model project at VCC. Case 1 and 2 concern the high-level manufacturing requirements on calibrating car functions/systems in manufacturing. Case 3 concerns the balancing of new high-level manufacturing requirements for enabling operator messages to be displayed in the car. The examples, given throughout this paper, are based on Case 1. To preserve confidentiality, the cases are fictitious but based on actual cases from the company.

Fig. 4 gives an overview of how each of the dimensions of BRASS was tailored. For example, the dimension communication (see Fig. 4(A)) consists of an iterative three-step process and the mediums used are focused balancing meetings and face-to-face meetings. A brief description of the cases and their main results of the tailoring for each of the dimensions are shown in Table 3 (further details are not possible due to confidentiality concerns). The following describes and exemplifies each of the dimensions in detail.

3.1.1. Connection

In order to use BRASS effectively, it is necessary to identify its connections to the surroundings (see Fig. 4(D)), and how they affect the balancing, as well as clarifying how BRASS should operate and interact with them. In the context studied here, BRASS's connections to processes and tools used for developing and steering decisions on designs and manufacturing processes, and managing requirements are central.

Typically, the development of the car and the manufacturing processes is governed by an overarching stage-gate model [35] with milestones (MSs). Project management then reviews and confirms that the criteria are met for the current MS, demonstrates preparation for the next MS and updates the project prediction for final delivery and associated risks. Certain decisions need to be made at each MS by gathering the necessary information between the MS's. At VCC, a decision on the project start is made at MS1. Exploration and balancing of requirements and solutions, and the status of specifications, development, integration, and verification for systems and components are reported on different levels (e.g., systems and components, see below) at MS2 and MS3. Complete car prototypes are verified at MS4. Trial production and verification of the manufacturing processes starts at MS5, and manufacturing readiness and whether to ramp-up manufacturing are decided at MS6. The processes in BRASS should be tailored so that they comply with the prescribed activities between the MS's and the expected deliveries at each MS (see Section 3.1.2).

The central information source for developing software-intensive systems at VCC is textual requirements, which are supplemented with use-cases, formal specifications, and models. Requirements management tools, based on database systems, are used for specifying and administrating the requirements (e.g., trace links between requirements and related objects and support for generating specification documents). Siemens Teamcenter Systems Engineering and Requirements Management [66] is the primary RE-tool used at VCC. However, since such tools are likely to be changed over time, BRASS has been designed to be effective and useful irrespective of the tools used. To achieve traceability—from customer needs and product strategies to requirements detailing the design of hardware and software components—the requirements are structured into four main abstraction levels (see Table 4).

Based on these levels, the RE process adopts a formal goal-based approach where the breakdown of requirements and designs, and testing, follows a top-down process guided by the V-model [36]. However, the complexity of breaking down requirements is high because there are requirements on a multitude of interdependent functions, systems and components and many disparate sources of the requirements, spanning over multiple departments, groups, roles, and engineering disciplines. Furthermore, there are often already settled requirements on underlying levels that must be considered (e.g., architectural constraints), and the abstraction levels are not distinct since the requirements' quality and precision varies within the levels. For example, on the attribute level, the requirements on the car are specified at a high abstraction level (e.g., PDREQ1 in Table 4) while manufacturing requirements and other stakeholders' requirements are seen as generic and cross-cutting and therefore always structured on the attribute level despite their level of detail (e.g., MANREQ1 in Table 4). This affects the tailoring of work procedures in BRASS for breaking down requirements and designs in order to identify interdependencies and balancing needs across PD and MAN (see Step 1 in Section 3.1.2).

Interactions with other organizations, teams or roles within or outside the organization that have an impact on the balancing are also important. When developing software-intensive automotive system, parts of the software and hardware development are commonly outsourced to suppliers [3]. For example, in Case 1, a Tier 1 sub-contractor was responsible for the sub-system tailgate maneuvering and a Tier 2 sub-contractor developed the software component controlling the logics of the tailgate maneuvering (see Section 3.1.2).

3.1.2. Communication

The communication dimension includes a process for balancing the manufacturing requirements (see Fig. 4(A)). It covers the annual work, pre-program study, concept and industrialization phase of the complete car development and consists of main three steps as described in Fig. 5: (1) identifying balancing needs, (2) negotiating and specifying, and (3) validating. The process is iterative and continues throughout the duration of the complete car development accompanying reiterations of the V-model cycle. The process has two main goals: (1) extending and deepening the balancing of requirements in early phases, and leveling it out throughout the development cycle and (2) improving the precision of the content being balanced and specified in early phases—lower

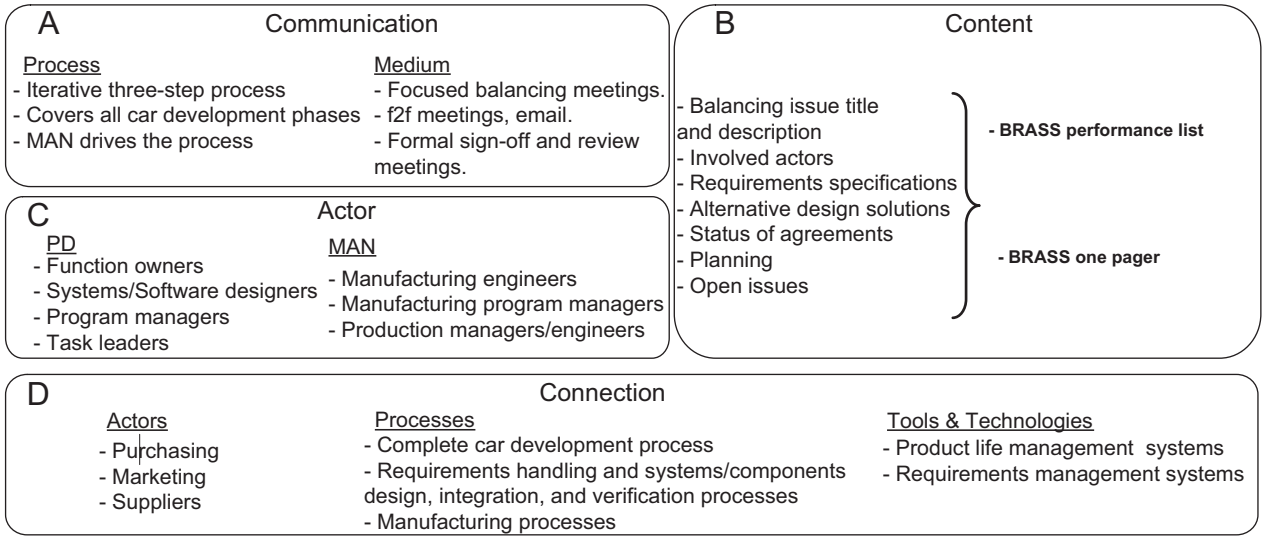

Fig. 4. Overview of the BRASS framework and its dimensions adapted for use at VCC.

Table 3
Main results of tailoring each case.

Dimension	**Case 1** Calibrating an exterior car function—automatic opening-and-closing of the tailgate in manufacturing. Mechanical, electrical and software engineering	**Case 2** Calibrating a chassis car function in manufacturing. Mechanical, electrical and software engineering	**Case 3** Displaying operator messages by using on board vehicle display. Electrical and software engineering
A. Connection	Tools: Siemens Teamcenter Systems Engineering and Requirements Management[65]; Elektra[a] Processes: Complete car development (see Section 3.1.1)	Tools: Siemens Teamcenter Systems Engineering and Requirements Management[65]; Elektra[a]; SystemWeaver [68] Processes: Complete car development (see Section 3.1.1)	Tools: Siemens Teamcenter Systems Engineering and Requirements Management[65]; Elektra[a] Processes: Complete car development; Manufacturing advanced engineering in annual work (see Section 3.1.1)
B. Communication	Initiated at MS1 (see Section 3.1.1); three focus group meetings; one formal system sign-off meeting	Initiated at MS1 (see Section 3.1.1); two focus group meetings; one formal system sign-off meeting	Initiated before MS1 (see Section 3.1.1); five focus group meetings; one formal function review meeting; one formal function sign-off meeting; one formal system sign-off meeting
C. Content	One manufacturing requirement; two product requirements; three alternative design solutions; one test specification	Six manufacturing requirements; one product requirement; three alternative design solutions; one test specification	Thirteen manufacturing requirements; two product requirements; three alternative design solutions; one function realization description
D. Actor	Three Manufacturing engineers; one Task leader; one Systems and software designer	Three Manufacturing engineers; one Task leader; two Systems and software designers	Three Manufacturing engineers; two Software designers; two Hardware designers

[a] Elektra is an information platform that VCC has developed together with Vector Informatik GmbH [67] for managing electrical and software requirements, architectures, components and their relationships.

level designs and narrowed solution space. In the three cases, the high-level manufacturing requirements are elicited, specified and owned by MAN and it is also MAN's responsibility to see to it that PD understands and realizes the requirements in a proper way. Consequently, MAN initiates and drives the communication process. Who should initiate and lead this process most likely differ between different organizational settings. But based on earlier experience from balancing requirements and the cases studied at VCC, we believe, in general, that the owner of the requirements being balanced is best suitable for this task. This is also in line with the Handshaking with implementation proposals technique where the product manager is responsible for the requirements specifications and initiates the handshaking process by sending the requirements to the development team [22]. Similarly, in SBCE, each function (e.g., MAN) is responsible for initiating and communicating requirements and alternative solutions (e.g., new manufacturing technologies and constraints) to other functions (e.g., PD) and collect their input in order to gradually converge the requirements and solution space and identify the best solution across the functions involved [18].

Step 1. The goal of Step 1 is to identify and get an overview of the balancing needs of the manufacturing requirements on lower design levels than they are formally analyzed and traded-off (e.g., systems area level, see Table 4 and Fig. 6). This should be performed for each new car project, and the goal is to identify the main part of the balancing needs in the early phases. In order to capture emerging balancing needs caused by late design and manufacturing changes, this step should also be continuously performed throughout the development cycle.

The identification can be performed in several ways, such as via experience and the introduction of design and manufacturing changes. For example, in Case 1 it was based on experience from concluded projects, in Case 2 the introduction of a new design of a car function, and in Case 3 new manufacturing requirements was elicited and specified. Based on the cases, MAN should at least pose the following three questions during the identification: (1)

Table 4
Design levels at VCC.

Level	Description	Example[a]
Attribute	Attributes describe how the customer (the user of a vehicle) should experience the car with their senses. An attribute can be achieved through several system solutions and functions. Manufacturing requirements and other stakeholders' requirements (e.g., aftersales) are also structured as attributes	MANREQ1: The duration time of calibration shall not exceed 10 s.[b]
Function	Functions describe what the car does and should be able to perform, either as a result of driver or passenger actions or as predefined behavior according to a specific situation. One function can be achieved through results from one or many systems. The function owner (FO) specifies the function in a function description and requirement (FDR)	PDREQ1: Opening and closing the tailgate of the car must be easy PDREQ: The tailgate of the car shall be possible to open and close automatically
System	Systems can either be a collection of sub-systems (so called system area, e.g., electrical architecture) or a sub-system consisting of a collection of hardware components (e.g., sensors, actuators and microcontrollers) and software components (applications and signal configurations) to which the functions are allocated. The system owner specifies the system in a systems requirement description (SRD)	PDREQ3: To be able to open and close the tailgate automatically, the range between its end positions must be calibrated in manufacturing
Component	On the component level the specifications are detailed on such a low level that each hardware and software component can be designed. Software and hardware designers specify the components in, such as software requirement specifications (SWRS) and design prerequisites hardware (DPR) HW)	PDREQ4: The software shall control the calibration of the end positions of the tailgate

[a] Product requirements are labeled PDREQ and manufacturing requirements are labeled MANREQ.
[b] Manufacturing requirements are always structured on attribute level despite their level of detail.

which designs and manufacturing operations have affected each other in earlier projects? (2) what design changes are being introduced? and (3) what manufacturing changes are being introduced?

After identifying the balancing needs in Step 1, the manufacturing requirements are broken down and cascaded to appropriate design levels, which are determined by the levels used in an organization. Based on Case 1, and the levels used at VCC, the breakdown of the manufacturing requirements is exemplified in Fig. 6. Here, a balancing need of the requirement, MANREQ1 (the duration time of calibration shall not exceed 10 s), structured on the attribute level, and the function automatic open-and-close of tailgate, structured on the function level, has been identified. The requirement of this function, PDREQ2 (the tailgate of the car should be possible to open and close automatically), is broken down to the sub-system tailgate maneuvering via the system area tailgate. The requirement on the tailgate maneuvering sub-system, PDREQ3 (to be able to open and close the tailgate automatically, while the range between its end positions must be calibrated in manufacturing), is in turn broken down to the software component controlling the logics of the automatic open-and-close of the tailgate). As PDREQ4 (the software shall control the calibration of the end positions of the tailgate) is a requirement on this software component, there is a need to explicitly balance its design in order to fulfill MANREQ1.

After the manufacturing requirements have been broken down and identified for the balancing needs (henceforth "balancing issues"), a BRASS performance list (see Fig. 4(B) and further details in Section 3.1.3) is established for each project. In this list, all the balancing issues in a project are compiled and their status is shown. The BRASS-performance list is a living document, where balancing issues are added and removed from the list and their status is continuously updated. After listing the balancing issues, the next step is to negotiate and specify requirements and solutions.

Step 2. Step 2 is initiated for each balancing issue listed in the BRASS performance list. In order to achieve a shared understanding and surfacing and clarifying the actors' underlying assumptions and differences in opinions, the manufacturing requirements and related design requirements are analyzed by suggesting and discussing alternative solutions. At first a relative large requirements and solution space is explored, which then gradually converges toward a final agreed-upon solution and requirements. In all three cases, MAN initiated and drove a series of focused group meetings, which constituted the core medium for communicating the balancing issues. In these meetings, a number of key persons participated and contributed with their expertise and knowledge. In between these meetings, the key persons communicated through email and face-to-face meetings in response to previous meetings and for preparing the next meeting (e.g., clarifying open issues and expose related information and decisions). Fig. 7 shows how Step 2 was carried out in Case 1.

In Case 1, the initial solution space consisted of a set of three alternative solutions for calibrating the end positions of the tailgate: (1) self-calibrated, (2) calibration by sending diagnostic services (e.g., ISO 14229 [54]) from external tester to the car, and (3) manual calibration. At Focus group meeting 1, it was agreed that Alternative 2 was the most feasible. In between this meeting and Focus group meeting 2, alternatives to shorten the calibration time in order to fulfill MANREQ1 (the duration time of calibration shall not exceed 10 s) had to be investigated and the impact of a newly proposed manufacturing set-up had to be clarified. To meet the time constraint, it was agreed to split the calibration into two algorithms at Focus group meeting 2. In between this meeting and Focus group meeting 3, it was necessary to investigate where these two algorithms had to be executed in the manufacturing set-up. At Focus group meeting 3, it was decided to perform the calibration at Stations 1 and 3 according to the new manufacturing set-up and the diagnostic services needed to be further detailed to be able to implement the algorithms in the manufacturing systems.

Relevant information being exchanged between the actors for sufficient descriptions and analysis of the identified balancing issues are specified in the BRASS one-pager (see Fig. 4(B) and further details in Section 3.1.3 and Appendix A). This information is also used for updating established managements systems for documenting requirements as well as for product and manufacturing solutions.

Step 3. Based on the implementation plans established in Step 2, Step 3 aims to validate the balanced and agreed-upon requirements and solutions. Both the quality of the test objects and their components/systems (digital mock-ups and physical cars) and the readiness of the methods and tools in manufacturing influenced the validation. Furthermore, even though it was endeavored to perform the validation in the early phases, in practice the solutions are primarily implemented and analyzed during the trial production on physically built vehicles in the late phases. This implies a high

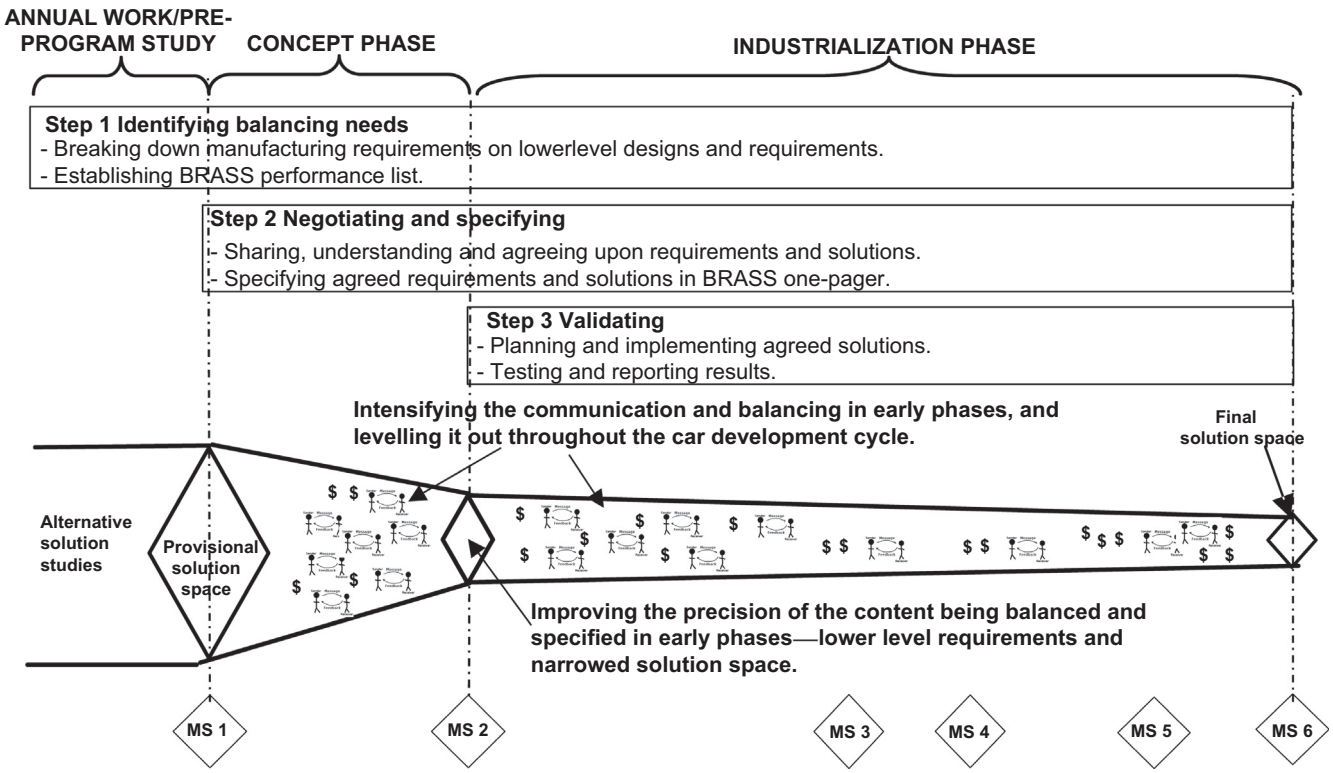

Fig. 5. The BRASS communication process at VCC.

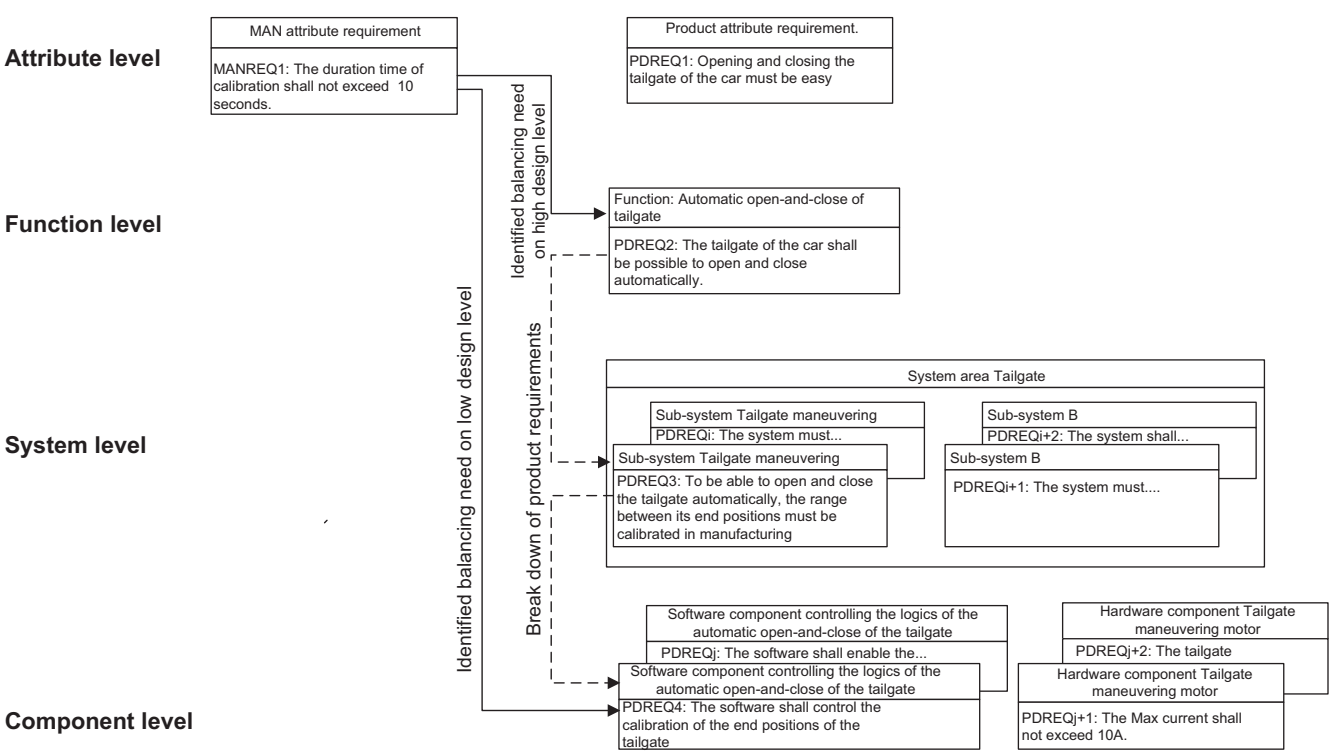

Fig. 6. Breakdown and cascading of manufacturing requirements of the fictitious tailgate example.

cost and a limited number of iterations before the start of production. Thus, Step 3 is merely performed in the industrialization phase and when planning it is important to carefully specify the conditions for proper validation.

During the validation, the feedback data is collected from building reports, but also, and even more importantly, by going and seeing for yourself in order to thoroughly understand how the implemented solutions actually work. Data are then analyzed

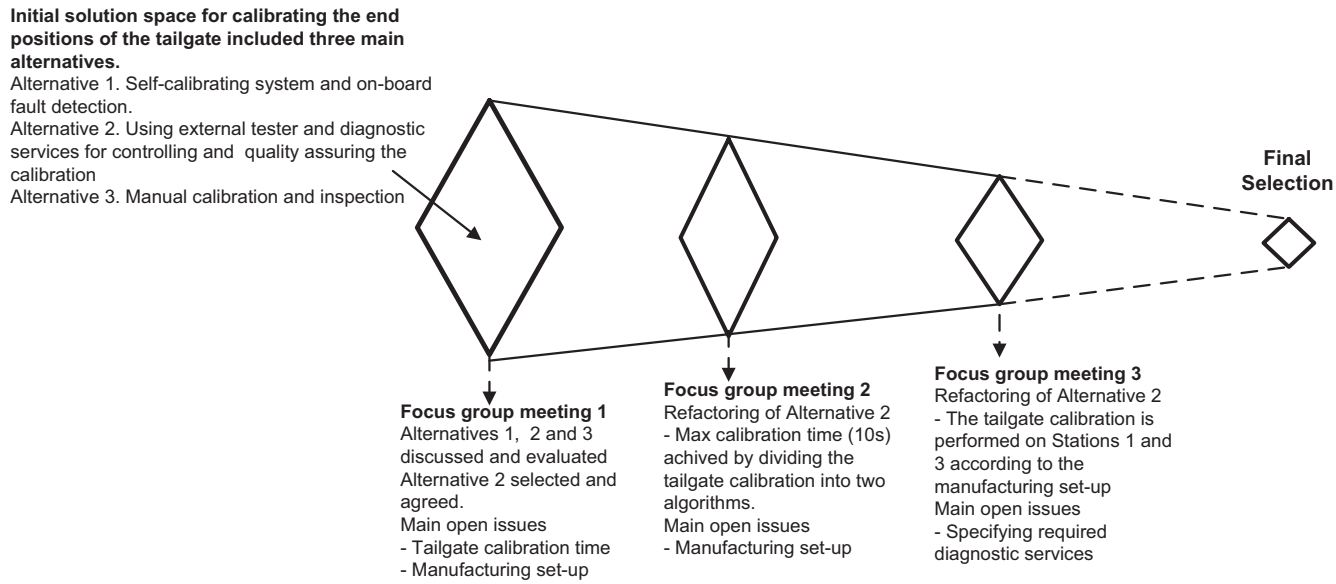

Fig. 7. Exemplification of Step 2 based on Case 1 (the fictitious tailgate example).

and the status of the agreed solutions and requirements is updated in the BRASS one-pager. If the validation failed, Steps 2 and 3 are iterated until manufacturing and design requirements, as well as feasible solutions, have been agreed upon and successfully validated.

Using the three cases for exemplifying Step 3 was not possible as the scheduling of the validation in the new car project is beyond the planning of the study presented in this paper.

3.1.3. Content

As previously mentioned, the content being communicated is documented in a BRASS performance list and a BRASS one-pager (see Fig. 4(B)). The BRASS performance list aims to get an overview of all the balancing issues identified in a car project and keep track of their status. For each balancing issue, requirements and solutions, but also other relevant information, are exchanged, such as open issues and established implementation plans, and are documented in a BRASS one-pager. The BRASS one-pager is primarily inspired by the A3 reports used at Toyota [20]. The A3 report refers to a standardized communication format and a way of expressing complex matters accurately and clearly on a single sheet of paper in order to achieve clear cross-functional communication. It was developed by Toyota based on the traditional A3 thinking approach [69] defined as an approach to solve problems and find opportunities for improvement in manufacturing on the shop-floor, but is also effectively used for cross-functional problem-solving in product development [20]. Typically, the A3 report is structured into elements (e.g., background, analysis, solutions/countermeasures, and implementation plan), which are guided by the Plan-Do-Check-Act (PDCA) paradigm for cyclic and continuous improvement [70]. Rudolf and Paulish [71] is the only study we have found using A3 reports in order to achieve early problem-solving in the development of large-scale software systems. However, the study merely reports on what to do on a high level, but not what the actual implementation should look like.

In order to capture relevant meta-data, a number of generic properties were used that we considered applicable to most issues. Table 5 describes and exemplifies the properties used in the BRASS performance list and Table 6 the ones used in the BRASS one-pager. Other company- or issue-specific properties can be attached to enrich analysis and understanding.

The extracted data of each of the balancing issues was filled out and documented in an MS Excel sheet (see Appendix A). Both the BRASS performance list and the BRASS one-pager were stored in a share-point site at VCC.

3.1.4. Actors

The dimension actor (see Fig. 4(C)) describes the required key parties involved (e.g., individuals, roles, and teams) and clarifies their responsibilities in the balancing. In total, 15 subjects (i.e. actors) representing different roles at MAN and PD were involved in the three cases studied (some of the MAN subjects were involved in more than one case). Most of them were identified during the identification of the balancing issues (Step 1 in Section 3.1.2) by the manufacturing engineer(s) who drives the balancing issue. Some of them were also identified during the focus group meetings (Step 2 in Section 3.1.2) such as one of the hardware designers in Case 3. Table 7 shows the roles involved for each of them.

In contrast to the industrial setting studied in [22], the actors of MAN and PD can play both the role of goal seeker and implementer. They may also shift during the development cycle, making the boundaries between the actors more fuzzy and complex in the setting studied here. For example, to meet the manufacturing requirements of a maximum time of 10 s (MANREQ1) for calibrating the end positions of the tailgate in Case 1, Alternative 2 was both implemented in the software by the systems/software designer (splitting the calibration into two algorithms) and in the manufacturing processes by the manufacturing engineer (deploying the diagnostic services in Stations 1 and 3). Therefore, it is important to clarify roles and responsibilities. This is in order to ensure that requirements and balancing tasks are not omitted or fall between two stools as well as to achieve traceability and consistency between the content produced by BRASS and related overarching documentation of product and manufacturing solutions.

4. Evaluation of BRASS at VCC

The evaluation of BRASS was performed as a dynamic validation [72] at VCC. The validation focused on Steps 1 and 2 in the communication process of BRASS (see Section 3.1.2) and the use of the BRASS one-pager for documenting the information. The main

Table 5
Properties of the BRASS performance list.

Property	Description	Example
Project	Project name	New car project XXX
Title	A descriptive title for the balancing issue, linking to the BRASS one-pager	Tailgate calibration in manufacturing
Impact	The parts of the design and operations in manufacturing that are affected	Function: Automatic open-and-close of the tailgate, System: Tailgate, Sub-system: Tailgate maneuvering, Component: The software controlling the logistics of the automatic opening-and-closing of the tailgate Stations 1 and 3 in manufacturing
Requirements	The requirements being balanced, including requirements ID and title	PDREQ3: To be able to open and close the tailgate automatically, the range between its end positions must be calibrated in manufacturing.PDREQ4: The software shall control the calibration of the end positions of the tailgate MANREQ1: The duration time of calibration shall not exceed 10 s
Status	Specifies the status of the balancing issue (e.g., not agreed, agreed, and validated)	Alternative 2 agreed

purpose was to get initial feedback from industry professionals (subjects) on four selected key aspects of BRASS: (1) ease of use, (2) leveraging mutual understanding, (3) quality of documentation, and (4) usefulness and applicability. Another reason is to get a buy-in from the company to implement BRASS [70]. The evaluation process consists of three main activities: (1) planning and preparation (Section 4.1), (2) applying BRASS in practice by using three real cases (Section 4.2), and (3) collecting and analyzing feedback on BRASS from questionnaires and follow-up interviews (Section 4.3). The results of these activities are reported in terms of lessons learned based on observations when applying BRASS to the three cases and the results from the questionnaires and interviews (Section 4.4).

4.1. Planning and preparations

Purposive sampling was used when selecting cases and subjects. The selection of cases should represent typical balancing issues between MAN and PD, but also include differences regarding involved organizational parts, engineering disciplines, and functions, systems and components. Three manufacturing engineers at VCC identified a number of cases based on experience and if new manufacturing requirements had been specified. Of these, three cases were then selected for this study. The subjects were selected in order to ensure that all the key roles were represented in the selected cases. Thus, they are the key persons at MAN and PD

Table 6
Properties of the BRASS one-pager.

Property	Description	Example
Project (see Table 5)	–	–
Title (see Table 5)	–	–
Impact (see Table 5)	–	–
Status (see Table 5)	–	–
Requirements specification	The requirements being balanced, including requirements title, ID, and specification	PDREQ3: To be able to open and close the tailgate automatically, the range between its end positions must be calibrated in manufacturing PDREQ4: The software shall control the calibration of the end positions of the tailgate MANREQ1:The duration time of calibration shall not exceed 10 s
Background/problem description	Description of background and problem	To enable the functioning of the automatic opening and closing of the tailgate, there is a need to calibrate the range between the tailgate's end positions and quality assuring the function in manufacturing
Solution alternatives	Descriptions of the alternative solutions to resolve the calibration of the tailgate's end positions	Alternative 1: Self-calibrating system and on-board fault detection Alternative 2: Using external tester and diagnostic services (e.g., ISO-14229–4:2012 [54]) for controlling and quality assuring the calibration Alternative 3: Manual calibration and inspection
Evaluation	Specifies the rationales for choosing a specific solution(s) (e.g., costs and quality).	Alternative 1: Not technically feasible Alternative 2: Moderate costs due to extra equipment and process time. High quality assurance level Alternative 3: High costs due to extra operator time. Low quality assurance level due to manual inspections Alternative 2 chosen (acceptable rise in cost and quality assurance level)
Selected solution	Detailed specification of the selected solution	Alternative 2 selected, see Test specification TS1
Actors	Required key parties	J. Andersson, manufacturing engineer (Station 1); S. Karlsson, manufacturing engineer (Station 3); S. Berg, function owner (Automatic open-and-close of the tailgate); K. Johansson, task leader; T. Larsson, system/software engineer (tailgate maneuvering system)
Planning	Descriptions and scheduling of necessary tasks (e.g., realizing and validating solutions, and refining requirements	2013 week10: Next meeting MS3: Detailing and specifying the calibration method in Test specification TS1 Requirement on diagnostic service for calibration specified in PDREQ5 MS4: Diagnostic service implemented in the software controlling the automatic opening and closing of the tailgate Calibration method implemented in external tester Validation of Alternative 2
Open issues	Actions required by the actors to progress with the balancing	PDREQ5 needs to be specified and implemented in software (resp. K. Johansson)

Table 7
Actors involved in each case.

Subjects	Organizational affiliation	Role	Case 1	Case 2	Case 3
Subject 1	MAN	Manufacturing engineer		X	X
Subject 2	MAN	Manufacturing engineer	X	X	X
Subject 3	MAN	Manufacturing engineer	X		
Subject 4	MAN	Manufacturing engineer			X
Subject 5	MAN	Manufacturing engineer	X	X	
Subject 6	PD	Task leader		X	
Subject 7	PD	Software designer			X
Subject 8	PD	Task leader	X		
Subject 9	PD	Systems and software designer	X		
Subject 10	PD	Software designer			X
Subject 11	PD	Task leader		X	
Subject 12	PD	Systems and software designer		X	
Subject 13	PD	Systems and software designer		X	
Subject 14	PD	Hardware designer			X
Subject 15	PD	Hardware designer			X

for balancing each of the identified and selected issues. In total, 15 subjects were involved (see Table 7 in Section 3.1.4).

In order to measure the subjects' view on BRASS after applying it in the three cases, a self-administrated questionnaire [73] was designed inspired by [49]. Building on the questions in the questionnaire, an interview instrument was also developed in order to follow-up and enrich the answers from the questionnaire.

The self-administrated questionnaire collected opinions on how the subjects thought that the current situation could be changed regarding the selected key aspects, if BRASS was implemented at VCC. To measure and evaluate the subjects' opinions, a nine-point Likert scale was used. The scale represents the levels of the subjects' estimated value of BRASS, which are then converted into numbers. Furthermore, the subjects were asked to give three estimates: (1) the probable value should reflect what they think is most probable, (2) the best value should reflect the best case, and (3) the worst value should reflect the worst case. This allows a rough estimation of the uncertainty of the answers by looking at the span of the best and worst values. In the example given in Fig. 8, the probable value is that the effort will be less than today. The given best and worst values show that there is a large span (i.e. high uncertainty), however, the difference between the best and worst values is one (3–2 = 1) indicating, as the probable value, that the effort will be less.

To ensure that the interview instrument and the questionnaire were comprehensible and unambiguous, drafts of them were iteratively reviewed and piloted several times.

4.2. Applying BRASS

Prior to applying BRASS, pre-meetings were held with the subjects where the background and the purpose of BRASS and its main properties were presented and discussed. In practice, the subjects representing MAN were informed at one group meeting while three separate meetings (one for each case) with subjects representing PD were held. 14 of the 15 subjects participated and the duration of each pre-meeting was about two hours. Even though about 28 man-hours (14 subjects times 2 h) were spent, we experienced that this was necessary in order to be efficient. For example, since one of the subjects involved in Case 1 could not participate in the pre-meeting, much of the discussion concerned the BRASS-framework itself instead of discussing the calibration of the tailgate during the first focus group meeting.

The central activity when applying BRASS to the selected cases consisted of a series of focus group meetings, involving the subjects selected for each case. During the meetings, the subjects negotiated and refined the related requirements and developed and agreed upon suitable solutions for them. In addition, necessary information was collected from archival records and documents at VCC (e.g., requirements and specifications, and continuously exchanged between the subjects through face-to-face meetings and e-mail. The exchanged information was documented in the BRASS one-pager and summarized in the BRASS performance list (see Tables 5 and 6 in Section 3.1.3). In total, seven meetings were conducted and the duration of the meetings was 1–2 h.

One of the researchers participated in all the meetings with the main task of observing them, and helping and explaining how to use BRASS (e.g., fill out the BRASS one-pager), but without influencing the results of the meetings.

4.3. Collecting and analyzing feedback data

After applying BRASS, the involved subjects were asked to answer the questionnaire. In total, 10 of the 15 subjects answered the questionnaire and the characteristics of them and their involvement in the Cases 1–3 are given in Table 8. Follow-up interviews were also conducted with two of the subjects (Subjects 1 and 6), using semi-structured interviews [73]. Subjects 1 and 6 (henceforth interviewees) and were chosen because they represent MAN and PD and had much experience working with software-intensive systems. Both interviews were conducted by one of the researchers and held in Swedish. The length of them varied between 30 and 40 min.

One of the researchers transcribed the recorded interviews and performed the data analysis as it was not possible to allocate extra resources for this work. The data analysis was divided into three main stages based on the approach described by Miles and Huberman [74] with three concurrent flows of activity: data reduction, data display and conclusion drawing/verification. First, statements from the raw data that expressed pros and cons in relation to the key aspects of the applicability of BRASS were extracted. Next, the statements were structured and displayed in an excel sheet based on the selected key aspects. The traceability between the statements and the raw data were also established. Finally, peer debriefing of the results was performed. The researcher reviewed and discussed the results together with the two other researchers and one industry representative in a series of meetings. All three stages were iterated several times.

4.4. Lessons learned

The results from the interviews and the self-administrated questionnaire are presented and discussed in the following sections. The diagrams in Figs. 9–16 show the normalized estimates.

4.4.1. Ease of use

For evaluating the ease of use, the subjects were asked to give their opinions on how easy it was to learn and understand

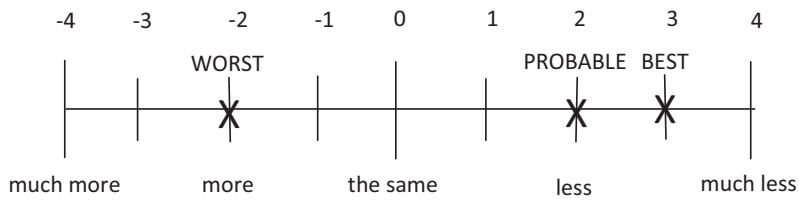

Fig. 8. Conversion of levels into nine-point Likert scale.

Table 8
Characteristics of the subjects.

Subject	Characteristics of the subjects						Case
	Organizational affiliation	Role	Time at VCC	Time at current position	Experience of developing software-intensive automotive systems	Gender	
Subject 1	MAN	Manufacturing engineer	>10 years	>10 years	>10 years	Male	1, 3
Subject 2	MAN	Manufacturing engineer	>10 years	6–10 years	>10 years	Male	1, 2, 3
Subject 3	MAN	Manufacturing engineer	>10 years	1–5 years	1–5 years	Female	1
Subject 4	MAN	Manufacturing engineer	>10 years	6–10 years	6–10 years	Male	3
Subject 5	MAN	Manufacturing engineer	>10 years	6–10 years	>10 years	Male	1, 2
Subject 6	PD	Task leader	>10 years	6–10 years	>10 years	Male	2
Subject 7	PD	Software designer	>10 years	>10 years	>10 years	Male	2
Subject 8	PD	Task leader	1–5 years	1–5 years	1–5 years	Female	1
Subject 9	PD	Systems and software designer	1–5 years	1–5 years	1–5 years	Male	1
Subject 10	PD	Software designer	>10 years	1–5 years	6–10 years	Male	3

BRASS, and the effort of use when compared with the practices currently used. The bars in Fig. 9 show the summarized estimations for all the subjects (total estimate) and the subjects belonging to PD (PD estimate) and MAN (MAN estimate) regarding how easy they perceived it was to learn and understand BRASS. The black bars show the estimated probable value, the white bars show the estimated best value, and the gray bars show the estimated worst value. The estimated best and worst values indicate the uncertainty in the estimated probable value. The total estimate in Fig. 9, shows that the subjects perceived that BRASS was easy to learn and understand. However, the gray and white bars showing the worst and best estimates show a high uncertainty in the total estimate, while the difference between the best and worst values indicates, as the probable value, that BRASS was easy to learn. Furthermore, both the MAN estimate and PD estimate show that it was easy to learn BRASS, but the uncertainty of the PD estimates is higher compared to the MAN estimate as BRASS is perceived as easy to learn even in the worst case of the MAN estimate.

During the interviews, both interviewees acknowledged that BRASS was easy to learn and its purpose was clear. However, one of them mentioned that the most difficult and challenging part of BRASS was to effectively identify the need of balancing requirements and solutions between PD and MAN. To tackle this, two extreme approaches were brought up. The first one implies that all requirements in a car project are reviewed and any identified dependences between MAN and PD requirements indicate a balancing need. However, a car project contains about 100,000 requirements along with other development artifacts (e.g., specifications, models and standards), which makes this approach enormously resource consuming and not feasible. The other extreme approach merely relies on experiences from concluded projects—i.e. the balancing needs are based on the needs identified in earlier projects. The effort is much less, but a big disadvantage is the obvious risk of omitting new balancing needs. Thus, a compromise was suggested between these two extremes by, for example, relying on experience along with singling out new product features and manufacturing prerequisites in the projects and reviewing related development artifacts.

Regarding the effort of using BRASS, the subjects estimated that the balancing effort is likely to be less than for the current practices

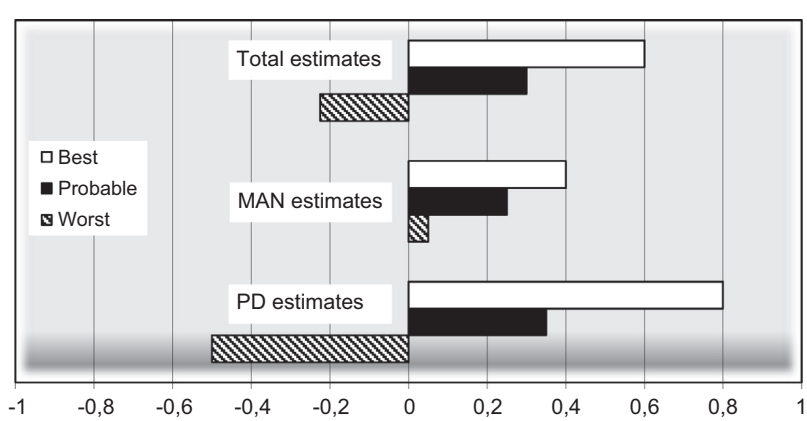

Fig. 9. Easy to learn (normalized estimates).

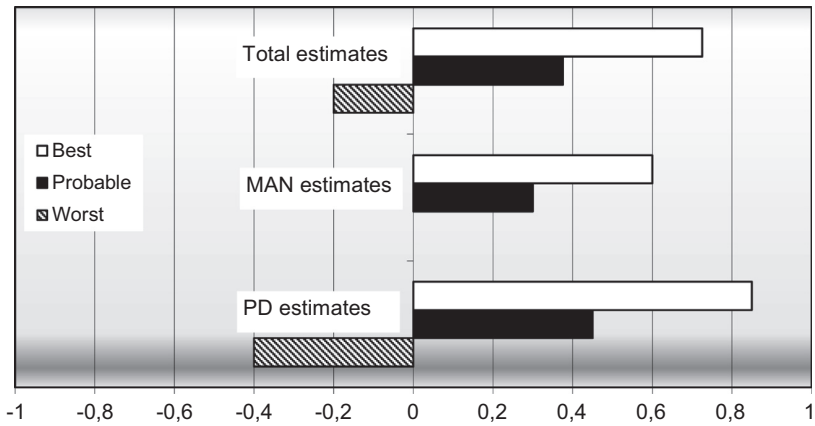

Fig. 10. Effort of use (normalized estimates).

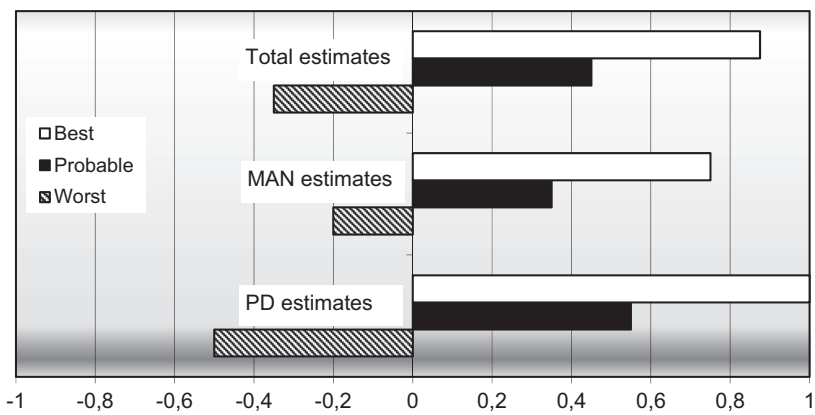

Fig. 11. Mutual understanding (normalized estimates).

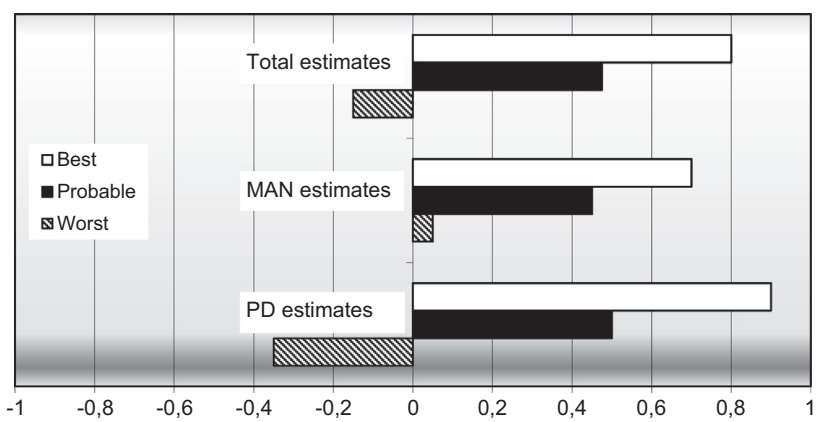

Fig. 12. Precision of the documentation (normalized estimates).

used as shown in Fig. 10 (black bars). This opinion was shared across PD and MAN (PD estimate and MAN estimate). The white and gray bars of the best and worst estimates indicate that the uncertainties in the estimates are high, however, the difference between the best and worst values for the total estimates is 0.52 (0.72–0.2), indicating, as the probable value, that the balancing effort was expected to decrease using BRASS.

The interviewees stated that the effort of arranging and participating in the focus group meetings, and documenting the exchange information was feasible. As for the results of the questionnaire, they also thought that BRASS could lessen the balancing efforts compared with the practices currently used. One reason mentioned for this was that the balancing must be performed regardless if BRASS was used or not, and it is most likely that BRASS would reduce the resources currently spent on fixing balancing issues in later project phases. One of the interviewees also claimed that BRASS could shorten the development time because BRASS provides a systematic way of discussing requirements and solutions, in contrast to current ad-hoc practices. The ad-hoc practices at VCC is generally characterized by the fact that the balancing is reactive initiated by the late detection of defects and information is mainly exchanged verbally via telephone calls

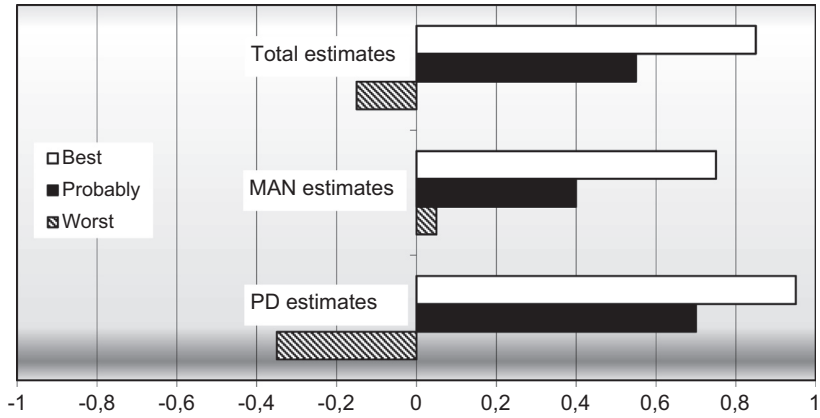

Fig. 13. Supporting decision-making (normalized estimates).

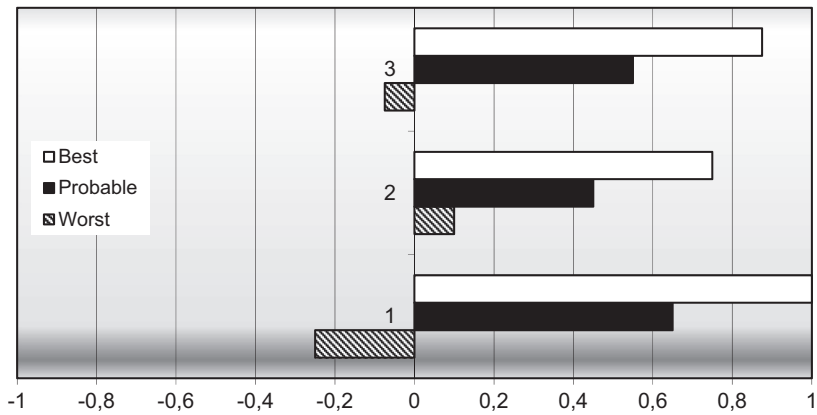

Fig. 14. Prevention of late defect detection (normalized estimates).

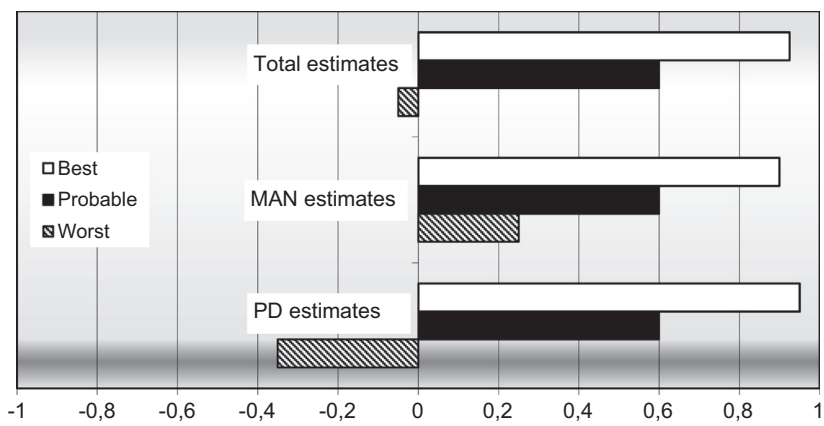

Fig. 15. Overall usefulness (normalized estimates).

and/or unscheduled face-to-face meetings, or email conversations. The information is also seldom sufficiently specified and a high reliance on the staffs' experience and knowledge is prevalent. BRASS can help visualize and clarify needs in the early phases, for example, by establishing validation plans and procedures and identifying needs of building and developing early prototypes. Furthermore, in order to be easy to use, both interviewees emphasized that this is highly dependent on whether the intended scope of BRASS and its purposes are kept, and its practices are continuously streamlined (e.g., optimizing the focus group meetings and the documentation in the BRASS one-pager).

4.4.2. Mutual understanding

In our earlier studies at VCC, we found that insufficient shared understanding of the information being communicated and coordinated between PD and MAN was one of the main causes for the balancing problems [27]. Thus, the subjects were asked to estimate if BRASS would improve the mutual understanding of PD and MAN needs. The black bars in Fig. 11 shows that it is likely that BRASS would improve the mutual understanding of each other's needs and aspirations, but the uncertainty in the estimates is high (white and gray bars in Fig. 11). It can also be seen that the subjects affiliated with MAN believe that the mutual understanding will

improve even in the worst case (see MAN estimate in Fig. 11) Furthermore, the difference between the best and worst values for the total estimates is 0.52 (0.87–0.35), indicating that BRASS has the potential of improving mutual understanding.

During the interviews, both interviewees highlighted the importance of gaining a better understanding and insight into each other's needs, for example, better insight into the implications of manual operations (e.g., manual calibration of the tailgate) in manufacturing. The interviewees believed that using BRASS would be an efficient way of intensifying the communication between PD and MAN, leading to an improved mutual understanding. For example, the effort used for preparing (about one man-hour per subject) and performing the three focus meetings (about two man-hours per subject) in Case 1 could be estimated at 50 man-hours. In comparison, for Case 1, the effort used to resolve the late detection of defects (resources dedicated to such activities as data collection, analyzing, reporting and decision making) due to a lack of mutual understanding in an earlier car program could be estimated at 480 man-hours in Pernstal et al. [27]. In addition, the interviewees believed that the quality of requirements and specifications and the efficiency of the solutions developed would increase and help tear down the imaginary barriers between the PD and MAN.

4.4.3. Quality of the documentation

The need to improve documentation was identified in Pernstal et al. [27]. Overall, the subjects believed that BRASS would probably improve the precision of the documentation (black bars in Fig. 12). However, the white and gray bars for the total estimate and the PD estimate show that the uncertainty in the estimates is high, while it is somewhat lower for the MAN estimate. In addition, the black bars in Fig. 13 show that the documentation produced by BRASS would most likely provide better support for decision making, but the uncertainty of the estimates is even higher than for the estimates of the precision of the documentation. On the other hand, the difference between the best and worst values for the total estimates of the precision of documentation is 0.65 (0.8–0.15) and the total estimates of the supporting decision-making is 0.7 (0.85–0.15), indicating that the subjects expect that the quality of the documentation will be improved using BRASS.

The interviewees thought that the BRASS performance list and the BRASS one-pager would be helpful. This is because the BRASS performance list provides an overview of what needs to be balanced and the status of each balancing issue in the projects, which is currently not available at VCC. Furthermore, gathering and structuring requirements and solutions in the BRASS one-pager clarify both the boundaries for the solutions and the consequences of realizing the requirements. The interviewees also believed that the BRASS documents could be used to show decisions taken during a project, learn from concluded projects, and transfer and improve solutions between projects. In addition, the BRASS documents could supplement requirements specifications in the requirements management systems and provide information for updating them. For example, suggested improvements in the BRASS one-pager are used as lessons learned when specifying the requirements for a new car model.

Storing and archiving the BRASS documentation in one place and in a structured way were brought up as a main prerequisite for the effective use of the documents. As one of the interviewees stated: "*A lot of information is spread in different systems and project databases, and much of my time is spent on finding the information, and sometimes I never find it.*" Furthermore, it was preferred to not integrate the documentation with the formal routines in order to keep the effort low for handling the documentation. One of the interviewees suggested that the BRASS documentation should be informal and stored in something like an internal wiki or share point.

4.4.4. Usefulness and applicability

To gauge the overall usefulness and applicability of BRASS the subjects were asked to estimate three aspects relative to the current situation: (1) the capability of BRASS to prevent late defect detection (Fig. 14), (2) the overall usefulness of BRASS (Fig. 15), and (3) if a full-scale implementation of BRASS would be beneficial and feasible (Fig. 16).

The black bars in Fig. 14 show that BRASS would probably reduce the number of defects detected in the late phases. The subjects also perceived BRASS as useful in general (black bars in Fig. 15) and a full-scale implementation was expected to improve the current situation balancing requirements and solutions across PD and MAN at VCC (black bars in Fig. 16). However, the gray and white bars for the total estimates and PD estimates in Figs. 13–15 indicate a high uncertainty in these estimates, while it is lower for the MAN estimates as they show that BRASS would have a positive effect on all three aspects even in the worst cases.

The interviewees believed that using BRASS for identifying and discussing balancing issues in early phases would in turn increase the possibilities of preventing late and costly changes. This opinion was also indicated by the results of applying BRASS to the cases. For example, in Case 2, the prior postmortem analysis [27] showed that the implication costs in manufacturing for managing the design of the chassis car function and systems could be estimated at 1000 KUS$ over the product cycle (seven years). Using BRASS resulted in a new system design that is better adapted to meet the manufacturing requirements, having the potential of

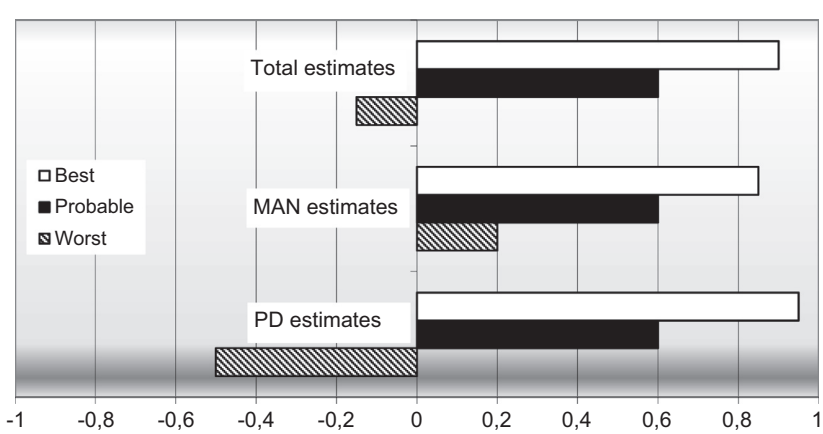

Fig. 16. Full-scale implementation (normalized estimates).

eliminating the manufacturing implication cost. However, the interviewees pointed out that BRASS could reduce, but not eliminate, late discovery and intervention of balancing issues, since there is always a risk of omitting issues or not agreeing on them.

Even though BRASS shows promising results and the subjects' view on the potential benefits of using BRASS seem overall to be positive, there are some concerns. First, regardless of using BRASS or not, there is always a challenge to acquire required resources and competencies in the early phases of the projects. On the other hand, one of the interviewees thought that BRASS could reduce this problem. This is because current practices for balancing are mainly performed on an ad-hoc basis with an often high dependency on the experiences of the staff involved. BRASS allows a reduction in this dependency as the balancing issues are identified and worked on in a more systematic way. For example, newly introduced designers would be much more aware of the balancing needs across PD and MAN if they are systematically identified in earlier project phases.

Second, like all introductions of new working methods, an overall concern that was brought up is the need of attaining acceptance and adoption of BRASS throughout the company. Issues that need to be balanced between PD and MAN usually become tangible and immediate first in the late project phases. Thus, there is always a risk of postponing or ignoring problems and later act as a firefighter instead of learning and using a systematic method for preventing problems in early phases. One of the interviewees suggested a step-wise implementation of BRASS where it first is applied to more cases in the new car model project used in this study, and then fully applied from the start of the next new car model project.

Third, to fully understand and evaluate BRASS, it should be used from the start of a project until the start of production. However, this kind of evaluation is difficult to carry out as the lead-time of car projects is usually between two and four years.

4.5. Limitations

A major threat is the external validity. In this study, BRASS has been tailored and applied in order to address core challenges and to resolve specific problems in RE identified at one company, namely VCC. Even though the industrial setting (complex products and organizational structure) and the core challenges at VCC are common in companies developing large-scale software-intensive systems, it is not representative of all companies developing such systems due to a number of company specific preconditions. VCC is a rather mature organization when it comes to RE, using a comprehensively documented processes and established RE-tools for managing requirements. Furthermore, the structure of the requirement abstraction levels and the process for breaking down and balancing requirements are pre-defined and the employees at VCC are used to work in formalized processes and handle large quantities of product documentation. Another important precondition to using BRASS was changing the established and formalized RE tools and processes at VCC in order to resolve their specific problems would be too complex and costly. It is our belief that BRASS works best when used as an add-on method needed for improving requirements breakdown and balancing in organizations with relatively mature RE-processes. Thus, the generalizability of the resulting tailoring and evaluation of BRASS presented in this study is limited.

However, case studies rather pursue to characterize, explain, and understand a phenomenon of interest by studying it in a real-life setting than generalize the findings beyond the actual setting. Besides the aim of solving the specific problems at VCC, a main objective of this case study is therefore to show and demonstrate how the BRASS framework can be tailored, applied and evaluated as a method used in a real industrial setting and not the actual resulting tailoring of BRASS. We believe that BRASS has the potential to help practitioners improve their RE, and in particular, requirements coordination and communication problems in other industrial settings. Thus, to enhance the possibilities for the readers to judge whether BRASS can be used in other settings, we provided a thorough description and exemplification of how BRASS was applied to the industrial setting at VCC in Section 3.1.

One major threat in this study is that unreliable and biased data are used. Despite the consistently positive attitude toward BRASS, any conclusions from the answers should be drawn with a great deal of caution, primarily because of the relatively small sample representing the organizations investigated and since the answers might have been biased by the range of options given in the questions, as some people tend to provide positive answers regardless of their real opinion [73]. To guard against this, the subjects were also guaranteed anonymity and that sensitive information would neither be published nor possible to trace to individuals. Two follow-up interviews were conducted. In addition, two of the members of the research team had worked for more than 10 years in the design and manufacturing engineering of software-intensive automotive systems at VCC. Analyses and conclusions were also discussed with experienced SPI researchers.

5. Conclusions

Industries involved in the large-scale development of software-intensive systems are facing many challenges as the complexity and uncertainty of development tasks are substantial. In some of these industries, such as the automotive one, the rapid increase of software has also led to new challenges concerning the organizational structure and the development processes since they have traditionally been adapted for the development of mechanical parts, which are relatively stable over time in contrast to software parts. A key success factor in such development is RE. In particular, the core challenges of attaining sufficient coordination and communication of requirements between actors are spread across a multitude of different departments and engineering disciplines. This is further complicated by the large amount of requirements that demand effective and efficient ways of enabling communication and coordination—simply writing "perfect" requirements is not an option due to the costs of achieving "perfection."

With the overall aim of addressing these challenges and in response to the needs identified at our industrial partner, VCC, this paper presents and applies a flexible and lightweight RE framework called Balancing Requirements and Solution Space (BRASS). BRASS combines goal-oriented requirements communication with the lean practice of Set-Based-Concurrent Engineering and emphasizes communication over focusing on achieving perfection in requirement specification.

BRASS consists of four dimensions, *communication, content, connections,* and *actors*, which are based on its main objectives: (1) intensifying requirements communication throughout the development cycle, (2) enhancing the mutual understanding and the quality of requirements and solutions specifications (3) being adaptable to the surroundings in which it is used (e.g., established RE processes and tools), and (4) clarifying the accountabilities and responsibilities of the actors concerned.

BRASS was developed in close cooperation with VCC driven by the core challenges and the specific problems identified at the company. As part of the development process, the applicability of BRASS was evaluated in comparison with current practices at VCC through a dynamic validation. For this, three real cases in an ongoing project, mirroring VCCs specific problems in balancing requirements and solutions across the departments of PD and

Appendix A. Brass one-pager

BRASS one-pager	Title: Tailgate calibration	Project: Car xxx	Status: Agreed
Type of requirements: Manufacturing and product requirements (Function, system and component)	Attribute/Function/System/Component: Function: Automatic open-and-close of the tailgate System: Tailgate, Sub-system: Tailgate maneuvering Component: The software controlling the logics		Date/Issue: 20121203
Requirement IDs: Man REQ MANREQ1:The duration time of calibration shall not exceed 10 seconds PD REQ PDREQ3: To be able to open and close the tailgate automatically, the range between its end positions must be calibrated in manufacturing. PDREQ4: The software shall control the calibration of the end positions of the tailgate.	Responsible Manufacturing: J. Andersson, manufacturing engineer (Station 1) S. Karlsson, manufacturing engineer (Station 3)	Responsible R&D: S. Berg, function owner (automatic open-and-close of the tailgate) K. Johansson, task leader and T. Larsson, system/software engineer (tailgate maneuvering system)	

Issue description	Set of alternative solutions	
To enable the function automatic open-and-close of the tailgate, there is a need of calibrating the range between the tailgate's end positions and quality assuring the funct ion in manufacturing.	**Description of alternatives**	**Evaluation of alternatives**
	A1: Adaptive self-calibrating system and on-board fault detection (DTCs) of incorrectly calibration are used for quality assurance.	Best alternative regarding production cost and quality assurance. Not possible due to difficulties in assuring that the tailgate is delivered fully calibrated from manufacturing.
	A2: The calibration and fault detection is performed by sending diagnostic requests from external tester. Faultcodes for detection of not and incorrectly calibrated system are used for quality assurance. The overall procedure for calibrating the windows is (see also attached PDF):	*Equipment with diagnostic communication needed → Production Cost ↗ *Extra process time for sending calibration command → Production Cost ↗ *Sensitivity to product and process changes → Quality Assurance ↘ and Production Cost ↗ Best alternative, because of relatively low production costs and high quality assurance.
	A3:. The tailgate calibration is carried out by manual activation. For quality assurance, functions are manually activated/deactivated by an operator who evaluates the functionality by inspection.	*Extra man-hour per unit →Production Cost↗↗ *Relying on operator commitment and understanding→ Quality Assurance↘↘ *Limited documentation and traceability→ Quality Assurance↘ *Sensitivity to product and process changes→ Quality Assurance↘ and Production Cost↗ Not feasable in manufacturing due to relatively high production costs and low quality assurance.
	Selected solution	
	Alternative 2 selected, see Test specification TS1	
Planning	**Open issues**	
2013 week10: Next meeting MS3: Detailing and specifying the calibration method in Test specification TS1. Requirement on diagnostic service for calibration specified in PDREQ5. MS4-4w: Diagnostic service implemented in the software controlling the automatic open-and-close of the tailgate. Calibration method implemented in external tester. MS4: Validation of Alternative 2.	20120917 Focus group meeting 1: Alternatives 1, 2 and 3 discussed and evaluated Alternative 2 selected and agreed. Main open issues - Calibration time - Manufacturing set-up Alternative 2 is agreed upon, but needs to be detailed. 20121028 Focus group meeting 2: Refactoring of Alternative 2 - Max calibration time (10s) achived by dividing the calibration into two algorithms. Main open issues - Manufacturing set-up 20121203 Focus group meeting 3 Refactoring of Alternative 2 - The calibration is performed on Stations 1 and 3 according to the manufacturing set-up Main open issues - PDREQ5 needs to be specified and implemented in software (resp K. Johansson).	

MAN, were used. Initial feedback was collected in questionnaires and interviews from 10 industry professionals involved in the case.

Overall, the results of using and validating BRASS show that it is feasible and applicable in an industrial setting. The feedback indicates that BRASS is easy to use and useful and cost-effective for resolving the specific problems because of its potential of systematically improving communication in early phases, mutual requirements understanding, and the quality of specifications. However, there are also concerns such as requiring necessary resources and competences in early phases and attaining acceptance and the adoption of BRASS throughout the company.

In this paper, we focused on the specific problems identified in the process of balancing requirements and solutions in the large-scale development of software-intensive systems at VCC. However, due to BRASS's flexible design, we believe that it can suit and satisfy the needs of other industrial settings, providing a starting point for practitioners to improve their specific problems in RE, and in particular, requirements coordination and communication.

6. Future work

The promising results from the initial validation has led to that VCC has decided to perform further validations and use BRASS in a large-scale pilot. This pilot will be undertaken over the next seven years (a typical new car life cycle) in order to obtain a better understanding of how all parts of BRASS can be tailored, evaluate its applicability over a full product development cycle, and identify what works and what needs to be changed, giving decision-support for refining BRASS before it is implemented in full-scale. In particular, the applicability of the validation step of balanced and agreed requirements and solutions, as described in Section 3.1.2 (Step 3), must be further investigated. Furthermore, improvements and extensions of BRASS are needed, including pertinent techniques and procedures in order to, for example, effectively and systematically identify the balancing needs in the early development phases.

Two main tactics will be used for achieving an understanding, acceptance, and adoption of BRASS throughout the company as a method supporting engineers in their daily balancing tasks. A step-wise introduction and implementation using the large-scale pilot where cases representing a wide range of different characteristics regarding technologies (e.g., active safety, HMI, and engine control), requirements and specifications (e.g., function and system requirements specifications), and roles and practitioners involved are selected. The main objective with this tactic is to informally diffuse awareness and knowledge about BRASS within VCC through the practitioners involved in the cases piloted.

The second tactic aims to integrate BRASS into established RE-processes and tools in a more formal way. Process descriptions and instructions prescribing what to do in projects are documented in the Business Management System (BMS) at VCC. Currently, an operational development project focusing on the work process

for handling car function designs affecting the manufacturing processes in car programs is in progress. This project is driven by the quality department at MAN and the complete vehicle department at PD. One aim is to change established instructions and create new ones in BMS. BRASS has guided the development of this handling where its parts have been integrated and realized in the instruction. For example, the Step 1 (identifying balancing needs) in the BRASS communication process uses an elaborated list where the functions are classified regarding their novelty for each car program. New or modified functions are singled out and the need of balancing them is analyzed. Traceability and consistency between the documents produced by BRASS and related formal documentation was also found important. Formal information (e.g., requirements and specifications) are documented in the established document management systems (e.g., Siemens Teamcenter Systems Engineering and Requirements Management) but it was preferred to not handle the BRASS performance lists and one-pagers in these systems in order to keep the effort low for handling the documentation. VCC has therefore established a shared website where these documents are stored with trace-links to documents in the established document management systems.

Acknowledgment

The authors would like to express their gratitude to all those involved in this study and especially the participants at VCC.

References

[1] K. Grimm, Software technology in an automotive company – major challenges, in: Proceedings of the 25th International Conference on Software Engineering (ICSE'03), Portland, Oregon, USA, 2003.
[2] K. Venkatesh Prasad, M. Broy, I. Krueger, Scanning advances in aerospace & automobile software technology, Proc. IEEE 98 (2010) 510–514.
[3] M. Broy, I.H. Kruger, A. Pretschner, C. Salzmann, Engineering automotive software, Proc. IEEE (2007) 95.
[4] J. Dannenberg, C. Kleinhans, The coming age of collaboration in the automotive industry, Mercer Manage. J. 18 (2004) 88–94.
[5] T. Gülke, B. Rumpe, M. Jansén, J. Axmann, High-level requirements management and complexity costs in automotive development projects: a problem statement, in: B. Regnell, D. Damian (Eds.), Requirements Engineering: Foundation for Software Quality Engineering and Managing Software Requirements, Springer, 2012.
[6] M. Weber, J. Weisbrod, Requirements engineering in automotive development: experiences and challenges, IEEE Softw. 20 (2003) 16–24.
[7] S. Konrad, M. Gall, Requirements engineering in the development of large-scale systems, in: Proceedings of the 16th IEEE International Requirements Engineering Conference, 2008, pp. 217–222.
[8] R.E. Kraut, L.A. Streeter, Coordination in software development, Commun. ACM 38 (3) (1995) 69–81.
[9] B. Curtis, H. Krasner, N. Iscoe, A field study of the software design process for large systems, Commun. ACM 31 (11) (1988).
[10] M.E. Conway, How do committees invent?, Datamation 14 (4) (1968) 28–31
[11] D.L. Parnas, On the criteria to be used in decomposing systems into modules, Commun. ACM 15 (12) (1972).
[12] J.D. Thompson, Organizations in Action, McGraw-Hill, New York, US, 1967.
[13] P. Adler, Interdepartmental interdependence and coordination: the case of the design/manufacturing interface, Organ. Sci. 6 (2) (1995) 147–167.
[14] CMMI, Capability Maturity Model Integration Version 1.3. Technical Report CMU/SEI-2010-TR-033, Software Engineering Institute (SEI), 2010. <http://www.sei.cmu.edu/cmmi/> (accessed February 2013).
[15] ISO/IEC Std. 15504: Information Technology – Process Assessment, Part 1 to Part 5, International Organization for Standardization, 1998–2005.
[16] J.P. Kuilboer, N. Ashrafi, Software process and product improvement: an empirical assessment, Inf. Softw. Technol. 42 (2000) 27–34.
[17] S. Zahran, Software Process Improvement: Practical Guidelines for Business Success, Reading, Addison-Wes, 1998.
[18] D.K. Sobek, A.C. Ward, J.K. Liker, Toyota's principles of set-based concurrent engineering, Sloan Manage. Rev. 40 (2) (1999).
[19] S.C. Wheelwright, K.B. Clark, Accelerating the design-build-test cycle for effective product development, Int. Market. Rev. 11 (1994) 32–46.
[20] J.M. Morgan, J.K. Liker, The Toyota Product Development System: Integrating People, Process, and Technology, Productivity Press, New York, 2006.
[21] K.B. Clark, T. Fujimoto, Product Development Performance: Strategy, Organization and Management in the World Auto Industry, Harvard Business School, Boston, 1991.
[22] S. Fricker, T. Gorschek, C. Byman, A. Schmidle, Handshaking with implementation proposals: negotiating requirements under-standing, IEEE Softw. 27 (2) (2010) 72–80.
[23] K. Schwaber, M. Beedle, Agile Software Development with Scrum, Prentice Hall, Upper Saddle River, 2001.
[24] T. Dybå, T. Dingsøyr, Empirical studies of agile software development: a systematic review, J. Inf. Softw. Technol. 50 (2008) 833–859.
[25] X. Wang, K. Conboy, O. Cawley, "Leagile" software development: an experience report analysis of the application of lean approaches in agile software development, J. Syst. Softw. 85 (2012) 1287–1299.
[26] J. Pernstal, R. Feldt, T. Gorschek, The lean gap: a review of lean approaches to large-scale software systems development, J. Syst. Softw. 86 (2013) 2797–2821.
[27] J. Pernstal, R. Feldt, T. Gorschek, D. Florén, Communication Problems in Software Development—A Model and its Industrial Application, 2013. <jpernsta@volvocars.com>.
[28] L. Almefelt, F. Berglund, P. Nilsson, J. Malmqvist, Requirement management in practice: findings from an empirical study in the automotive industry, Res. Eng. Design 17 (3) (2006).
[29] A. Abran, J.W. Moore, et al., Guide to the software engineering body of knowledge (SWEBOK®), IEEE Computer Society Guide, 2004.
[30] G. Kotonya, I. Sommerville, Requirements Engineering: Processes and Techniques, John Wiley, New York, 1998.
[31] B.H. Reich, I. Benbasat, Measuring the linkage between business and information technology objectives, MIS Quarterly 20 (1) (1996) 55–81.
[32] G.C. Murphy, D. Notkin, K.J. Sullivan, Software reflexion models: bridging the gap between design and implementation, IEEE Trans. Softw. Eng. 27 (4) (2001) 364–380.
[33] M. Unterkalmsteiner, R. Feldt, T. Gorschek, A taxonomy for requirements engineering and software test alignment, ACM Trans. Softw. Eng. Methodol. V, N, Article A (January YYYY), 2013, 39 pages, http://dx.doi.org/10.1145/0000000.0000000.
[34] Volvo Car Corporation, 2011. <https://www.media.volvocars.com/global/enhanced/en-gb/Media/Preview.aspx?mediaid=41678> (accessed February 2013).
[35] R.G. Cooper, Winning at New Products, third ed., Perseus Publishing, 2001.
[36] ABG, V-Model: Development Standard for IT-Systems of the Federal Republic of Germany, Lifecycle Process Model, 1997, <http://www.v-modell.iabg.de> (accessed February 2013).
[37] A. Peters, E. Rooney, J. Rogerson, R. McQuater, M. Spring, B. Dale, New product design and development: a generic model, TQM Mag. 11 (3) (1999) 172–179.
[38] F. Charfi, F. Sellami, Overview on dependable embedded systems in modern automotive, in: Proc. of the 2004 IEEE Int. Conference on Industrial Technology, 2004.
[39] Object Management Group OMG, OMG UML 2.3, Infrastructure, 2010. <http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/>.
[40] J.R. Galbraith, Designing Complex Organizations, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1973.
[41] N. Maiden, What has requirements research ever done for us, IEEE Softw. 22 (4) (2005) 104–105.
[42] I. Jacobson, M. Christerson, P. Jonsson, G. Övergaard, Object-Oriented Software Engineering – A use Case Driven Approach, Addison Wesley, 1992.
[43] B. Potter, J. Sinclair, D. Till, An Introduction to Formal Specification and Z, Prentice Hall, 1996.
[44] I. Sommerville, Software Engineering, eighth ed., Addison-Wesley Publishers Ltd., 2007.
[45] J. Pernstål, A. Magazinius, T. Gorschek, A study investigating challenges in the interface between product development and manufacturing in the development of software-intensive automotive systems, Int. J. Software Eng. Knowl. Eng. 22 (7) (2012) 965–1004.
[46] Object Management Group OMG, (OMG UML) Version 2.3, Infrastructure, 2010. <http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/> (accessed February 2013).
[47] IEEE Standard: 830–1998, Recommended Practice for Software Requirements Specifications.
[48] T. Gorschek, M. Svahnberg, K. Tejle, Introduction and application of a lightweight requirements engineering process evaluation method, in: Proceedings of the Ninth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03), Essen, Germany, 2003, pp. 101–112.
[49] M. Ivarsson, T. Gorschek, Practice selection framework, Int. J. Software Eng. Knowl. Eng. 22 (1) (2012) 17–58.
[50] ISO/IEC 15288: 2008 Systems engineering—system life cycle processes.
[51] ISO/IEC 12207: 1997 Software Life Cycle Process.
[52] T. Gorschek, A.M. Davis, Requirements engineering: in search for the dependent variables, Inf. Softw. Technol. 50 (2008) 67–75.
[53] AUTOSAR Automotive Open System Architecture, <http://www.autosar.org/> (accessed February 2013).
[54] ISO 14229-4: 2012 Road vehicles – Diagnostics systems –Diagnostic Services. <http://www.iso.org/iso/catalogue_detail.htm?csnumber=55285> (accessed February 2013).
[55] C. Karlsson, P. Åhlström, Assessing changes towards lean production, Int. J. Oper. Prod. Manage. 16 (1996) 24–41.
[56] P. Middleton, P.S. Taylor, A. Flaxel, A. Cookson, Lean principles and techniques for improving the quality and productivity of software development projects: a case study, Int. J. Prod. Qual. Manage. 2 (2007) 387–403.

[57] T. Sekimura, T. Maruyama, Development of enterprise business application software by introducing Toyota production system, Fujitsu Sci. Tech. J. 42 (2006) 407–413.
[58] K. Petersen, C. Wohlin, Software process improvement through the Lean Measurement (SPILEAM) method, J. Syst. Softw. 83 (2010) 1275–1287.
[59] B. Ramesh, L. Cao, R. Baskerville, Agile requirements engineering practices and challenges: an empirical study, Inf. Syst. J. 20 (5) (2010) 449–480.
[60] J. Savolainen, J. Kuusela, A. Vilavaara, Transition to agile development – rediscovery of important requirements engineering practices, In: Proceedings of the Requirements Engineering Conference, IEEE, 2010, pp. 289–294.
[61] N. Lakemond, G. Johansson, T. Magnusson, K. Safsten, Interfaces between technology development, product development and production: critical factors and a conceptual model, Int. J. Technol. Intell. Planning 3 (4) (2007) 317–330.
[62] A. Vandevelde, R. Van Dierdonk, Managing the design-manufacturing interface, Int. J. Oper. Prod. Manage. 23 (11) (2003) 1326–1348.
[63] C.J. Hardy, J.B. Thompson, H.M. Edwards, The use, limitations and customization of structured systems development methods in the United Kingdom, Inf. Softw. Technol. 37 (9) (1995) 467–477.
[64] J. Pernstal, T. Gorschek, R. Feldt, D. Florén, Software process improvement in inter-departmental development of software-intensive automotive systems – a case study, Product-Focused Software Process Improvement∗ Lecture Notes in Computer Science, vol. 7983, Springer, 2013, pp. 93–107.
[65] F. Pettersson, M. Ivarsson, T. Gorschek, P. Öhman, A practitioner's guide to light weight software process assessment and improvement planning, J. Syst. Softw. 81 (2008) 972–995.
[66] Siemens Teamcenter Systems Engineering and Requirements Management. <http://www.plm.automation.siemens.com/en_us/products/teamcenter/index.shtml> (May 2015).
[67] Vector Informatik GmbH. <http://vector.com/> (May 2015).
[68] SystemWeaver. <http://www.systemweaver.se/> (May 2015).
[69] J. Shook, Toyota's secret: the A3 report, MIT Sloan Manage. Rev., Summer 50 (4) (2009) 30.
[70] D.K. Sobek, A. Smalley, Understanding A3 Thinking—A Critical Component of Toyota's PDCA Management System, CRC Press, Boca Raton, USA, 2008.
[71] H. Rudolf, F. Paulisch, Experience report: product creation through lean approaches, Lecture Notes in Business Information Processing, vol. 65, Springer, 2010, pp. 104–110.
[72] T. Gorschek, P. Garre, S. Larsson, C. Wohlin, A model for technology transfer in practice, IEEE Softw. 23 (2006) 88–95.
[73] C. Robson, Real World Research: A Resource for Social Scientists and Practitioners-Researchers, second ed., Blackwell, 2002.
[74] M.B. Miles, A.M. Huberman, Qualitative Data Analysis: An Expanded Sourcebook, second ed., Thousand Oaks, Calif.: Sage, 1994.